599 Menlo Drive, Suite 100
Rocklin, California 95765, USA
Office: (916) 624-8333
Fax: (916) 624-8003

General: info@parallax.com
Technical: support@parallax.com
Web Site: www.parallax.com
Educational: www.parallax.com/sic

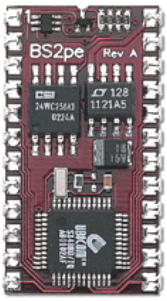
BS2pe Data Logger Application Note

Parts used:

- BS2pe (#BS2PE)
- BS2p24 demo board (#45183) (suggested but not required)
- Parallel LCD (#603-00006)
- DS1822 (#604-00013)
- Push button (#400-00001)
- 10k ohm resistor (#150-01030)
- 4.7k ohm resistor (#150-04720)

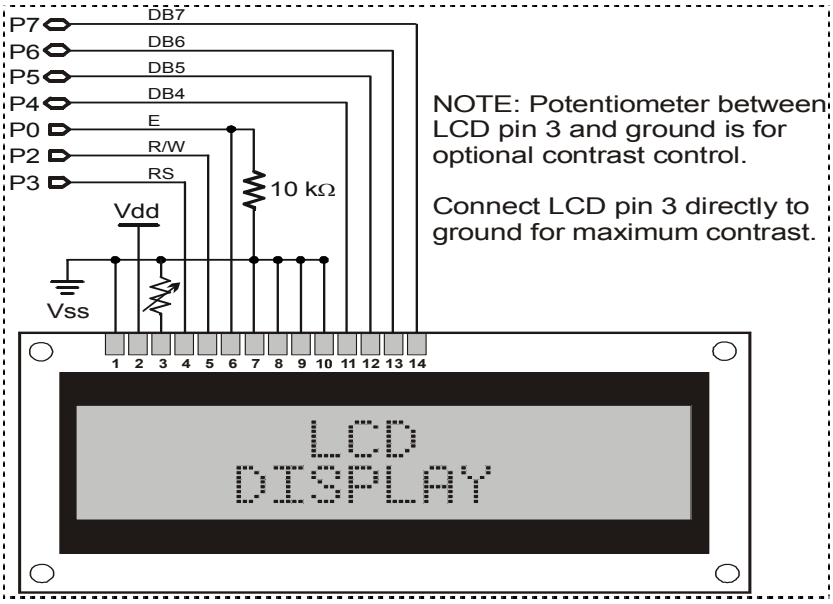
General Information

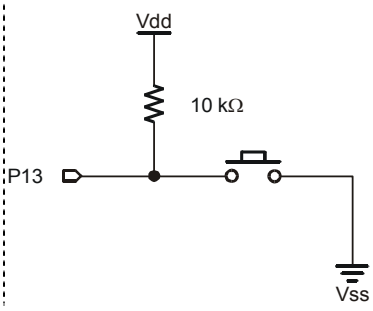
The following code and setup will implement the BS2pe features of data logging, Parallel LCD interface, and expanded EEprom.

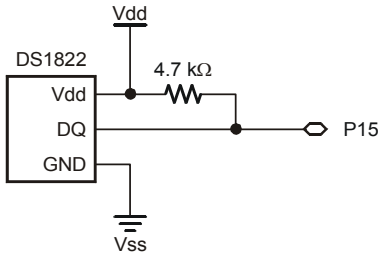


Control from a BASIC Stamp

For easy wiring of the LCD, Parallax would suggest the BS2p24 demo board, but it is not required. The wiring for the LCD does require numerous I/O pins as seen below. If you use the demo board then you will not need to wire the LCD section below.








```
' Temp_lcd_data_demo.bs2
' Purpose... Reads and displays information from a Dallas DS1822,and stores data across memory
blocks.
' {$STAMP BS2pe}
' {$PBASIC 2.5}

' -----[ Declarations ]----------------------------------------------------
OWpin           CON     15
LCDpin              CON       0
NoCmd           CON     $00                     ' No command in LCDOUT
ClrLCD          CON     $01                     ' clear the LCD
CrsrHm          CON     $02                     ' move cursor to home position
CrsLf           CON     $10                      ' move cursor left
CrsRt           CON     $14                      ' move cursor right
DispLf          CON     $18                     ' shift displayed chars left
DispRt          CON     $1C                     ' shift displayed chars right
DDRam           CON     $80                     ' Display Data RAM control
CGRam           CON     $40                     ' Custom character RAM
Line1           CON     $80                     ' DDRAM address of line 1
Line2           CON     $C0                     ' DDRAM address of line 2
UNit_off        CON     $08
Unit_on             CON     $0d

OW_FERst        CON     %0001                   ' Front-End Reset
OW_BERst        CON     %0010                   ' Back-End Reset
OW_BitMode      CON     %0100
OW_HighSpd      CON     %1000

ReadROM         CON     $33                     ' read ID, serial num, CRC
MatchROM        CON     $55                     ' look for specific device
SkipROM         CON     $CC                     ' skip rom (one device)
SearchROM       CON     $F0                     ' search


CnvrtTemp       CON     $44                     ' do temperature conversion
RdScratch       CON     $BE                     ' read scratchpad

NoDevice        CON     %11                     ' no device present
DS1822          CON     $22                     ' device code
DegSym          CON     176


DevCheck        VAR     Nib                     ' device check return ocde
Idx             VAR     Byte                    ' loop counter
RomData         VAR     Byte(8)                 ' ROM data from DS1820
TempIn          VAR     Word                    ' raw temperature
Sign            VAR     tempIn.Bit11            ' 1 = negative temperature
TLo             VAR     tempIn.LowByte
THi             VAR     tempIn.HighByte
TSign           VAR     Bit
TempF           VAR     Word                    ' Fahrenheit
TempC           VAR     Word                    ' Celsius
Run_bit   VAR     Word
Bank                VAR       Byte                          'bank address
Address   VAR       Word                        'address
Readings  VAR     Word                          'Readings
Temp                VAR     Word                          'Work space varablie


' -----[ Initialization ]-------------------------------------------------
'
Initialize:


          PAUSE 500                                     ' let the LCD settle
          LCDCMD LCDpin,%00110000 : PAUSE 5             ' 8-bit mode
          LCDCMD LCDpin,%00110000 : PAUSE 0
          LCDCMD LCDpin,%00110000 : PAUSE 0
          LCDCMD LCDpin,%00100000 : PAUSE 0             ' 4-bit mode
          LCDCMD LCDpin,%00001100 : PAUSE 0             ' no crsr, no blink
          LCDCMD LCDpin,%00000110 : PAUSE 0                     ' inc crsr, no disp shift
          LCDCMD LCDpin,%00101000 : PAUSE 0                     ' 2 Line 5x8 font
```

```
          LCDCMD LCDpin,ClrLCD
pause 1000


' -----[ Main Routine ]----------------------------------------------------
          Bank = 1
Main:


          IF run_bit = 1 then Display_Temperatures
GOSUB Device_Check                          ' look for device
          IF (devCheck <> NoDevice) THEN Get_ROM



No_Device_Found:
          LCDOUT LCDpin,Line1,["   No DS1822"] : PAUSE 5
          LCDOUT LCDpin,Line2,["    Present"]
          PAUSE  1500
          LCDCMD LCDpin,ClrLCD  : PAUSE 0
          LCDOUT LCDpin,Line1,["Insert Ds1822"] : PAUSE 5
          LCDOUT LCDpin,Line2,["and reset stamp"]
          PAUSE 1500
          LCDCMD LCDpin,ClrLCD  : PAUSE 0
goto main

Get_ROM:
          OWOUT OWpin,OW_FERst,[ReadROM]                ' send Read ROM command
          OWIN  OWpin,OW_BERst,[STR romData\8]          ' read serial number & CRC
          IF (romData(0) = DS1822) THEN Show_Data
          LCDCMD LCDpin,ClrLCD  : PAUSE 0
          LCDOUT LCDpin,Line1,["Installed device"]
          PAUSE 5
          LCDOUT LCDpin,Line2,["is not DS1822"]    : PAUSE 5
          LCDOUT LCDpin,NoCmd,["Code = ",HEX2 romData(0)]
          PAUSE 1500
goto main

Show_Data:

          LCDCMD LCDpin,ClrLCD  : PAUSE 0
          LCDOUT LCDpin,Line1,["DS1822 Data"]
          PAUSE 1500
          LCDCMD LCDpin,ClrLCD  : PAUSE 0
          LCDOUT LCDpin,Line1,["Serial Number : "]    : PAUSE 5
          FOR idx = 6 TO 1
          LCDOUT LCDpin,Line2,[HEX2 romData(idx)]
NEXT
          PAUSE 1500
          LCDCMD LCDpin,ClrLCD  : PAUSE 0
          LCDOUT LCDpin,Line1,["Checksum : ",HEX2 romData(7)]
          PAUSE 2000



Display_Temperatures:

          run_bit = 1

GOSUB Get_Temp

          LCDCMD LCDpin,ClrLCD  : PAUSE 0
          LCDOUT LCDpin,Line1,["Temp ","F:", SDEC tempF]     : PAUSE 0

skip_over:

          readings = tempF
          IF in13 = 0 THEN full
GOSUB record

GOTO Main


' -----[ Subroutines ]------------------------------------------------------
'
' This subroutine checks to see if any 1-Wire devices are present on the
' bus.  It does NOT search for ROM codes
'
Device_Check:

          devCheck = 0
          OWOUT OWpin,OW_FERst,[SearchROM]              ' reset and start search
          OWIN  OWpin,OW_BitMode,[devCheck.Bit1,devCheck.Bit0]
RETURN
```

```
Get_Temp:
          FOR temp = 1 TO 250
          OWOUT OWpin,OW_FERst,[SkipROM,CnvrtTemp]      ' send conversion command
          PAUSE 1                                       ' give it some time

NEXT

          OWOUT OWpin,OW_FERst,[SkipROM,RdScratch]      ' go get the temperature
          OWIN  OWpin,OW_BERst,[tLo,tHi]
          tSign = sign                                  ' save sign bit
          tempC = tempIn
          tempC = tempC >> 4                            ' round to whole degrees

          IF (tSign = 0) THEN NoNegC
          tempC = tempC | $FF00                         ' extend sign bits for negs

NoNegC:
           tempF = tempC */ $01CD                        ' multiply by 1.8
           IF tSign = 0 THEN NoNegF                      ' if neg, extend sign bits
           tempF = tempF | $FF00

NoNegF:
            tempF = tempF + 32                            ' finish C -> F conversion
RETURN

Full:

          FOR bank = 1 TO 15
          FOR address = 0 TO 2048 step 2
          STORE bank
          READ address ,readings.lowbyte
          READ address + 1,readings.highbyte

          LCDCMD LCDpin,ClrLCD  : PAUSE 0
          LCDOUT LCDpin,Line1,["Reading: ",dec address/2]   : PAUSE 0
          LCDOUT LCDpin,Line2,["Bank:",dec bank,"Temp ","F:", SDEC tempF]     : PAUSE 0
          PAUSE 1000 ' lessen this pause to increase rate of display
          NEXT
          NEXT

Sleeping:
          SLEEP 3
          LCDCMD LCDpin,unit_off  : PAUSE 0
          IF IN12 = 1 THEN sleeping
          LCDCMD LCDpin,unit_on  : PAUSE 0
GOTO sleeping



Record:
          PAUSE 1000 'Adjust to vary the rate of storage
          If address < 2040 THEN no_up
          bank = bank +1
          address = 0
No_up:

          'DEBUG 0,cr,"record: ",? bank,? address,? readings," Low ",dec readings.lowbyte,"  High
",dec readings.highbyte
          STORE bank
          WRITE address ,readings.lowbyte
          WRITE address + 1,readings.highbyte
          address = address + 2
          IF bank = 15 AND address = 2040 THEN filled
          return
filled:

          LCDCMD LCDpin,ClrLCD  : PAUSE 0
          LCDOUT LCDpin,Line1,["Full"]    : PAUSE 5
STOP
```