

## Errata for *What's a Microcontroller? v2.2* (#28123)

**Page 140:** In *What's a Microcontroller Parts Kits* purchased after January 10, 2007, the potentiometer may look different from the one shown in the text. The two styles are functionally the same, with the same terminals. Here are alternative drawings:

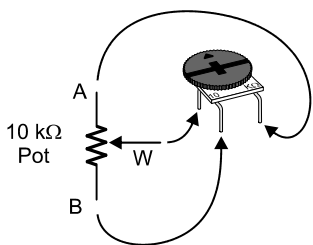
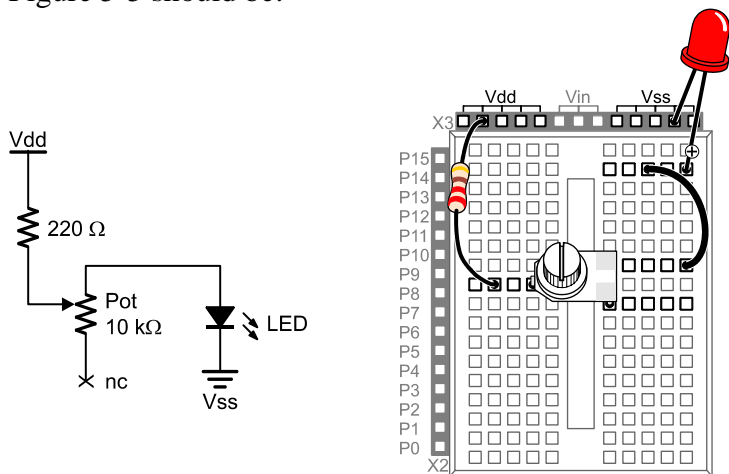


Figure 5-3



Figure 5-4 (left)

**Page 142:** In the text, the circuits represented by the schematic and wiring diagram in Figure 5-5 are not identical though they are functionally equivalent. To be consistent with each other, Figure 5-5 should be:



**Page 151:** Paragraph 1 should read:

The example program in Activity #2 measured the RC decay time by checking whether  $IN7 = 0$  every 100 ms, and it kept track of how many times it had to check. When  $IN7$  changed from 1 to 0, it indicated that the capacitor's voltage decayed below 1.4 V. The result when the program was done polling was that  $IN7$  equaled the number of tenths of a second it took for the capacitor's voltage to decay below 1.4 V.

This next example program uses a PBASIC command called **RCTIME** that makes the BASIC Stamp measure RC decay in terms of 2  $\mu$ s units. So, instead of tenths of a second, the result **RCTIME 7, 1, time** stores in the **time** variable is the number of two-millionths of a second that it takes for the capacitor's voltage to decay below 1.4 V. Since the **RCTIME** command has such fine measurement units, you can reduce the capacitor size from 3300  $\mu$ F to 0.1 or even 0.01  $\mu$ F, and still get time measurements that indicate to the resistor's value. Since the resistance between the potentiometer's A and W terminals changes as you turn the knob, the **RCTIME** measurement will give you a time measurement, which is a number that indicates the position of the potentiometer's knob.

**Page 183:** The first full paragraph should read (changed sentence is bolded):

Unless you tell it to make a different kind of comparison the **LOOKDOWN** command checks to see if a value is equal to an entry in the list. You can also check to see if a value is greater than, less than, or equal to, etc. **For example, to search for an entry that the value variable is less than or equal to, use the <= operator just before the first bracket that starts the list.**

In other words, in this context the operator returns the index of the first value in the list that makes the statement in the instruction true.

**Page 217:** For Solution 2, also add:

```
Index = 5
```

...to the Initialization section of your program.

**Page 228:** In the top row of Figure 8-3, the number 1217.5 should be 2217.5.

**Page 265:** In the second checklist instruction, P3 should be P14.

**Page 277:** The program below was printed with several lines omitted. The complete correct listing is included below, with the added lines in **bold**.

```
' -----[ Title ]-----
' What's a Microcontroller - Ch9Prj01_PhotoControlledDigitalPot.bs2
' Corrected version of program on page 277 of v2.2 text.
' Update digital pot's tap based on photoresistor reading.
' Based on TerminalControlledDigitalPot.bs2

' {$STAMP BS2}
' {$PBASIC 2.5}
DEBUG "Program Running!"

' -----[ Declarations and Initialization ]-----

PhotoPin      CON      2           ' Photoresistor on pin P2
UdPin         CON      5           ' Set values of I/O pins
ClkPin        CON      6           ' connected to CLK and U/D.
DelayPulses   CON      10          ' Delay to observe LED fade.
DelayReader    CON      2000

counter       VAR      Byte        ' Counter for FOR...NEXT.
oldTapSetting VAR      Byte        ' Previous tap setting.
newTapSetting VAR      Byte        ' New tap setting.
lightReading  VAR      Word        ' reading from photoresistor
```

```

oldTapSetting = 0           ' Initialize new and old
newTapSetting = 0           ' tap settings to zero.

LOW UdPin                   ' Set U/D pin for Down.
FOR counter = 0 TO 128      ' Set tap to lowest position.
  PULSOUT 6,5
  PAUSE 1
NEXT

' -----[ Main Routine ]-----

DO:
  GOSUB Read_Photoristor
  lightReading = lightReading MAX 127      ' Constrain to max of 127
  newTapSetting = lightReading
  GOSUB Set_Ud_Pin                       ' Set U/D pin for up/down.
  IF newTapSetting <> oldTapSetting THEN    ' No change if light level steady
    GOSUB Pulse_Clk_pin                   ' Deliver pulses.
  ENDIF
LOOP

' -----[ Subroutines ]-----

Set_Ud_Pin:                  ' Examine new and old tap values
                             ' to decide value of U/D pin.

  IF newTapSetting > oldTapSetting THEN
    HIGH UdPin
    oldTapSetting = oldTapSetting + 1
  ELSEIF newTapSetting < oldTapSetting THEN
    LOW UdPin
    oldTapSetting = oldTapSetting - 1
  ENDIF
  RETURN

Pulse_Clk_pin:              ' Deliver pulses from old to
  FOR counter = oldTapSetting TO newTapSetting ' new values.
    PULSOUT ClkPin, 1
    PAUSE DelayPulses
  NEXT
  oldTapSetting = newTapSetting           ' Keep track of new and old
  RETURN                                  ' tapSetting values.

Read_Photoristor:
  HIGH PhotoPin
  PAUSE 100
  RCTIME PhotoPin, 1, lightReading
  RETURN

```