Dios - Beta Brite

Dios to Beta Brite Interface

Iwas walking through one of our local warehouse shopping stores and came across a couple of these LED signs. Ihad seen them before for almost $200 but they were on sale for $60 each. I picked up two of them.

These were branded Beta-Brite but Iknow they are sold under other names as well. They come with a CD with a very basic piece of software. I soon decided that I had to create an interface to the Dios.

The Beta-Brite units came with a DB9 cable with an RS232 interface. After a bit of research Ifound the baud rate to be 9600 at 7 bits even parity.

The built-in UART and the standard serial routines on the Dios are 8 bits no parity so I had to come up with a routine to create the bits needed.

I wanted to keep the interface as simple as possible so I decided to send raw TTL data to the Beta-Brite. The Beta-Brite supports this but the output needs to be inverted. Normally this is taken care of by the RS232 Driver. Icreated a routine to take care of sending characters in the inverted form.

Now all that was needed was the actual physical connection. First off you will need a male DB9 connector. You can use the straight PC mount connectors from Jameco #111691CC or the right angle PC mount #104942CC. Of course you may use any male DB9 you wish as these are just suggestions.

For this project I will use the right angle connector. We will make a few modifications to the connector so it can be plugged into the Dios WorkBoard.

Refer to Figure 1 and cut the 2 large prongs as well as the 4 inside pins. This will allow you to plug the connector into the Dios WorkBoard as shown in Figure 2.

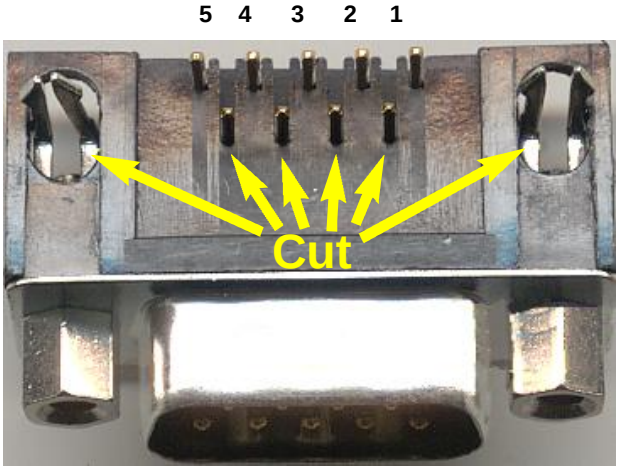
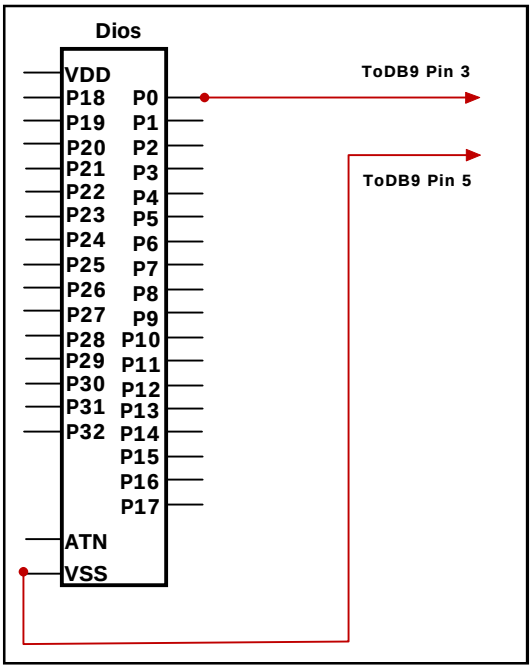


Figure 1



Schematic 1

Hookup

Using Schematic 1 and Figure 2 hookup is a breeze. For a more permanent connection you can solder the leads to the appropriate connector.

Software

This is where the real work gets done. Lets look at the lowest level functions then work our way up.

The function **sendchar7e** takes an 8bit character and converts it to 7-bits, even parity. For each character send the function

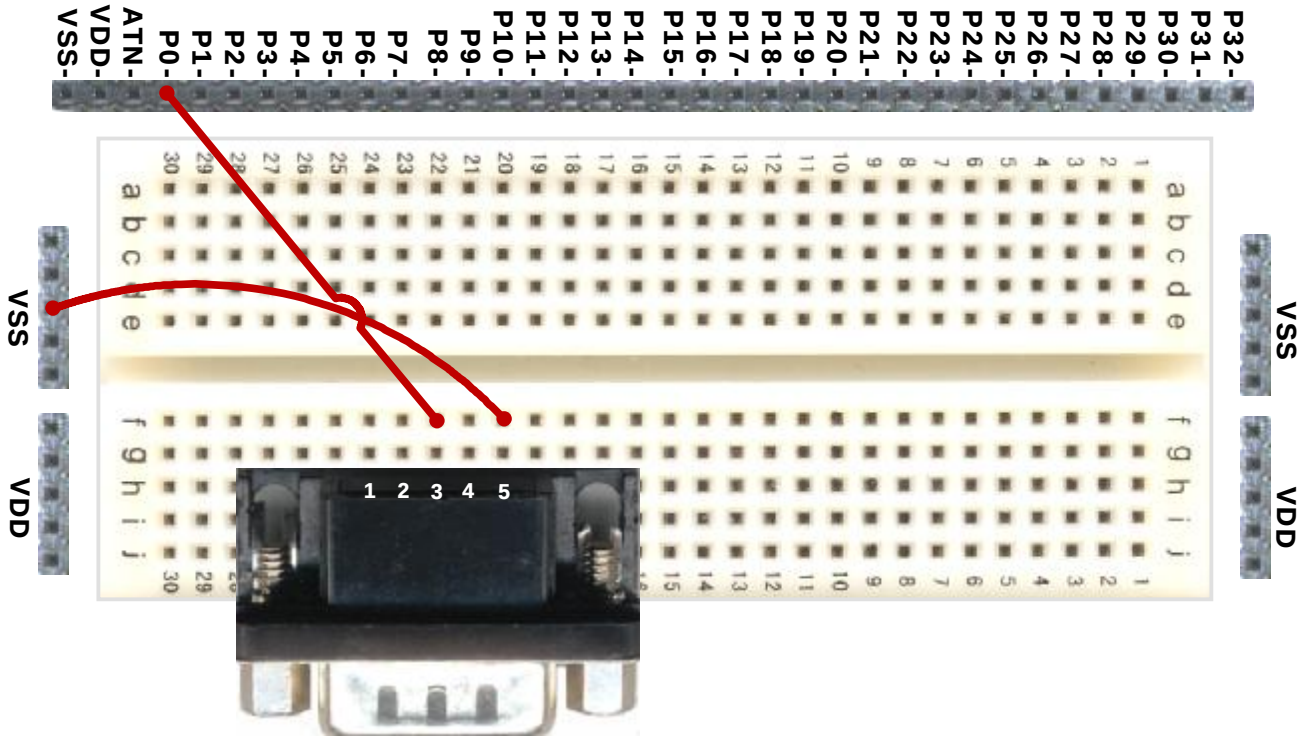


Figure 2 Dios WorkBoard Layout

seroutrawi is called. This function sends the modified character out to the Beta-Bright.

The function **sendstring7e** was created to send whole strings to the Beta-Bright. It simply scans through the string making calls to **sendchar7e**.

Before we move up to the main demo routine lets take a look at some of the support routines that are called.

bbsettime
This function takes single variable and breaks it down and sets the clock in the Beta-Brite display. The time variable should contain a value in 24 hour format. For example 204 would be 02:04 AM. 1404 would be 02:04 PM.

Note that some displays do not have a real time clock so this command may not be used.

bbsendtext
This function breaks down the text sent into the Beta-Brite protocol.

bbready
This function tells the display that we are ready to send data.

Now lets take a look at the main demo routine. The most important part here are the string constants that we have set up to simplify the actual call the the bbquicktext function. Simply include the appropriate control in the string using the + operator.

Example:

bbsendtext(0,"Hello "+ColorDGreen+"World")

Note that the time can be inserted into the text by using the time control option as shown in our example.

Please note that not all control sequences are compatible with each other. Also some options may not be supported by your Beta-Brite display.

The best thing to do is to experiment with the bbsendtext call in the main demo routine.

```
'Beta-Brite Interface Demo

'Sends data out a serial port
func main()

   output 0
   low 0
   sersetup baud,SBAUD9600
   pause 1000


   bbready(0)
   sendchar7e 0,2
   sendstring7e 0,"A0"
   sendstring7e 0,"Hello World"
   sendchar7e 0,4


endfunc



'-----------------------------------------------------------
'Tell the display we are ready to send data
'-----------------------------------------------------------
func bbready(port)
   sendchar7e port,0
   sendchar7e port,0
   sendchar7e port,0
   sendchar7e port,0
   sendchar7e port,0

   sendchar7e port,1
   sendchar7e port,1
   sendchar7e port,1
   sendchar7e port,1
   sendchar7e port,1
   sendchar7e port,1


   sendchar7e port,'Z' '5A
   sendstring7e port,"00" 'Device Address
endfunc




'-----------------------------------------------------------
'Send a string of data at 9600 7e2
'-----------------------------------------------------------
func sendstring7e(port,addr)
  dim x,y
  x = 0
loop:
    getbyte addr + x,y
    if y = 0 then exit 0
    sendchar7e port,y
    debug y
 x = x + 1
 goto loop
endfunc


'-----------------------------------------------------------
'Send a byte of data at 9600 7e2
'-----------------------------------------------------------
func sendchar7e(port,dat)
   dim x,b
   dim ep
   ep = 0
   for x = 0 to 6
     if dat.bit(x) = 1 then
       ep = ep + 1
     endif
   next
   dat.bit(7) = ep.bit(0)
   seroutrawi port, dat
   pauseus 108
endfunc


'-----------------------------------------------
'Raw output to IO port.
'No need for RS232 driver
'-----------------------------------------------
startasmcommand seroutrawi(exp,exp)
   getexpression OPP1
   getexpression OPP2

   movffOPP1,ACUMEL

soriasyxmt:
        gosub setpin
        gosub sorifull
        movlw   08
        movwf   BITCOUNTER
soriasyxmt1:
        rrcf     OPP2,f
        btfsc   STATUS,C
        goto    soriasyxmt2
        gosub setpin
        goto    soriasyxmt3
soriasyxmt2:
        call resetpin
soriasyxmt3:
        nop
        nop
        nop
        nop ; Used to keep timing the same as getchar
        nop
        nop
        nop
        nop
        gosub sorifull
        decfsz  BITCOUNTER,f
        goto    soriasyxmt1
        gosub resetpin
        gosub sorifull
        gosub sorifull
        exit

sorifull:
        movff   SOFTBAUDH,DELAYCOUNTH
        incf    DELAYCOUNTH,f
        movff   SOFTBAUDL,DELAYCOUNTL
        incf    DELAYCOUNTL,f
        gosub    sorivdly
        return


sorivdly:
        decfsz  DELAYCOUNTL,f
        goto    sorivdly
        dcfsnz  DELAYCOUNTH,f
        return
        clrf DELAYCOUNTL
        goto sorivdly

endasmcommand
```

Related Products

Kronos Robotics . . . . . http://kronosrobotics.com

DiosPro 40 . . . . . http://kronosrobotics.com/xcart/customer/product.php?productid=16428
Dios WorkBoard . . . . . http://kronosrobotics.com/xcart/customer/product.php?productid=16452

9 Pin Cable . . . . . http://kronosrobotics.com/xcart/customer/product.php?productid=16259

Jameco . . . . . http://Jameco.com