

```

1 ' -----[ Title ]-----
2 ' PS2 Control - DisplayValuesOfButtonsAndJoysticks.bs2
3 '
4 '
5 ' File..... PSX_Demo.BS2
6 ' Purpose... PlayStation Controller Interface
7 ' Author.... Jon Williams
8 ' E-mail.... jwilliams@parallax.com
9 ' Started...
10 ' Updated... 17 JUL 2003
11 '
12 '
13 ' {$STAMP BS2}
14 ' {$PBASIC 2.5}
15 '
16 ' -----[ Program Description ]-----
17 '
18 ' The purpose of this program is to show the values of all the buttons and
19 ' the joysticks. This information can be used later on to set up a MATLAB
20 ' program that can take these values (and the buttons these values
21 ' correspond to) and converts them into useful information. After finding
22 ' which values correspond to which buttons pressed you can enter this
23 ' information into the diagram located directly below the main loop.
24 '
25 ' This program demonstrates the essential interface between the BASIC
26 ' Stamp and a Sony PlayStation (or compatible) game controller. This
27 ' code assumes that the clock signal is inverted between the Stamp and
28 ' the controller to allow simpler [less sophisticated] interface with
29 ' SHIFTOUT and SHIFTIN.
30 '
31 ' Note: The interface and portions of code are based on previous work by
32 ' Aaron Dahlen.
33 '
34 ' -----[ Variables/Constants/Pins ]-----
35
36 FreqDetectable    CON    3000
37
38 Piezospeaker      PIN    4           'Speaker
39
40 ' For PS2 controller
41 ' -----[ I/O Definitions ]-----
42
43 PsxClk    PIN 9           ' PSX joystick interface
44 PsxAtt    PIN 8
45 PsxCmd    PIN 7
46 PsxDat    PIN 6
47
48 ' -----[ Constants ]-----
49
50 Inverted    CON 1           ' inverted clock signal
51 Direct      CON 0           ' no inverter in clock line
52 ClockMode   CON Inverted
53
54
55 ' -----[ Variables ]-----
56
57 idx        VAR Nib         ' loop counter
58 psxOut     VAR Byte        ' byte to controller
59 psxIn      VAR Byte        ' byte from controller
60
61 ' joystick packet
62 psxID      VAR Byte        ' controller ID
63 psxThumbL  VAR Byte        ' left thumb buttons
64 psxThumbR  VAR Byte        ' right thumb buttons
65 psxStatus  VAR Byte        ' status ($5A)
66 psxJoyRX   VAR Byte        ' r joystick - X axis

```

```

67 psxJoyRY  VAR Byte          ' r joystick - Y axis
68 psxJoyLX  VAR Byte          ' l joystick - X axis
69 psxJoyLY  VAR Byte          ' l joystick - Y axis
70
71
72 ' -----[ Initialization ]-----
73
74 FREQOUT Piezospeaker, 2000, FreqDetectable      'Signal program start/reset.
75
76 Setup:
77 HIGH PsxAtt          ' deselect PSX controller
78 OUTPUT PsxCmd
79 PsxClk = ~ClockMode ' release clock
80 OUTPUT PsxClk        ' make clock an output
81
82 ' -----[ Program Code ]-----
83 Main:
84 DO ' Main loop
85   GOSUB Get_PSX_Packet_Fast ' type and packet
86   DEBUG HOME, "Type = "
87   IF (psxStatus = $5A) THEN
88
89     DEBUG "psxThumbL = ", DEC psxThumbL, " ",
90     "psxThumbR = ", DEC psxThumbR, " "
91
92     DEBUG IHEX2 psxId, "paxStatus (", DEC1 psxStatus, ")",
93     CLREOL, CR, CR,
94     "ThumbL = ", BIN8 psxThumbL, " ThumbR = ", BIN8 psxThumbR, " "
95     IF (psxId <> $41) THEN
96       DEBUG " JoyLX = ", DEC3 psxJoyLX, " JoyLY = ", DEC3 psxJoyLY, " ",
97       "JoyRX = ", DEC3 psxJoyRX, " JoyRY = ", DEC3 psxJoyRY
98     ELSE
99       DEBUG CLREOL
100    ENDIF
101  ELSE
102    DEBUG "Unknown. No response.", " psxID = ", DEC psxID, CR, CLRDN
103    PAUSE 1000
104  ENDIF
105  PAUSE 100
106 LOOP
107 END ' Main loop
108
109
110 ' --- Enter the button values you found below (for your own record) ---
111 '
112 ' Diagram 1
113 '
114 '           L (left)           R (right)
115 '
116 '           |                   |
117 '           | [ ] | A           | [ ] | B
118 '           | [ ] | C           | [ ] | D
119 '           |                   |
120 '
121 '
122 '           |                   |
123 '           |                   |
124 '           / H \ / L \
125 '           | O   M   N   O |
126 '           | EO  OG  [ ] [ ] IO  OK |
127 '           | O           [ ]   O |
128 '           \ F / \ O / \ J /
129 '           |           |           |
130 '           |           |           |
131 '           |           |           |
132 '           |           |           |

```

```

133 '
134 '
135 ' Button Numbers Chart:
136 '
137 ' For an indepth explanation of this section, see the notes
138 ' section of PS2ControllerConnectToMATLAB.bs2.
139 '
140 ' **Replace the BIN8 and DEC numbers with the values you found**
141 '
142 ' psxThumbR:
143 ' A: (L 2),          BIN8 = **11111110**, DEC = **254**
144 ' B: (R 2),          BIN8 = **11111101**, DEC = **253**
145 ' C: (L 1),          BIN8 = **11111011**, DEC = **251**
146 ' D: (R 1),          BIN8 = **11110111**, DEC = **247**
147 '
148 ' psxThumbL
149 ' E: (L arrow),      BIN8 = **01111111**, DEC = **127**
150 ' F: (down arrow),  BIN8 = **10111111**, DEC = **191**
151 ' G: (R arrow),      BIN8 = **11011111**, DEC = **223**
152 ' H: (up arrow),     BIN8 = **11101111**, DEC = **239**
153 '
154 ' psxThumbR
155 ' I: (Square),       BIN8 = **01111111**, DEC = **127**
156 ' J: (X),            BIN8 = **10111111**, DEC = **191**
157 ' K: (Circle),       BIN8 = **11011111**, DEC = **223**
158 ' L: (Triangle),     BIN8 = **11101111**, DEC = **239**
159 '
160 ' psxThumbL
161 ' M: (Select),       BIN8 = **11111110**, DEC = **254**
162 ' N: (Start),        BIN8 = **11110111**, DEC = **247**
163 ' O: (Analog),       Press to activate joystic controles
164 '
165 '
166 ' Below is the numbering schemae for the two joysticks (thy both use the same numbering scheme).
167 ' All numbers shown below are in decimal format.
168 '
169 '
170 '
171 '
172 ' Q: (psxJoyR),
173 ' P: (psxJoyL),      (000, 127) <-- (127, 127) --> (255, 127)
174 '
175 '
176 '
177 '
178 '
179 '
180 '
181 '
182 ' -----[ Subroutines ]-----
183 '
184 ' Transmit psxOut to, and receive psxIn from the
185 ' PSX controller
186 '
187 PSX_TxRx:
188   FOR idx = 0 TO 7
189     PsxCmd = psxOut.LOWBIT(idx)          ' setup command bit
190     PsxClk = ClockMode                   ' clock the bit
191     psxIn.LOWBIT(idx) = PsxDat           ' get data bit
192     PsxClk = ~ClockMode                  ' release clock
193   NEXT
194 RETURN
195
196
197 ' This routine combines manual and built-in shifting
198 ' routines to get the best speed and all valid data.

```

