Here is a schematic of the circuit we will build.

Cut off the connector from the end of your 5V supply and strip the insulation off for one half inch. Tin the striped ends with solder.
Plug in your supply and measure the ends with your multimeter. Mark the positive end. Place the positive end into the red binding post with a red jumper wire. Place the negative end into the black binding post with a red jumper wire as shown below.
Next take your switch and cut off one of the silver mounting posts as shown below.

Next we will start the wiring of the proto board. Install the two red power wires and switch as shown. Then install the two orange jumper wires. See below.
Next we can install the green LED and the 240ohm ballast resistor as shown below.
Now we can install the red LED with its ballast resistor. See below.
We can now install the two capacitors on the power buss. A 0.1uf and a 47uf. Pay attention to the polarity on the 47uf capacitor. The long lead is positive and the negative side is marked with a stripe on the side of the body.

The last step is to secure your switch on to the board with super glue. This will keep the switch from popping off of the board during opera.
Here is the schematic for our PIC microprocessor demo.
This is the code used for the demo setup.

' PicBasic program to demonstrate operation of an LCD in 4-bit mode
' It also flashes an LED and beeps a Piezo element
' LCD should be connected as follows:
'   LCD     PIC
'   DB4     PortA.0
'   DB5     PortA.1
'   DB6     PortA.2
'   DB7     PortA.3
'   RS      PortA.4 (add 4.7K pullup resistor to 5 volts)
'   E       PortB.3
'   RW      Ground
'   Vdd     5 volts
'   Vss     Ground
'   Vo      10K-20K potentiometer
'   DB0-3   No connect

 Define ADC_BITS   10      ' Set number of bits in result
 Define ADC_CLOCK  3       ' Set clock source (3=rc)
 Define ADC_SAMPLEUS 50     ' Set sampling time in uS

 OSCCON = %01100010  ' Set up internal oscillator
 ADCON0 = %00010011' Set up the A/D channel and references
 ADCON1 = %01111111' Set up the I/O pins as digital or analog

cnt var byte
TRISB.0 = 0 ' Set PORTB, pin 0 to an output
Pause 500  ' Wait 500mS for LCD to startup

Top:
  Lcdout $fe, 1   ' Clear LCD screen
  Pause 500      ' Wait  .5 second
  lcdout $fe,$80 ' Move cursor to the begening of the first line
  Lcdout "Junior Design" ' DisplayJunior Design
  Pause 500      ' Wait  .5 second
Lcdout $fe,$c0  ' Move cursor to the begening of the second line
Lcdout "PIC Lecture"  'Display PIC Lecture
Pause 500       ' Wait .5 second
gosub alert
Goto Top       ' Do it forever

Alert:
    for cnt = 1 to 4
    freqout portb.4,75,2000 'send 2KHz tone on Portb.4
toggle portb.0  'Toggle port B.0 output state or Blink LED
    pause 75                ' Pause 75mS
    next cnt
return

end
Here is the schematic for our PIC microprocessor A/D demo.

This is the code used for the second demo setup.

' PicBasic program to demonstrate operation of an LCD in 4-bit mode
' using the A/D input on portb.0 as display data
' LCD should be connected as follows:
' LCD     PIC
' DB4     PortA.0
' DB5     PortA.1
' DB6     PortA.2
' DB7     PortA.3
' RS      PortA.4 (add 10K pull up resistor to 5 volts)
' E       PortB.3
' RW      Ground
' Vdd     5 volts
' Vss     Ground
' Vo      10K-20K potentiometer
' DB0-3  No connect

Define  ADC_BITS        10      ' Set number of bits in result
Define  ADC_CLOCK       3      ' Set clock source (3=rc)
Define  ADC_SAMPLEUS    50     ' Set sampling time in uS

OSCCON = %01100010  ' Set up internal oscillator
ADCON0 = %00010011 ' Set up the A/D channel and references
ADCON1 = %01101111 ' Set up the I/O pins as digital or analog
ADCON2 = %10111110 ' Set up A/D result format and acquisition

time

volts VAR WORD       ' Variable to store A/D result in
dec0  VAR BYTE      ' Variable to store ones digit
ten0  VAR BYTE      ' Variable to store tenths digit

TRISB.0 = 1 ' Set PORTB, pin 0 to an input

Pause 500       ' Wait 500mS for LCD to startup
LCDOut $fe, 1   ' Clear LCD screen
Pause 1       ' Wait 1 millisecond
LCDOut $fe,$80  ' Move cursor to the beginning of the first line
LCDOut "A/D Demo using"  ' Display
LCDOut $fe,$c0  ' Move cursor to the beginning of the second line
LCDOut "PIC18F1320"  'Display PIC Lecture
Pause 2000      ' Wait 2 seconds

loop:
    ADCIN 4,volts       ' Read in the A/D value
    dec0 = volts / 204    ' Calculate the ones digit
    ten0 = volts // 204 ' Calculate tenths digit
    ten0 = ten0 / 21     ' Finish calculating tenths
    LCDOut $fe, 1       ' Clear LCD screen
    LCDOut $fe,$80      ' Move cursor to the beginning of the first line
    LCDOut DEC1(dec0),","DEC1(ten0),"Volts"," ",dec4 volts,"DEC"
    LCDOut $fe,$c0      ' Move cursor to the beginning of the second line
    'lcdout rep 255(volts/51)
    LCDOut BIN10 volts,"Bin ",HEX3 volts,"Hex" 'Display PIC Lecture
    Pause 100       ' Wait 0.1 second

    GoTo loop       ' Do it forever

End
Ramsey’s Ten Commandments of Good PC-Board Soldering

1. If the soldering iron tip is covered with burned rosin, it cannot heat your connection very well.
2. If you heat only the wire and not the wire and PC trace together, a cold, bad connection is likely.
3. If your soldering tip is big enough to bridge two adjoining connections, it probably will!
4. Dirty, grubby solder will contribute to dirty, grubby connections.
5. Any use of acid core (Plumbers!) solder in electronics work will destroy everything… DON”T USE IT!!
6. A connection in a large area of PC-board copper requires more heat than one pin of an IC.
7. If your connection looks dull or brittle, it’s no good.
8. If your connection looks like a ball instead of a shiny cone, it’s no good.
9. Thin fresh shiny rosin core solder is easier to use for PC-board work than thicker “hardware store” solder.
10. Pre-tin any stranded hookup wires leading in and out of your PB-board kit project. It will prevent problems later.