```
{{  ___________________________________________________________________
    *************************************************
    *      BS2 Function Library Test Routines       *
    *               Version 1.0                     *
    *                 3/12/06                       *
    *          Contact: Martin Hebel                *
    *   martin@selmaware.com  or  mhebel@siu.edu    *
    *************************************************

    Test of BS Function Library

    This test reads RC Time of Cap & Photoresistor on A4 on one cog
    and plays frequency of RCTime value on another cog from A0
    for multiprocessor BS2.  Data is passed using global variable.

    Other examples use output and input on the same pins to reduce
    wiring and demonstrate how one cog can be used to test another.

    A WatchDog Cog is also demonstrated.
    Since waiting for pulses, etc, can prevent the cog from
    continuing with this library, another cog monitors to see if
    another cog did not get data in time.
    __________________________________________________________________
}}


CON
  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000

  RC = 4
  Buzzer = 1

VAR     ' Global Variables and Cog Stack Space
  Long RC_Val, stack0[50],stack1[50],stack2[50],stack3[50],stack4[50],stack5[50]
  Long RCVal
  Byte myString[50]
  Long WD1_Flag , WD1_Timer


OBJ

  BS2 : "BS2_Functions"                    ' Create BS2 Object


PUB Start | x
    BS2.start (31,30)                       ' Initialize BS2 Object, Rx and Tx pins for DEBUG
    CogNew(RCTime_Read, @Stack0)            ' Cog to measure RC Time
    CogNew(Sound_Off,   @Stack1)            ' Cog to sound tone based on RC Time
    CogNew(SendData,    @Stack2)            ' Cog to send various data
    CogNew(ReadData,    @Stack3)            ' Cog to read the various data
    CogNew(SendPWM,     @Stack4)            ' Cog dedicated to PWM (0-100%)
    CogNew(WatchDog,    @Stack5)            ' Cog to monitor for failures

    Repeat                                  ' Stay alive

PUB RCTime_Read
  repeat
    dira[RC]~~                                          ' RC output
    outa[RC]:=1                                         ' Set High
    BS2.Pause(10)                                       ' Allow charge
    RCVal := BS2.RCTime(4,1)                            ' Get RC Time
    BS2.Pause(100)


PUB Sound_Off
    dira[Buzzer]~~                                      ' Set output
    repeat
      BS2.Freqout_Set(Buzzer,RCVal)                     ' Use time read in other cog to produce tone

PUB  SendData | Freq
     Freq := 1
     Repeat
       Freq := (Freq + 5) // 1000                       ' Add 5 to freq, modulus 1000
       BS2.Freqout_Set (16,Freq)                        ' produce output at freq

       BS2.Pulsout(17,RCVal * 4)                        ' Make pulse based on RCTime value
       ' BS2.Pulsout(19,100)                            ' -- Enable to test watchdog

       BS2.Pause(500)                                   ' 1/2 second wait

PUB SendPWM | x
    Repeat
       x := (x+1)//101                                  ' Ramp up value 0 to 99
       BS2.pwm_100(23,x, 10)                             ' PWM it on A23

PUB ReadData | x
    Repeat

       BS2.Debug_Str(string(13, "RCTime = "))           ' Show RCTime generated by one cog, and read by
another
       BS2.Debug_Dec(RCVal)

       x := BS2.Freqin(16,500)                          ' Measure frequency on A16
       BS2.Debug_Str(string( "   Freq  = "))
       BS2.Debug_Dec(x)

       X := BS2.PULSIN(17,1)                            ' Measure pulsin based on RC Time
       BS2.Debug_Str(string( "   Pulse  = "))
       BS2.Debug_Dec(x)

       X := BS2.PULSIN_uS(23,1)                         ' Measure time PWM high for
       BS2.Debug_Str(string( "   PWM  = "))
       BS2.Debug_Dec(x)

       WD1_Flag:=1                                      ' Set Watch dog flag
       WD1_Timer := 100                                 ' Set timeout period, re-intialize each pass
       X:= BS2.PULSIN(19,1)                             ' Get/wait for pulse
       IF WD1_Flag == 1                                 ' Check status of Flag
          BS2.Debug_Str(string( "  A19's Pulse  = "))   ' If still 1, show time
          BS2.Debug_Dec(x)
       Else                                             ' If 0, timed out
          BS2.Debug_Str(string( " **** Watch Dog Release ! ****",07))

PUB WatchDog                                            ' WatchDog Cog to monitor other cog
    Repeat
       If WD1_Flag == 1                                 ' Monitor condition?
          WD1_Timer--                                   ' Decrement by 1
          If WD1_Timer == 0                             ' Reach 0?
             WD1_Flag:=0                                ' Yes, clear flag
             BS2.Pulsout(19,30)                         ' Send simulated pulse to nudge cog
        BS2.Pause(10)                                   ' Little time delay

            ' More WD's can be added for other measurement
```