

```

1 ' -----[ Title ]-----
2 ' PS2Control - PS2ControllerConnectToMATLAB.bs2
3 ' Modification of Aaron Klapheck's program: RoboticHandAndMovementsUsingPs2WithPing.bs2
4 ' Modified on: 27-Aug-08
5 '
6 ' File..... PSX_Demo.BS2
7 ' Purpose... PlayStation Controller Interface
8 ' Author.... Jon Williams
9 ' E-mail.... jwilliams@parallax.com
10 ' Started...
11 ' Updated... 17 JUL 2003
12 '
13 '

```

```

14 ' {$STAMP BS2}
15 ' {$PBASIC 2.5}

```

```

17 ' -----[ Program Description ]-----

```

```

19 ' This program demonstrates the essential interface between the BASIC
20 ' Stamp and a Sony PlayStation (or compatible) game controller. This
21 ' code assumes that the clock signal is inverted between the Stamp and
22 ' the controller to allow simpler [less sophisticated] interface with
23 ' SHIFTOUT and SHIF TIN.

```

```

25 ' Note: The interface and portions of code are based on previous work by
26 ' Aaron Dahlen.

```

```

28 ' -----[ Aaron's Add-On's ]-----

```

```

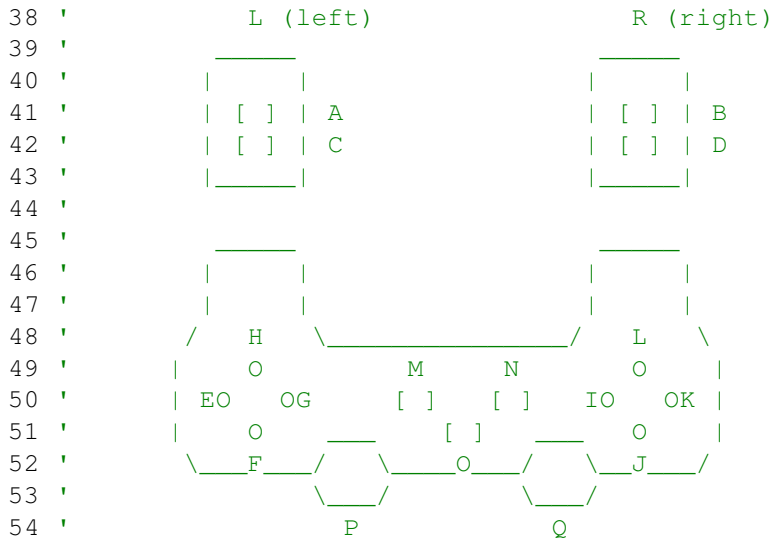
30 ' This section of code is written by Aaron Klapheck.
31 ' 6 AUG 2008
32 ' Below is the diagram of a Sony PlayStation 2 remote control.
33 ' The numbers below are specific to Sony brand controlers.

```

```

36 ' Diagram 1.

```



```

59 ' Button Numbers Chart: (see explanation below)

```

```

61 ' **Replace the BIN8 and DEC numbers with the values you found
62 '   when you ran DisplayValuesOfButtonsAndJoysticks.bs2**

```

```

64 ' psxThumbR:
65 ' A: (L 2),           BIN8 = 11111110, DEC = 254
66 ' B: (R 2),           BIN8 = 11111101, DEC = 253

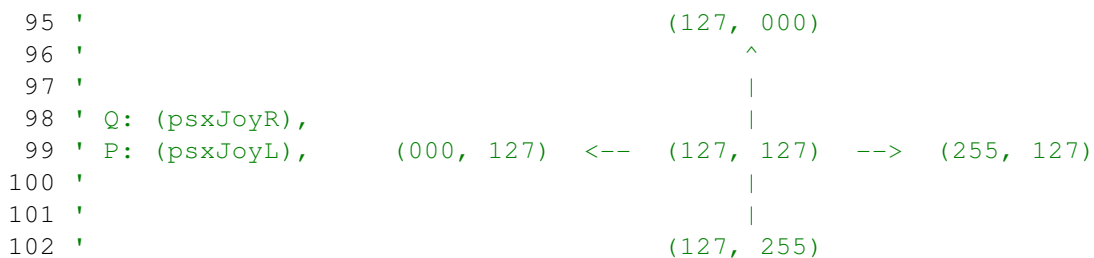
```

```

67 ' C: (L 1),          BIN8 = 11111011, DEC = 251
68 ' D: (R 1),          BIN8 = 11110111, DEC = 247
69 '
70 ' psxThumbL
71 ' E: (L arrow),      BIN8 = 01111111, DEC = 127
72 ' F: (down arrow),  BIN8 = 10111111, DEC = 191
73 ' G: (R arrow),      BIN8 = 11011111, DEC = 223
74 ' H: (up arrow),     BIN8 = 11101111, DEC = 239
75 '
76 ' psxThumbR
77 ' I: (Square),       BIN8 = 01111111, DEC = 127
78 ' J: (X),            BIN8 = 10111111, DEC = 191
79 ' K: (Circle),       BIN8 = 11011111, DEC = 223
80 ' L: (Triangle),     BIN8 = 11101111, DEC = 239
81 '
82 ' psxThumbL
83 ' M: (Select),       BIN8 = 11111110, DEC = 254
84 ' N: (Start),        BIN8 = 11110111, DEC = 247
85 ' O: (Analog),       Press to activate joystic controles
86 '
87 ' No buttons pressed:
88 ' psxThumbL,         BIN8 = 11111111, DEC = 255
89 ' psxThumbR,         BIN8 = 11111111, DEC = 255
90 '
91 '

```

92 ' Below is the numbering schemae for the two joysticks (thy both use the same numbering scheme).  
93 ' All numbers shown below are in decimal format.



107 ' Explanation of Button numbers:

108 ' For each button pressed there is a specific number stored in either  
109 ' the psxThumbR or psxThumbL variable depending on which side of the controller  
110 ' the button is on.  
111 ' The button values stored in the psxThumbR or psxThumbL variables are displayed  
112 ' to the right of the button label in both binomial and decimal format.  
113 ' When only one button is being pressed at a time the decimal format will work  
114 ' fine. But if you want to be able to press multiple buttons at a time then the  
115 ' binomial format may be easier to use.  
116 ' Although the numbering scheme for all playstation remotes should be the same double check!

120 ' How to use the Button Numbers Chart:

122 ' What is shown above in "Button Numbers Chart":

```

123 ' psxThumbR:
124 ' A: (L 2),          BIN8 = 11111110, DEC = 254
125 ' .
126 ' .
127 ' .
128 '

```

129 ' What this means:

130 ' psxThumbR. Means: The variable used to store information  
131 ' A. Means: the arbitrary designation of the button in question (See Diagram 1)  
132 ' (L 2). Means: the more descriptive name that I refer to when typing the code

```

133 ' BIN8 = 11111110. Means: the binomial number that is stored in the variable listed
134 ' DEC = 254. Means: the decimal number that is stored in the variable listed
135 '
136 '
137 ' -----[ MATLAB transfer ]-----
138 '
139 ' In digital mode
140 ' All button values will be sent to MATLAB. Only one button value at a
141 ' time will be sent to MATLAB (only press one button at a time). However,
142 ' two buttons can be pressed if one is stored in psxThumbR and the other
143 ' is stored in psxThumbL.
144 '
145 ' In analog mode
146 ' Don't use this.
147 '
148 '
149 ' -----[ Variables/Constants/Pins ]-----
150
151
152 FreqDetectable CON 3000
153
154 Piezospeaker PIN 4 ' Speaker
155
156
157 ' Stuff for MATLAB communication
158 sPin CON 16 'Serial Pin - P16, Programming port
159 Baud CON 84 'Baud mode for a rate of 9600, 8-N-1
160 'BS2P, BS2SX use 240 for 9600, 8-N-1
161
162
163 ' For PS2 controller
164 ' -----[ I/O Definitions ]-----
165
166 PsxClk PIN 9 ' PSX joystick interface
167 PsxAtt PIN 8
168 PsxCmd PIN 7
169 PsxDat PIN 6
170
171 ' -----[ Constants ]-----
172
173 Inverted CON 1 ' inverted clock signal
174 Direct CON 0 ' no inverter in clock line
175 ClockMode CON Inverted
176
177 ' -----[ Variables ]-----
178
179 idx VAR Nib ' loop counter
180 psxOut VAR Byte ' byte to controller
181 psxIn VAR Byte ' byte from controller
182
183 counter VAR Word
184
185 ' joystick packet
186 psxID VAR Byte ' controller ID
187 psxThumbL VAR Byte ' left thumb buttons
188 psxThumbR VAR Byte ' right thumb buttons
189 psxStatus VAR Byte ' status ($5A)
190 psxJoyRX VAR Byte ' r joystick - X axis
191 psxJoyRY VAR Byte ' r joystick - Y axis
192 psxJoyLX VAR Byte ' l joystick - X axis
193 psxJoyLY VAR Byte ' l joystick - Y axis
194
195
196 ' -----[ Initialization ]-----
197
198 FREQOUT Piezospeaker, 2000, FreqDetectable 'Signal program start/reset.

```

```

199 SEROUT sPin, Baud, [LF] 'Send a lone LF to signal MATLAB start
200
201
202 SEROUT sPin, Baud, ["This is your Boe-Bot", LF]
203
204 Setup:
205 HIGH PsxAtt ' deselect PSX controller
206 OUTPUT PsxCmd
207 PsxClk = ~ClockMode ' release clock
208 OUTPUT PsxClk ' make clock an output
209
210 ' -----[ Program Code ]-----
211
212 Main:
213 DO
214 PAUSE 200 ' Slow the transfer down otherwise information is transferred
215 ' too quickly in MATLAB for the user to see it.
216
217 GOSUB Get_Psx_Packet_Fast ' type and packet
218
219 IF (psxId <> $41) THEN ' If in analog mode (Don't use)
220
221 ' Only send information from the right joystic controller
222 ' SEROUT sPin, Baud, [DEC psxJoyRX, ",", DEC psxJoyRY, LF]
223
224 ELSE ' If in digital mode
225
226 ' All button values will be sent to MATLAB
227 SEROUT sPin, Baud, [BIN8 psxThumbL, ",", BIN8 psxThumbR, LF]
228
229 ENDIF ' (psxId <> $41)
230 LOOP ' Main loop
231
232 END
233
234
235 ' -----[ PS2 Subroutines ]-----
236
237
238 ' This routine combines manual and built-in shifting
239 ' routines to get the best speed and all valid data.
240 '
241 ' Execution time on BS2 is ~40 ms.
242
243 Get_Psx_Packet_Fast:
244 LOW PsxAtt ' select controller
245 SHIFTOUT PsxCmd, PsxClk, LSBFIRST, [$01] ' send "start"
246 psxOut = $42 : GOSUB PSX_TxRx ' send "get data"
247 psxId = psxIn ' save controller type
248 SHIFTIM PsxDat, PsxClk, LSBPOST, [psxStatus] ' should be $5A ("ready")
249 SHIFTIM PsxDat, PsxClk, LSBPOST, [psxThumbL]
250 SHIFTIM PsxDat, PsxClk, LSBPOST, [psxThumbR]
251 SHIFTIM PsxDat, PsxClk, LSBPOST, [psxJoyRX]
252 SHIFTIM PsxDat, PsxClk, LSBPOST, [psxJoyRY]
253 SHIFTIM PsxDat, PsxClk, LSBPOST, [psxJoyLX]
254
255 SHIFTIM PsxDat, PsxClk, LSBPOST, [psxJoyLY] ' If you want to use the left joystick
256 ' then delete this part and uncomment
257 ' the gosub PSX_TxRx below.
258 'GOSUB PSX_TxRx : psxJoyLY = psxIn
259 HIGH PsxAtt ' deselect controller
260 RETURN
261
262
263
264

```

