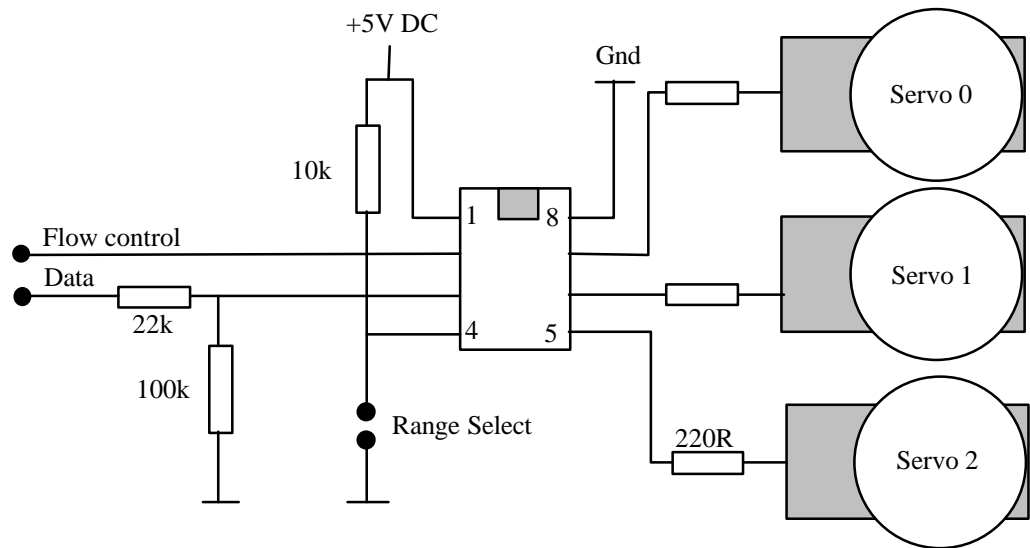


MILFORD INSTRUMENTS Ltd



Micro Serial Servo Driver Chip

Select the range required- Extended servo range (0.5-2.5msecs) with pin4 pulled HIGH,
Standard servo range (1.0-2.0msecs) with pin4 pulled LOW

Sample BASIC Stamp2 code:

```
'Micro Servo Driver Chip July 2000 Milford Instruments
'info@milinst.com
'Sample code for use with the BS2-IC

'USDC connections:
'Pin 1 +5Vdc
'Pin 2 Flow control- low when chip is busy,high when ready to receive data
'Pin 3 RS232 data- 9600 inverted format- pull to ground with 100K resistor
'Pin 4 Unused
'Pin 5 Servo 2 output- use 220R in series to protect chip
'Pin 6 Servo 1 output- use 220R in series to protect chip
'Pin 7 Servo 0 output- use 220R in series to protect chip
'Pin 8 Ground

'Usdc command info:
'Send two bytes, <command><position>
'Command byte format: XXXXYYYY
'XXXX is speed 0 to 15- number of frames per increment (0:fast,15:slow)
'YYYY is action :
'Move To actions:
'      0000- send servo 0 (pin 7) to position
'      0001- send servo 1 (pin 6) to position
'      0010- send servo 2 (pin 5) to position
'      0011- not implemented- ignored
'Report current position actions:
```

```

'          0100- report servo 0 position
'          0101- report servo 1 position
'          0110- report servo 2 position
'          0111- not implemented- ignored
'          response will be number from 0 to 255 representing current servo
position.
'          1000- report all servos
'          response will be %BBBBlXYZ where XYZ may be 1 or 0
'          depending on whether that servo has reached its end
'          point (1) or not (0)- a quick way to read all servos.
'          Note that bits B depend on whether the servo is switched ON
'          (1) or switched OFF (0) - mask them if you don't know
'          by ANDing with %00001111
'          1001,1010,1011 commands ignored
'Servo Active actions
'          1100- Switch OFF servo 0
'          1101- Switch OFF servo 1
'          1110- Switch OFF servo 2
'          1111- Not implemented- ignored
'          Servo turned ON again when Move to command received
'Position byte is between 0 and 255

'Servo connections and constants:
tx      con    0      'serial out pin to usdc pin 3
flow    con    5      'flow control pin to usdc pin 2: low on usdc pin 2 prevents tx
baud     con   16468  '9600 baud- inverted

'Servo action basics:

move      con    0      'move servo to new position
report    con    4      'report current servo position
report_all con    8      'where are all 3 servos?
disable   con   12      'switch off a particular servo

command    var    byte    'whole command byte- action and speed
action     var    command.lownib 'servo action
servo      var    nib      'servo number 0 to 2
position   var    byte
speed      var    command.highnib
result     var    byte
mask       var    byte      'result mask

start:                                           'Normal motion

    speed=1                                     'select speed, 0 to 15
    for servo=0 to 2                             'each servo at a time
    action= servo+move                          'build the action part of the command
    for position= 0 to 250 step 50
    serout tx\flow,baud,[command,position]      'send the command and position
    gosub finished_all                          'finished move?
    next
    next

all_off:                                        'switch all the servos OFF for 2 secs
    for servo= 0 to 2
    action =servo+disable
    serout tx\flow,baud,[command,0]             'send the command and position
    next                                         'speed is ignored

    pause 2000                                  'wait 2 secs
    goto start

Finished:                                       'loop here until either move finished

    serout tx\flow,baud,[report_all,0]          '?status of servo
    serin tx,baud,50,finished,[result]          'loop until status received
    mask=dcd servo                             'calculate the bit mask
    result=result&mask                          'and mask the unwanted bits

```

```
if result=0 then finished
return
```

```
'wait for servo to reach end point
'then return
```

Finished_all:

```
serout tx\flow,baud,[report_all,0]
serin tx,baud,50,finished_all,[result]
mask=%00001111
result=result&mask
if result<>%00001111 then finished_all
return
```

```
'As Finished but looks at all 3 servos

'?status of servo
'loop until status received
'calculate the bit mask
'and mask the unwanted bits
'wait for all servos to reach end point
```