DPS-NEAT SJ-2 & SJ-3
RS-232 SERIAL JOYSTICKS

Operators Manual
Version 1.2

- **Two axis (SJ-2) and three axis (SJ-3) versions**
- **9600 baud, RS-232 serial output**
- **Uncommitted toggle and momentary push-buttons**

The NEAT SJ-2 and SJ-3 Serial Joysticks provide convenient digitization and communication of operator manual movements for use in a wide range of position and velocity control applications. The SJ-2 digitizes two axes ("X" and "Y"), while the SJ-3 incorporates a stick-mounted, spring-loaded potentiometer for control of a third axis ("Z" or "θ"). Both units feature an eight bit digitization of joystick angle, together with a 9600 baud, RS-232 serial output for processing by virtually any computer. Data is transmitted via ASCII characters, using a ten bit frame with one start bit, an eight bit word, one stop bit, and no parity. Serial data is output on a standard PC-AT 9 pin DE-9P connector, in a three wire implementation (TXD, RXD, and GND). An uncommitted enclosure mounted toggle switch and stick-mounted momentary push-button are provided, with the state of the two switches encoded and transmitted out the serial port. These switches can perform a variety of roles, such as axis selection, fast/slow speed control, teach function, etc., as determined by user supplied software. Additional plastic slide switches adjacent to the joystick bezel allow the "X" and "Y" axes to be configured as free-floating or spring-return-to-center.

The command set of the SJ-2 and SJ-3 is brief; upon receipt of a carriage return, the digitized, ASCII encoded values of two (SJ-2) or three (SJ-3) joystick axes are transmitted out the serial port, together with a single numeric character representing the state of the two switches. A typical exchange would be: host sends carriage return (ASCII 13): The SJ-3 responds with: 129, 42, 230, 3 <c.r.> <l.f.>. The first three numbers represent the eight bit digitization of each of the three axes, left justified and separated by commas, where the value returned may range from 0 to 255 (0 corresponds to full left for the "X" axis, fully towards the operator for "Y", and full CCW rotation for "Z"). The spring-return neutral position for each axis returns approximately 128. In the above response, the X axis would be in the centered position, the Y axis would be perhaps 16 degrees toward the operator, and the Z axis would be twisted about 36 degrees clockwise. The last digit, which can range from 0 to 3, encodes the state of the two switches. The stick-mounted momentary switch has a weight of zero or one, with zero corresponding to the depressed state, while the toggle switch has a weight of zero or two, with zero corresponding to a position towards the rear of the unit. Accordingly, zero = momentary depressed and toggle towards the rear; one = momentary released and toggle towards the rear; two = momentary depressed and toggle towards the front; and three = momentary released and toggle towards the front. If desired, custom silkscreened or engraved plates can be provided to identify special functions associated with the toggle switch. A carriage return (ASCII 13) and linefeed (ASCII 10) follow the data. With the SJ-2, there will only be two eight bit values, corresponding to the "X" and "Y" axes, followed by the switch digit, carriage return, and linefeed. In both the SJ-2 and SJ-3, all returned data is left justified; that is, no spaces are inserted to maintain a constant frame length. Commas delimit individual fields within the frame, and the carriage return and line feed serve as terminators.

In addition to the above response (initiated by a carriage return), two additional commands are available: an uppercase "R" will initiate a continuous series of reports, which is terminated upon the receipt of any other character. Sending an uppercase "I" to the SJ-2 or SJ-3 will result in an identifier prompt being returned, of the form: NEAT SJ-3 Serial Joystick v 1.2 <c.r.> <l.f.>. All characters other than the carriage return, uppercase "R", and uppercase "I" will be echoed to the host with no other action taken.

Serial transmission on the SJ-2 and SJ-3 occurs via a DE-9P, pin type D-submini connector located on the right rear of the enclosure. The pin-out is as follows:

1.	GND	4.	+5 volts input
2.	RxD (at host)	5.	GND
3.	TxD (at host)	6-9	not connected

The pin-out follows the three-wire implementation of the nine pin PC-AT serial convention, with the exception that pin #1 (nominally DCD) is tied to ground and +5 volts is capable of being brought in on pin #4 (nominally DTR). This allows a modified host serial port interface to supply operating voltage to the unit, eliminating the standard wall-pack power supply and cord. Please note that TxD and RxD in the above pin-out refer to the PC-AT standard, and do not reflect the actual roles played by these pins at the SJ-2 or SJ-3. That is, the SJ-2/3 transmits data on pin #2 (since that data is received on pin #2 of the PC-AT, the line is defined as RxD). Similarly, the SJ-2/3 receives data on pin #3; since that data is transmitted from the computer on pin #3, the pin is defined as TxD. The mating connector for the SJ-2/3 is a nine pin DE-9S (socket contacts); jackscrews are provided to allow a rigid interconnection. The standard serial port cable is an insulation displacement, six foot long, 9 pin ribbon cable with a connector at each end. Optional 9 pin to 25 pin adaptors are available for use with 25 pin serial port connectors. Power is normally supplied via a wall-mounted, regulated +5 volt DC supply, which plugs into a standard 5.5mm/2.1mm rear mounted coaxial connector. The power supply has a current rating of 600mA, while the SJ-2 and SJ-3 consume 330mA @ 5.0 volts DC. Do not replace the wall mounted power supply with any other voltage level or type (i.e., unregulated or AC transformers), as damage to the circuitry will result.

The SJ-2 and SJ-3 Serial Joysticks provide low level information regarding the operators hand movements. This data is then acquired over the serial port by the host computer, and interpreted by host-resident software according to the needs of the application. A typical host processing routine would be:

1. Parse the returned data to separate the axis values and switch data.

2. Decode the momentary and toggle switch status, and set the speed range, axis selection, etc. accordingly.

3. For any given axis, determine sign (direction) of intended motion by comparing the value to the neutral setting (128).

4. Determine the magnitude of the joystick angle by taking the absolute value of the difference between the measured value and 128.

5. Initiate no action if the magnitude is below a specific (possibly variable) deadband window around 128.

6. If the magnitude exceeds the deadband value, square the difference between the magnitude and the deadband threshold. This will provide a square law relationship between joystick angle and velocity, increasing dynamic range.

7. Initiate a large move in the previously determined direction, at a velocity proportional (via a specific, possibly variable, scaling value) to the result of #6 above.

8. Loop back to re-acquire current joystick angle. If direction is same but angle has changed, update velocity. If direction (relative to the spring-return center position) has changed, terminate the previous move and begin a new move in the current direction, determining the velocity as before.

A sample software driver routine is listed on page 6. This routine simply displays the joystick variables in continuous fashion; it does not perform the sign extraction, squaring, etc. described above. The program is written in Microsoft Quick Basic 4.0, and is supplied in both source (TSTJOY.BAS) and executable (TSTJOY.EXE) versions on the 5 1/4" low density diskette which accompanies this manual.

Due to the sensitivity of the digitization (1 bit = 0.2 degrees), mechanical variations in joystick stroke, and supply variations, the values returned by the joystick should not be expected to exactly match the nominal 0-128-255 span. Given the approximate nature of operator hand movements, returned values which differ from nominal by several counts are normally of no consequence. On-board regulation of potentiometer drive voltage and A/D reference voltage are provided, minimizing drift of returned values over time or temperature. If it is desired to modify the joystick response, internal adjustments can be performed. The operating principle is fairly simple: each axis consists of a 5k ohm potentiometer, with +2.5 volts across the pot. The center tap is therefore at a nominal 1.25 volts, although a mechanical centering tab allows this to be adjusted. Since the potentiometers allow 270 degrees of rotation, and the X and Y joystick angles range between ± 25° (± 45° for Z/θ), typical maximum and minimum readings vary from about 1.125 to 1.375 volts. Since the full range for these axes is only about 0.25 volts, the A:D converters are operated in differential mode, with an offset and gain adjustment to match the full eight bit digitization to the actual range. The offset adjustment for each axis should be set to equal the lowest voltage value at full joystick excursion (left on the X axis, toward the operator for Y, and CCW for Z/ø). The gain adjustment should be adjusted to equal the voltage difference between each end of joystick actuation. The specific procedure is as follows (see Fig. 1, attached, for test points):

1. Remove the six screws located on the base of the SJ2/3 and separate the enclosure.

2. Locate test points O1, O2, O3, G1 &2, and G3; Trimpots R5, R6, R7, R8, and R9; and points CT1, CT2, and CT3 (pins 7, 6, & 5 respectively) on connector J2.

3. Connect the voltmeter ground clip to the ground pin near J1 (see Fig. 1); this is power supply ground.

4. For each axis, record the voltage at full joystick negative excursion (left for X, front for Y, CCW for Z/0) by moving the joystick and measuring the voltage at points CT1, CT2, and CT3 (pins 7, 6, and 5) on connector J2).

5. Now move the joystick axes (one at a time) to full positive excursion and record the voltage at points CT1, CT2, and CT3 on connector J1.

6. For each axis, place the voltmeter on the offset point (O1, O2, and O3 for X, Y, and Z respectively) and adjust the offset pot (R6, R7, and R9 for X, Y, and Z respectively) until the previously recorded lowest value (item #4 above) is measured.

7. Place the voltmeter probe point on G1 & G2, and adjust pot R5 until the meter reads a value equal to the difference between the maximum and minimum values recorded in steps #4 and #5 for either the X or Y axis (the two differences should be very close).

8. Place the voltmeter probe on point G3 and adjust pot R8 until the meter reads a value equal to the difference between the maximum and minimum values recorded in steps #4 and #5 for the Z axis.

9. Using a terminal (or computer with terminal emulator), confirm that, for each axis, the minimum ASCII value returned over the serial position is approximately O, the neutral position returns approximately 128, and the maximum value is around 255. Readjust the pots if and as needed to trim the returned values.

```
'  TSTJOY.BAS:  NEAT SJ-3 / SJ-2 Serial Joystick Test Program
'
'  Data format from joystick:
'         SJ-3:            XXX,YYY,ZZZ,B
'         SJ-2:            XXX,YYY,B
'   where XXX=X-axis value, YYY=Y axis value, ZZZ-Z-axis value, and
'   B-buttons bit code, Bit 0 for stick push button, Bit 1 for switch
'
DEFINT A-Z
OPEN "COM1:9600,N,8,1,RS,CS,DS,CD" FOR RANDOM AS #1     'open serial port
PRINT : PRINT : PRINT "NEAT SJ- 3 / SJ-2 Serial Joystick Test Program":  PRINT
LINE INPUT "Enter 1 for SJ-3, 2 for SJ-2 (<Enter=1>) : "; A$:  PRINT
IF A$ = "2" THEN NAXES = 2 ELSE NASES = 3         ' # of axes for joystick
PRINT #1, "I";                                    ' send ' I ' for Identify command
LINE INPUT  #1, A$                                ' get ID string
IF INSTR (A$,  "Serial Joystick")  =  0 THEN      ' confirm ID string
   PRINT  "Serial Joystick not found" :  PRINT
   END
END IF
PRINT  "Device:  "; A$                            ' show ID string
PRINT  :  PRINT  "Hit <Esc> to stop":  PRINT
PRINT  "    X";  TAB (12)  ;  "Y";  TAB (22) ;  "Z"
CLINE = CSRLIN - 1:  PRINT                  ' save line and scroll screen
DO
   PRINT  #1,  CHR$ (13) ;                        ' send  <CR> to get report
   LINE INPUT #1,  A$                             ' read in report line
   LOCATE LINE<  1                                ' put cursor at beginning of line
   AX = 1:  P  =  1
   DO WHILE AX  <=  NAXES                         ' get each axis value
         Q  =  INSTR(P,  A$,  " , ")              ' each axis value followed by comma
         IF Q > P THEN
             PRINT VAL(MID$(A$, P, Q -P) ) ; "        ";  TAB(AX  *  10) ;        ' display val
             P = Q + 1
         ELSE
             PRINT  "Error reading from joystick"                    ' stop if I/O error
             END
         END IF
         AX = AX + 1
   LOOP
   BVAL = VAL(MID$ (A$, P, 1)                     ' get buttons value, show states
   IF BVAL AND 1 THEN PRINT "  Button Up          ";  ELSE PRINT "   Button Down   ";
   IF BVAL AND 2 THEN PRINT "  Switch Down        ";  ELSE PRINT "   Switch Up     ";
LOOP UNTIL INKEY$ = CHR$ (27                      ' loop until <Esc> pressed
PRINT  :  PRINT
END
```

04/94