Summary

The following demonstrates how to format serial data.

Description

In this example, data is coming in and going out of serial port 2. The incoming data is format the same way as the outgoing data. The format is as follows:

- 5 characters representing an ASCII string command
- 1 comma character
- 5 characters representing an double integer
- 1 carriage return

SEND.COMMAND takes a double integer and a string from the stack and sends out to the serial port the data packet. RECEIVE.COMMAND waits for a data packet from the serial port and when it gets it, it puts the data on the stack as a string and a double integer.

Error checking is not done, therefore improper packets or input data will not be detected.

```
DECIMAL
4 USE.PAGE
20 WIDTH !
ANEW SERIAL.COMMUNICATIONS

12 CONSTANT SERIAL2.BUFFER.LEN      \ Length of serial2.buffer
12 CONSTANT PACKET.LEN              \ = COMMAND.LEN + 1 (COMMA) + PARAMETER.LEN + 1
(CR)
5  CONSTANT COMMAND.LEN             \ Length of command string
5  CONSTANT PARAMETER.LEN           \ Length of command parameter
1  CONSTANT CNT.LEN                 \ 1st byte of string is the string count
6  CONSTANT TEMP.BUF.LEN            \ = CNT.LEN + PARAMETER.LEN
7  CONSTANT PARAMETER.OFFSET        \ = CNT.LEN + COMMAND.LEN + 1 (COMMA)


10 CONSTANT LINE.FEED               \ ASCII number for line.feed
13 CONSTANT CARRIAGE.RETURN         \ ASCII number for carriage return
44 CONSTANT COMMA                   \ ASCII number for comma

4800 BAUD2

VARIABLE SERIAL2.BUFFER             \ Set up serial2's buffer
SERIAL2.BUFFER.LEN 1 + VALLOT

VARIABLE TEMP.BUFFER                \ Set up a temp buffer
SERIAL2.BUFFER.LEN 1 + VALLOT
```

```
\ SEND.COMMAND will take a command string (must be 5 characters only)
\ and a command parameter (double int) and send them out serial 2.
\ typing in the following
\
\      " AMOVE" 39878 SEND.COMMAND
\
\ will send the command AMOVE to the position 39878. It doesn't check
\ to see if the command string is longer or shorter than 5 characters.
\ Parameter must be an integer less than 5 digits.  Parameter will be
\ emitted right justified, for example 1 will be
\ sent out as ____1 where _ indicates a space (ASCII 32)
: SEND.COMMAND      ( xaddr\d1 -- | command string\command parameter )
    LOCALS{ d&input.parameter x&input.string }
    4800 BAUD2
    USE.SERIAL2
    COMMAND.LEN 0 DO                          \ Sends first 5 characters of input.string
        x&input.string CNT.LEN I + XN+ C@  \ Retrieve Ith input string character
        EMIT2                                 \ Emit character to SERIAL2
    LOOP
    COMMA EMIT2                               \ Emit comma to SERIAL2
    d&input.parameter PARAMETER.LEN D.R    \ Emit command parameter, right justified
    CARRIAGE.RETURN EMIT2                     \ Emit carriage return
    LINE.FEED EMIT2                           \ Emit line feed
    USE.SERIAL1
;

\ RECEIVE.COMMAND will receive a data packet and return a command and parameter.
\ There no error checking, therefore the inputs must be the correct format or this
will
\ not work. The SERIAL2.BUFFER is initially used to store the input string. Next, the
\ size of the buffer is shortened to only include the first COMMAND.LEN characters.
: RECEIVE.COMMAND        ( -- xaddr\d1 | command string\command parameter )
    LOCALS{ | d&output.parameter x&output.string }
    4800 BAUD2
    USE.SERIAL2
    UEMIT X@ CFA.FOR DROP UEMIT X!                    ( emit.xcfa -- ) \ kill the echo
    SERIAL2.BUFFER CNT.LEN XN+ PACKET.LEN EXPECT    \ Receives packet to buffer
    UEMIT X!                                          \ restore echo
    COMMAND.LEN SERIAL2.BUFFER C!                     \ shorten length of
SERIAL2.BUFFER
    SERIAL2.BUFFER                                    \ Leave SERIAL2.BUFFER on stack

    PARAMETER.LEN 0 DO
        SERIAL2.BUFFER PARAMETER.OFFSET I + XN+ C@  \ Retrieve Ith character
        TEMP.BUFFER CNT.LEN I + XN+ C!                \ Store character in temp string
    LOOP
    BL TEMP.BUFFER TEMP.BUF.LEN XN+ C!                \ Store space bar at end of
string
    PARAMETER.LEN TEMP.BUFFER C!                      \ Store count at start of string

    TEMP.BUFFER NUMBER                                \ Convert string to number
    CASE
        0 OF         ENDOF                            \ Error condition - no conversion
        1 OF S>D    ENDOF                             \ Convert to double precision
    ENDCASE
    USE.SERIAL1
;
```

Mosaic Industries
5437 Central Ave Suite 1, Newark, CA 94560 Telephone: (510) 790-8222 Fax: (510) 790-0925