

```

*****
*' Name      : Submarine PIC16F88.BAS          *
*' Author    : Mech307 Group53                *
*' Notice    : Copyright (c) 2011 Brandon McDowall *
*'           : All Rights Reserved            *
*' Date      : 11/12/2011                     *
*' Version   : 11.0 / k                       *
*' Notes     : corresponds with version 11/k of PIC16F917 *
*'           : This pic goes within the sub itself. *
*****
define OSC 8          'establish processor speed
OSCCON.4 = 1 : OSCCON.5 = 1 : OSCCON.6 = 1      'set speed on pic
ANSEL = 0           'turn off analog connections to use digital only

'           76543210
TRISA = %00000000   'sets inputs and outputs
TRISB = %00010001

'below is setting inputs to various pins
recv var PORTB.0    'receiver connection
xsrv var PORTB.1    'x axis servo
ysrv var PORTB.2    'y axis servo
accel var PORTB.4   'accelerometer input
pump1 var PORTB.5   'next four to h bridge
pump2 var PORTB.6
mot1 var PORTA.6    'if one channel is high, the motor/pump will turn one way,
mot2 var PORTA.7    'if other channel high, motor/pump turns other way

xtran var byte     'xaxis value to be recieved
ytran var byte     'yaxis value to be recieved
xpls var word      'x-pulse calculated from xtran
ypls var word      'y-pulse calculated from ytran
ytilt var word     'pulsein from accelerometer
speed var byte     'speed value to be recieved
ballast var byte   'ballast value to be recieved
acc var bit        'accelerometer on/off to be recieved
i var byte         'for counting

low pump1          'initiates pump and motors to low or off
low pump2
low mot1
low mot2
low recv

pause 1000        'standard initial pause to make sure everything starts

xpls=275          'set calibration pulses
ypls=275

for i = 1 to 50   'Send 50 pulses to each servo with a 20ms pause inbetween
pulsout xsrv,xpls 'each pair of pulses.
pulsout ysrv,ypls 'Centers the servos.
pause 20
next i

top              'top of the program

serin recv,0,["sync"],xtran,ytran,speed,ballast,acc      'receive data on recv,
'with mode 0 or T2400. Waits for the word "sync" to be
'recieved before putting the next bytes recieved into
'each of the listed variables (in that order).

if ballast==1 then 'evaluates ballast value for desired result.
low mot1          'Turns off drive motor while ballast motor is on.
low mot2

```

```

    low pump2          'Pump 1 is high and pump 2 is low.
    high pump1
    goto main         'Skip across the remaining motor commands.
endif

if ballast==3 then   'Second ballast state.
    low mot1
    low mot2
    low pump1         'Pump 2 is high and pump 2 is low.
    high pump2
    goto main
endif

low pump1            'If neither above ballast state is met, set both to low.
low pump2

if speed==4 then    'Evaluate speed value.
    low mot1         'Motor 1 is low and motor 2 is high.
    high mot2
    goto main       'Skip remaining motor commands.
endif

if speed==6 then    'Second motor state.
    low mot2         'Motor 1 is high and motor 2 is low.
    high mot1
    goto main
endif

low mot1            'If neither above motor state is met, set both low.
low mot2

main                'Where the motor commands skip to if they are true.

xpls=450-xtran*7/5  'Calcing pulse lengths for servos from transmitted value.
ypls=100+ytran*7/5  'These equations were designed by looking at the max/min
                    'desired pulses and adjusting x/ytran (which was itself
                    'designed to fall into the range of 0-250)

if acc==1 then      'if the accelerometer is enabled, then
    pulsln accel,1,ytilt 'measure pulse length from accelerometer
                        'which has a linear relation with angle.
    ypls=ypls-3/2*(ytilt-1000) 'Adjust y-pulse using angle.
endif

if xpls>450 then xpls=450 'Sets upper and lower limits on pulses to ensure
if xpls<100 then xpls=100 'that no damage is incurred on servos.
if ypls>450 then ypls=450 'Upper limit to 450, lower 100. Center is 275
if ypls<100 then ypls=100

pulsout xsrv,xpls   'Send pulse to pin "xsrv" with period "xpls".
pulsout ysrv,ypls   'Note: The period is measured in increments
                    'which differ based on processor speed. At 8MHz,
                    'the increment is 5us. This means a pulse with a
                    'period of 100 will have a time of 5*100=500us,
                    'or 0.5ms and my upper limit 450 has 5*450=2250us,
                    'or 2.25ms.

                    'Note: The number of pulses being sent to the servos
                    'is very important. Too many Pulses Per Second (PPS)
                    'will result in damage to the servo and too few PPS
                    'will cause the servo to move jerkily if at all.
                    'Aim for about 50 PPS meaning that a pause of 20ms
                    'is needed between each pulse to a servo. As you can
                    'see, there is no 20ms pause in this program, that is
                    'because it is in the program on the PIC16F917 that

```

*'is sending transmissions to this PIC. This program
'pauses at the SERIN command until it receives the
'next set of values that are spaced out at 20ms
'intervals.*

`goto top`

'goes to the top of the program

'By Brandon McDowall