```
' -----[ Title ]-----------------------------------------------------------
' PS2Control - PS2ControllerConnectToMATLAB.bs2
' Modification of Aaron Klapheck's program: RoboticHandAndMovementsUsingPs2WithPing.bs2
' Modified on: 27-Aug-08
'
' File..... PSX_Demo.BS2
' Purpose... PlayStation Controller Interface
' Author.... Jon Williams
' E-mail.... jwilliams@parallax.com
' Started...
' Updated... 17 JUL 2003
'
'
' {$STAMP BS2}
' {$PBASIC 2.5}
'
' -----[ Program Description ]---------------------------------------------
'
' This program demonstrates the essential interface between the BASIC
' Stamp and a Sony PlayStation (or compatible) game controller. This
' code assumes that the clock signal is inverted between the Stamp and
' the controller to allow simpler [less sophisticated] interface with
' SHIFTOUT and SHIFTIN.
'
' Note: The interface and portions of code are based on previous work by
' Aaron Dahlen.
'
' -----[ Aaron's Add-On's ]------------------------------------------------
'
' This section of code is written by Aaron Klapheck.
' 6 AUG 2008
' Below is the diagram of a Sony PlayStation 2 romote control.
' The numbers below are specific to Sony brand controlers.
'
'
' Diagram 1.
'
'           L (left)                R (right)
'          _____                    _____
'         |     |                  |     |
'         | [ ] | A                | [ ] | B
'         | [ ] | C                | [ ] | D
'         |_____|                  |_____|
'
'          _____                    _____
'         |     |                  |     |
'         |     |                  |     |
'       /   H   \________________/   L   \
'      |    O          M     N       O    |
'      | EO   OG     [ ]   [ ]   IO   OK |
'      |    O    ___     [ ]   ___   O    |
'       \___F___/   \____O___/   \__J___/
'                \___/         \___/
'                  P             Q
'
'
'
'
' Button Numbers Chart: (see explination below)
'
' **Replace the BIN8 and DEC numbers with the values you found
'   when you ran DisplayValuesOfButtonsAndJoysticks.bs2**
'
' psxThumbR:
' A: (L 2),             BIN8 = 11111110, DEC = 254
' B: (R 2),             BIN8 = 11111101, DEC = 253
```

```
' C: (L 1),            BIN8 = 11111011, DEC = 251
' D: (R 1),            BIN8 = 11110111, DEC = 247
'
' psxThumbL
' E: (L arrow),        BIN8 = 01111111, DEC = 127
' F: (down arrow),     BIN8 = 10111111, DEC = 191
' G: (R arrow),        BIN8 = 11011111, DEC = 223
' H: (up arrow),       BIN8 = 11101111, DEC = 239
'
' psxThumbR
' I: (Square),         BIN8 = 01111111, DEC = 127
' J: (X),              BIN8 = 10111111, DEC = 191
' K: (Circle),         BIN8 = 11011111, DEC = 223
' L: (Triangle),       BIN8 = 11101111, DEC = 239
'
' psxThumbL
' M: (Select),         BIN8 = 11111110, DEC = 254
' N: (Start),          BIN8 = 11110111, DEC = 247
' O: (Analog),         Press to activate joystic controles
'
' No buttons pressed:
' psxThumbL,           BIN8 = 11111111, DEC = 255
' psxThumbR,           BIN8 = 11111111, DEC = 255
'
'
' Below is the numbering schemae for the two joysticks (thy both use the same numbering scheme).
' All numbers shown below are in decimal format.
'
'                                  (127, 000)
'                                       ^
'                                       |
' Q: (psxJoyR),                         |
' P: (psxJoyL),     (000, 127)  <--  (127, 127)  -->  (255, 127)
'                                       |
'                                       |
'                                  (127, 255)
'
'
'
'
' Explination of Button numbers:
'   For each button pressed there is a specific number stored in either
'   the psxThumbR or psxThumbL variable depending on which side of the controler
'   the button is on.
'   The button values stored in the psxThumbR or psxThumbL variables are displayed
'   to the right of the button label in both binomial and decimal format.
'   When only one button is being pressed at a time the decimal format will work
'   fine. But if you want to be able to press multiple buttons at a time then the
'   binomial format may be easer to use.
'   Although the numbering scheme for all playstation remotes should be the same double check!
'
'
'
' How to use the Button Numbers Chart:
'
'   What is shown above in "Button Numbers Chart":
'     psxThumbR:
'     A: (L 2),            BIN8 = 11111110, DEC = 254
'     .
'     .
'     .
'
'   What this means:
'     psxThumbR. Means: The variable used to store information
'     A. Means: the arbitrary designation of the button in question (See Diagram 1)
'     (L 2). Means: the more discriptive name that I reffer to when typing the code
```

```
'     BIN8 = 11111110. Means: the binomial number that is stored in the variable listed
'     DEC = 254. Means: the decimal number that is stored in the variable listed
'
'
' -----[ MATLAB transfer ]-------------------------------------------------
'
' In digital mode
'   All button values will be sent to MATLAB. Only one button value at a
'   time will be sent to MATLAB (only press one button at a time). However,
'   two buttons can be pressed if one is stored in psxThumbR and the other
'   is stored in psxThumbL.
'
' In analog mode
'   Don't use this.
'
'
' -----[ Variables/Constants/Pins ]----------------------------------------


FreqDetectable    CON    3000

Piezospeaker      PIN    4                ' Speaker


' Stuff for MATLAB communication
sPin      CON   16              'Serial Pin - P16, Programming port
Baud      CON   84              'Baud mode for a rate of 9600, 8-N-1
                                'BS2P, BS2SX use 240 for 9600, 8-N-1


' For PS2 controler
' -----[ I/O Definitions ]-------------------------------------------------

PsxClk    PIN 9             ' PSX joystick interface
PsxAtt    PIN 8
PsxCmd    PIN 7
PsxDat    PIN 6

' -----[ Constants ]-------------------------------------------------------

Inverted  CON 1             ' inverted clock signal
Direct    CON 0             ' no inverter in clock line
ClockMode CON Inverted

' -----[ Variables ]-------------------------------------------------------

idx       VAR Nib           ' loop counter
psxOut    VAR Byte          ' byte to controller
psxIn     VAR Byte          ' byte from controller

counter   VAR Word

' joystick packet
psxID     VAR Byte          ' controller ID
psxThumbL VAR Byte          ' left thumb buttons
psxThumbR VAR Byte          ' right thumb buttons
psxStatus VAR Byte          ' status ($5A)
psxJoyRX  VAR Byte          ' r joystick - X axis
psxJoyRY  VAR Byte          ' r joystick - Y axis
psxJoyLX  VAR Byte          ' l joystick - X axis
psxJoyLY  VAR Byte          ' l joystick - Y axis


' -----[ Initialization ]--------------------------------------------------

FREQOUT Piezospeaker, 2000, FreqDetectable          'Signal program start/reset.
```

```
199
200 SEROUT sPin, Baud, [LF]                        'Send a lone LF to signal MATLAB start
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 transfered
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 controler
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   SHIFTIN PsxDat, PsxClk, LSBPOST, [psxStatus]    ' should be $5A ("ready")
249   SHIFTIN PsxDat, PsxClk, LSBPOST, [psxThumbL]
250   SHIFTIN PsxDat, PsxClk, LSBPOST, [psxThumbR]
251   SHIFTIN PsxDat, PsxClk, LSBPOST, [psxJoyRX]
252   SHIFTIN PsxDat, PsxClk, LSBPOST, [psxJoyRY]
253   SHIFTIN PsxDat, PsxClk, LSBPOST, [psxJoyLX]
254
255   SHIFTIN 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
```

```
' Transmit psxOut to, and receive psxIn from the PSX controller.
' This allows a simultanious data exchange between the
' PS2 controler and the Basic Stamp.
' See PSX_TxRx.bs2 to see how this works

PSX_TxRx:
  FOR idx = 0 TO 7
    PsxCmd = psxOut.LOWBIT(idx)                    ' setup command bit
    PsxClk = ClockMode                             ' clock the bit
    psxIn.LOWBIT(idx) = PsxDat                     ' get data bit
    PsxClk = ~ClockMode                            ' release clock
  NEXT
RETURN
```