



Column #126, October 2005 by Jon Williams:

## Tricks and Treats with LEDs

*It's finally here: October – time to build (okay, finish building) our Halloween displays, whether they're just in our yard or in a full-blown, professional haunt. Nothing adds spooky ambience like a candle in a darkened room, but candles can be dangerous – unless you build them yourself and substitute the coolness of LEDs for the heat of an open flame.*

A few months ago my friend Brian Bayliss – who runs the Methodz of Madness Halloween prop building forum – asked me if I could help him create a set of faux candles using a Prop-1 (BS1) controller. “Sure!” I quickly exclaimed, and in short order I e-mailed a bit of code for him to test. After a few rounds of fine-tuning, Brian was pretty happy.

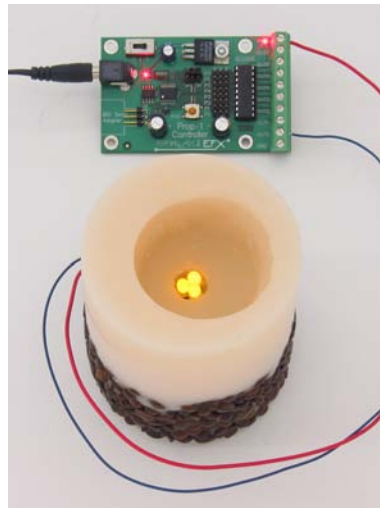
But... it didn't look quite right. Nobody has ever accused the BS1 of being a speed demon, so I had to keep things simple. And that's exactly what I did: I tumbled a random number and spit it out to the pins (that were connected to LEDs). While it looked pretty good, there was something distinctly “digital” about it. And no wonder, all I was doing was turning LEDs on and off.

Then my colleague John Barrowman stepped in and added a capacitor to each “wick” output. Bam! Now we have something. The capacitor doesn't change the turn-on time of the circuit,

but it does cause the LED to fade off instead of simply snapping off. It's amazing how something so simple can have such a dramatic visual impact.

Back in July, John, Jen Jacobs (our graphic designer), and I attended the Midwest Haunters Convention in Columbus, Ohio. It was great time with lots of interesting people and the opportunity to show some amazingly skilled prop builders how simple controllers like the BS1 can add real life to their displays. The booth next to us had a hanging man that was animated, but... not very "animated." John asked the owner if we could add a controller to the prop and he agreed. While John wired up the Prop-1, I wrote a simple program and after about 15 minutes the hanging man prop was thrashing around as if he'd actually been lynched. It was fun watching the crowd gather, and the prop owner was smiling for the rest of the show (Interestingly, he had purchased a few Prop-1 controllers from Parallax, but hadn't learned how to use them – this demo showed him what one could do with very simple code).

Our booth didn't have any thrashing props, but one thing we did have was a huge hit: our faux candles (we had two sets of six). In fact, some folks were irritated that we weren't offering them for sale! Figure 126.1 gives you an idea of what a completed candle looks like.



**Figure 126.1 Electronic Candle with Prop-1 Controller**

The candles require very little effort to put together, and I'm going to give you all the details – mechanical and electrical. Now, if you're skeptical about how good an electronic candle can look, perhaps this will sway you: on three separate occasions we were told we could not have

an open flame on the trade show floor! Our response was to put a hand over the faux candle – in a manner that one could never do with the real thing – and then ask, “What open flame?”

### **Waxing Digital**

Okay, how do we get the electronic wicks into the candles? It’s really pretty easy, and we’ll do that first so that we can fit the electronics to it. Run down to your local dollar store and buy a bunch of cheap candles. We tend to go with the short, stubby kind, and find that lighter colors work best with yellow LEDs (Note: While white LEDs seem like a good idea, we’ve tried them and didn’t care for the results – and they’re really expensive, so don’t bother with them). When you get home, setup your candles in a safe location and light them. What we want to do is allow the wick to burn about a one-inch deep well into the top of the candle. When that’s complete (it may take a couple hours), extinguish the candles and let them cool completely.

Now take a half-inch drill bit and bore a hole down the center of the body. After the centerline holes are drilled, add a small hole between the wick channel and the outside edge of the body to hold the capacitor. You may also want to cut a small notch into the bottom of the candle to route wiring.

### **Electronic Wicks**

Now that we have candle bodies, let’s build some electronic wicks to light them. The circuit is quite straightforward: we drive three yellow (you could mix in an orange too) LEDs using a ULN2003 (so we don’t overburden the BASIC Stamp). Figure 126.2 shows the circuit.

Depending on the LEDs and power supply you use, you may want to adjust the resistor value. We’ve found that calculating 20 mA through the LEDs works well, and with three LEDs per candle body gives very nice results. Of course, if your LEDs can take more continuous current and you’d like things to be brighter, go ahead and reduce the size of the resistor. The ULN2003 can take it; it’s able to sink 500 mA per channel.

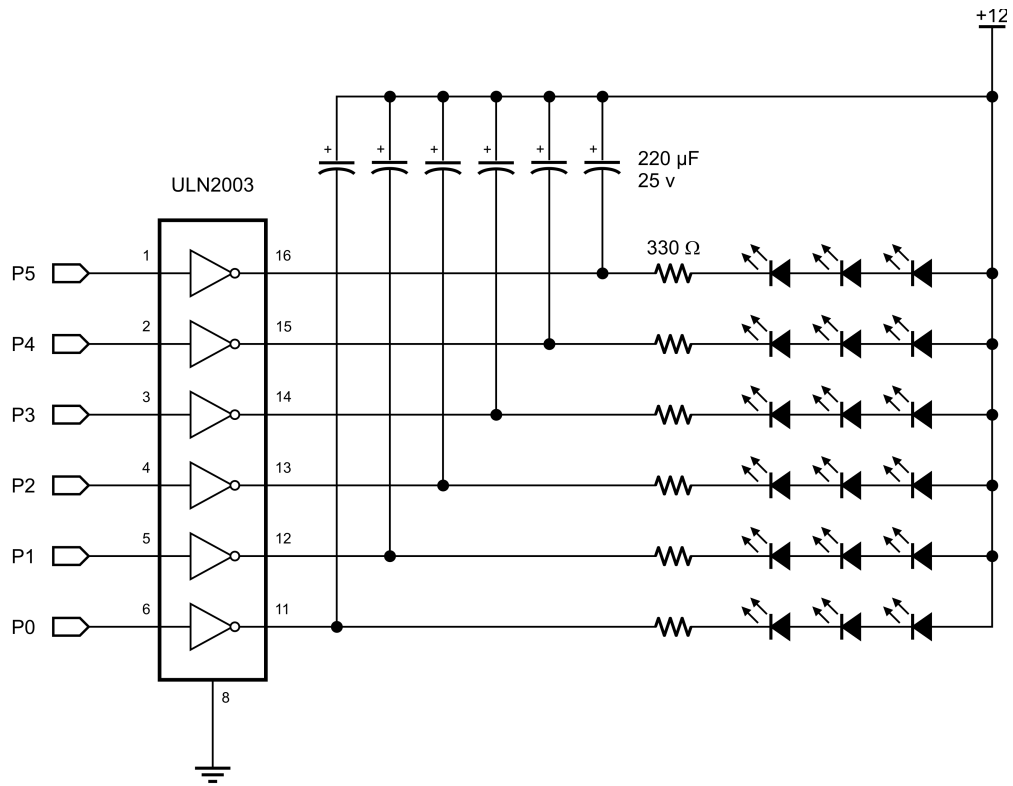


Figure 126.2 Electronic Candle Schematic

Note that the capacitors are rated for twice the voltage of the supply; this is important. And do make sure that you connect the capacitors properly as they are polarized. If you don't you'll get a very digital look to the candle, followed by a loud crack! of the capacitor popping. I learned this the hard way after a late-night soldering session. What you'll want to do is insert the LEDs into the body so that they just pop up from the well. Then add the capacitor into the control lines so that it can be pushed into its mount hole and the wires routed through the notch, as shown in Figure 126.3. You can use a bit of duct tape to hold everything in place.



**Figure 126.3 Candle Capacitor Mounting**

### Candle Code

Okay, the hard part is done – now for the really easy part: the software. This version is a bit of an update from what I did for Brian a few months ago, but no more difficult. In fact, the code is so simple we can look at it in one shot:

```

Reset:
  DIRS = %00111111

Main:
  RANDOM flicker

Check_Dark:
  wicks = flicker & %00111111
  IF wicks = %000000 THEN Main

Flame_On:
  Candles = wicks
  rate = flickVar & %00001111 + FlickBase
  PAUSE rate
  GOTO Main

```

The program starts by making the candle pins outputs. At the top of the main loop, the variable flicker is tossed with RANDOM; this is what will drive the candle outputs. Before we do that though, we need to ensure that at least one “wick” is lit. Generally we’re going to group the candles together, and if all are out – even very briefly – it doesn’t look quite right.

The randomized value (flicker) is masked to get just the lower six bits. If these bits are all zeros then the program is directed back to the top to stir the random number generator. If not zero, the value is moved to the outputs and a short delay is created.

That delay is randomized as well. The variable called flickVar is actually aliased as the upper byte of flicker (a Word). We don’t need a lot of variability with the flicker delay, so only the lower four bits are used. This gets added to a base delay and is used by PAUSE. We do all this through a separate variable (rate) so that we don’t modify the random value flicker.

Then we start over. It’s as simple as that. And honestly, still photos do no justice to this project – you really have to see it in action to appreciate how cool it looks. Even if you have to spend a couple minutes breadboarding the project (you can do it on the PDB) to convince yourself, the time will be worth it.

One thing you may notice is a bit of low-bit to high-bit motion in the pattern. This is a result of the LFSR (linear feedback shift register) used by RANDOM. Don’t worry, this is easily dealt with by arranging the candles so that they’re not physically placed in the same order as their output pin connections.

### **Blowin’ in the Wind**

Another one of John’s good ideas for this project was to add a sensor input so that the program could simulate wind. Perhaps you’ve got a mat switch or PIR sensor detecting visitors in your haunt; you could use this to affect the candles as if the air around them had been disturbed.

And what you may find – if you’re anything like me – is that a lot of tweaking takes place to get the candles to look just right. Let’s fix that with a potentiometer circuit. Figure 126.4 shows the added controls. Note that the pushbutton circuit could be changed to an active-high PIR sensor for non-contact people detection.

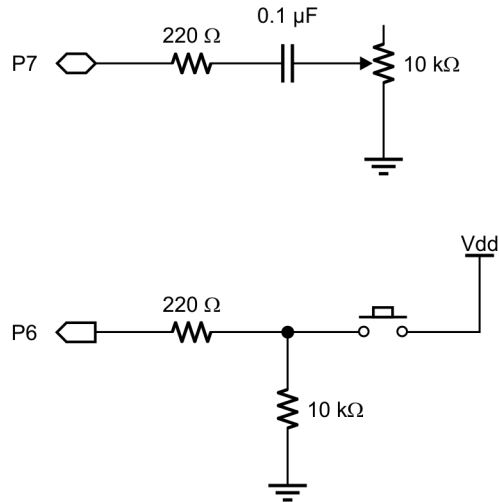


Figure 126.4 Candle Control Schematic

```

Main:
  RANDOM flicker

Check_Dark:
  wicks = flicker & %00111111
  IF wicks = %000000 THEN Main

Get_Rate:
  POT RateCtrl, 100, rate

No_Wind:
  IF Windy = Yes THEN Has_Wind
  rate = rate / 25 + 5
  GOTO Flame_On

Has_Wind:
  rate = rate / 10 + 25

Flame_On:
  Candles = wicks
  PAUSE rate
  GOTO Main
    
```

This version of the program starts out the same as the last, and then drops into a subroutine called `Get_Rate` to read the potentiometer. Remember that POT (BS1 only) is different from RCTIME that we use on the BS2 – it takes care of charging and discharging the capacitor, hence the difference in the RC circuit used. POT also returns a byte value, which is fine for this application because we don't need big numbers.

Once the raw rate value has been read from the potentiometer, the “wind” input is checked. If there is no wind the flicker rate is scaled to 5 - 15. This produces a nice rate to make the candle look like it's sitting in a still room. If the wind input is active, then the rate value is scaled to 35 - 60. The longer timing allows the LED capacitor to discharge more completely and creates the illusion that air has been moved around the flame, and that the flame was nearly extinguished.

Okay, how about a bit more realism? Yes, it can be done – if you only need one or two candles (per controller). The idea is to use three independent LED circuits for each wick. This will give movement to the flame, in addition to more modulation of the candle brightness. Use a 510 ohm resistor for the LED (Figure 126.5), and remember that you have to mount three capacitors on the bottom of the candle and account for four wires.

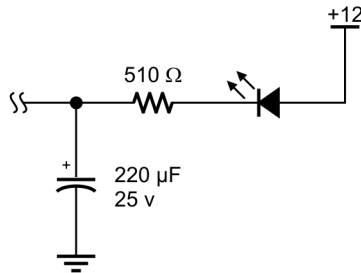


Figure 126.5 Independent LED Circuit

The code doesn't change dramatically. What we have to modify for this version is the “dark” check; one for each candle.

```

Main:
  RANDOM flicker

Check_Dark1:
  wicks = flicker & %00000111
  IF wicks > %000000 THEN Check_Dark2
    flicker = flicker | %000010

Check_Dark2:
  wicks = flicker & %00111000
  IF wicks > %000000 THEN Get_Rate
    flicker = flicker | %010000

Get_Rate:
  POT RateCtrl, 100, rate

No_Wind:
  IF Windy = Yes THEN Has_Wind
    rate = rate / 25 + 5
    GOTO Flame_On

Has_Wind:
  rate = rate / 10 + 25

Flame_On:
  Candles = flicker & %00111111
  PAUSE rate
  GOTO Main

```

You can see the independent darkness checks at Check\_Dark1 and Check\_Dark2. If all three LEDs for either candle are out, these routines light one LED to prevent the candle from going completely dark.

One final reminder before I close: Component values will vary, so be sure to run the Pot Scaling program for the Editor before you finalize your code. This will let you get the complete range out of the potentiometer.

Now get to it: go build some candles. Remember, candles aren't just for Halloween, they work at Thanksgiving, at Hanukkah, at Christmas, at Kwanzaa, and probably on lots of other occasions I know nothing about. Just don't put them on a birthday cake – you'll never get to eat!

Until next time, have a safe and fun Halloween, and Happy Stamping!

**Additional Resources:**

Methodz of Madness Prop Builders forum  
[www.methodzofmadness.com](http://www.methodzofmadness.com)

Midwest Haunters Convention  
[www.midwesthauntersconvention.com](http://www.midwesthauntersconvention.com)

```

' =====
'
' File..... Candles.BS1
' Purpose.... Light faux candles (with LED wicks)
' Author..... Parallax Team EFX
' E-mail..... teamefx@parallax.com
' Started....
' Updated.... 18 OCT 2005
'
' {$STAMP BS1}
' {$PBASIC 1.0}
' =====

' ----[ Program Description ]-----
'
' This program controls up to six faux candles. Each candle "flame" is
' comprised of three LEDs, a resistor, and a capacitor. The capacitor
' allows the flame to fade when the BASIC Stamp output goes off, creating
' a more realistic effect.

' ----[ Revision History ]-----

' ----[ I/O Definitions ]-----
SYMBOL Candles      = PINS          ' candle outputs, P0-P5

' ----[ Constants ]-----
SYMBOL FlickBase    = 20           ' flicker base timing

' ----[ Variables ]-----
SYMBOL flicker      = W1           ' random flicker value
SYMBOL flickVar     = B3           ' affects flicker rate
SYMBOL idx          = B4           ' loop control
SYMBOL wicks        = B5           ' to test for dark
SYMBOL rate         = B6           ' flicker rate

' ----[ EEPROM Data ]-----

' ----[ Initialization ]-----
Reset:

```

## Column #126: Tricks and Treats with LEDs

```
DIRS = %00111111          ' make P0 - P5 outputs
flicker = 1031            ' seed random generator

' -----[ Program Code ]-----
Main:
  FOR idx = 1 TO 3          ' tumble random generator
    RANDOM flicker
  NEXT
Check_Dark:
  wicks = flicker & %00111111 ' test value
  IF wicks = %000000 THEN Main ' if dark, try again
Flame_On:
  Candles = wicks          ' update outputs
  rate = flickVar & $0F + FlickBase ' randomize timing
  PAUSE rate              ' hold flames
  GOTO Main                ' start again

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

```

' =====
'
' File..... Candles-2x.BS1
' Purpose.... Light faux candles (with LED wicks)
' Author..... Parallax Team EFX
' E-mail..... teamefx@parallax.com
' Started....
' Updated.... 22 AUG 2005
'
' {$STAMP BS1}
' {$PBASIC 1.0}
' =====

' ----[ Program Description ]-----
'
' This program controls two faux candles. Each candle "flame" is comprised
' of three independent LEDs circuits. The capacitor allows the flame to
' fade when the BASIC Stamp output goes off, creating a more realistic
' effect.
'
' This version also uses a trigger input to simulate wind, and a pot for
' fine-tuning candle flicker.

' ----[ Revision History ]-----

' ----[ I/O Definitions ]-----
SYMBOL RateCtrl      = 7           ' POT circuit input
SYMBOL Windy         = PIN6       ' active-high button
SYMBOL Candles       = PINS       ' candle outputs, P0-P5

' ----[ Constants ]-----
SYMBOL Yes           = 1
SYMBOL No            = 0

' ----[ Variables ]-----
SYMBOL flicker       = W1         ' random flicker value
SYMBOL wicks         = B4         ' to test for dark
SYMBOL rate          = B5         ' flicker rate

' ----[ EEPROM Data ]-----

```

## Column #126: Tricks and Treats with LEDs

```
' -----[ Initialization ]-----
Reset:
  DIRS = %00111111          ' make P0 - P5 outputs

' -----[ Program Code ]-----
Main:
  RANDOM flicker            ' tumble random generator

Check_Dark1:
  wicks = flicker & %00000111  ' isolate candle #1
  IF wicks > %000000 THEN Check_Dark2  ' if dark
  flicker = flicker | %000010  ' light middle

Check_Dark2:
  wicks = flicker & %00111000  ' isolate candle #2
  IF wicks > %000000 THEN Get_Rate  ' if dark
  flicker = flicker | %010000  ' light middle

Get_Rate:
  POT RateCtrl, 100, rate      ' read speed pot, 5 - 255

No_Wind:
  IF Windy = Yes THEN Has_Wind
  rate = rate / 25 + 5        ' scale to 5 - 15
  GOTO Flame_On

Has_Wind:
  rate = rate / 10 + 25      ' scale to 35 - 60

Flame_On:
  Candles = flicker & %00111111  ' update outputs
  PAUSE rate                  ' hold flames
  GOTO Main                   ' start again

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

```

' =====
'
' File..... Candles-6x.BS1
' Purpose.... Light faux candles (with LED wicks)
' Author..... Parallax Team EFX
' E-mail..... teamefx@parallax.com
' Started....
' Updated.... 22 AUG 2005
'
' {$STAMP BS1}
' {$PBASIC 1.0}
' =====

' ----[ Program Description ]-----
'
' This program controls six faux candles. Each candle "flame" is comprised
' of three LEDs, a resistor, and a capacitor. The capacitor allows the
' flame to fade when the BASIC Stamp output goes off, creating a more
' realistic effect.
'
' This version also uses a trigger input to simulate wind, and a pot for
' fine-tuning candle flicker.

' ----[ Revision History ]-----

' ----[ I/O Definitions ]-----

SYMBOL RateCtrl      = 7           ' POT circuit input
SYMBOL Windy         = PIN6        ' active-high button
SYMBOL Candles       = PINS        ' candle outputs, P0-P5

' ----[ Constants ]-----

SYMBOL Yes           = 1
SYMBOL No            = 0

' ----[ Variables ]-----

SYMBOL flicker       = W1           ' random flicker value
SYMBOL wicks         = B4           ' to test for dark
SYMBOL rate          = B5           ' flicker rate

' ----[ EEPROM Data ]-----

```

## Column #126: Tricks and Treats with LEDs

```
' -----[ Initialization ]-----  
Reset:  
  DIRS = %00111111          ' make P0 - P5 outputs  
  
' -----[ Program Code ]-----  
Main:  
  RANDOM flicker           ' tumble random generator  
  
Check_Dark:  
  wicks = flicker & %00111111    ' test value  
  IF wicks = %000000 THEN Main    ' if dark, try again  
  
Get_Rate:  
  POT RateCtrl, 100, rate        ' read speed pot, 5 - 255  
  
No_Wind:  
  IF Windy = Yes THEN Has_Wind  
  rate = rate / 25 + 5          ' scale to 5 - 15  
  GOTO Flame_On  
  
Has_Wind:  
  rate = rate / 10 + 25         ' scale to 35 - 60  
  
Flame_On:  
  Candles = wicks              ' update outputs  
  PAUSE rate                    ' hold flames  
  GOTO Main                     ' start again  
  
' -----[ Subroutines ]-----
```

```

' =====
'
' File..... Candles.BS2
' Purpose.... Light faux candles (with LED wicks)
' Author..... Parallax Team EFX
' E-mail..... teamefx@parallax.com
' Started....
' Updated.... 15 OCT 2005
'
' {$STAMP BS2}
' {$PBASIC 2.5}
' =====

' ----[ Program Description ]-----
'
' This program controls six faux candles. Each candle "flame" is comprised
' of three LEDs, a resistor, and a capacitor. The capacitor allows the
' flame to fade when the BASIC Stamp output goes off, creating a more
' realistic effect.

' ----[ Revision History ]-----

' ----[ I/O Definitions ]-----
Candles      VAR      OUTL      ' candle outputs, P0-P5

' ----[ Constants ]-----
FlickBase    CON      10          ' flicker base timing

' ----[ Variables ]-----
flicker      VAR      Word         ' random flicker value
flickVar     VAR      flicker.BYTE1 ' affects flicker rate
wicks        VAR      Byte         ' to test for dark
rate         VAR      Byte         ' flicker rate
idx          VAR      Nib

' ----[ EEPROM Data ]-----

' ----[ Initialization ]-----
Reset:

```

## Column #126: Tricks and Treats with LEDs

```
DIRL = %00111111          ' make P0 - P5 outputs

' -----[ Program Code ]-----
Main:
DO
  FOR idx = 1 TO 3
    RANDOM flicker          ' tumble random generator
  NEXT
  wicks = flicker & %00111111 ' test value
  IF (wicks <> %000000) THEN ' if not dark...
    Candles = wicks         ' update outputs
    rate = flickVar.NIB0 + FlickBase ' isolate lower 4 bits
    PAUSE rate              ' hold flames
  ENDIF
LOOP                        ' start again

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

```

' =====
'
' File..... Candles-2x.BS2
' Purpose.... Light faux candles (with LED wicks)
' Author..... Parallax Team EFX
' E-mail..... teamefx@parallax.com
' Started....
' Updated.... 15 OCT 2005
'
' {$STAMP BS2}
' {$PBASIC 2.5}
' =====

' ----[ Program Description ]-----
'
' This program controls two faux candles. Each candle "flame" is comprised
' of three independent LEDs circuits. The capacitor allows the flame to
' fade when the BASIC Stamp output goes off, creating a more realistic
' effect.
'
' This version also uses a trigger input to simulate wind, and a pot for
' fine-tuning candle flicker.

' ----[ Revision History ]-----

' ----[ I/O Definitions ]-----

RateCtrl      PIN      7           ' RCTIME circuit input
Windy         PIN      6           ' active-high button
Candles       VAR      OUTL        ' candle outputs, P0-P5

' ----[ Constants ]-----

Yes           CON      1
No            CON      0

' ----[ Variables ]-----

flicker      VAR      Word         ' random flicker value
wicks        VAR      Byte         ' to test for dark
rate         VAR      Word         ' flicker rate
idx          VAR      Nib

' ----[ EEPROM Data ]-----

```

## Column #126: Tricks and Treats with LEDs

```
' -----[ Initialization ]-----  
  
Reset:  
  DIRL = %00111111          ' make P0 - P5 outputs  
  
' -----[ Program Code ]-----  
  
Main:  
  FOR idx = 1 TO 3  
    RANDOM flicker          ' tumble random generator  
  NEXT  
  
Check_Dark1:  
  wicks = flicker & %00000111      ' isolate candle #1  
  IF (wicks = %000000) THEN        ' if dark  
    flicker = flicker | %000010    ' light middle  
  ENDIF  
  
Check_Dark2:  
  wicks = flicker & %00111000      ' isolate candle #2  
  IF (wicks = %000000) THEN        ' if dark  
    flicker = flicker | %010000    ' light middle  
  ENDIF  
  
Get_Rate:  
  HIGH RateCtrl            ' charge cap  
  PAUSE 1  
  RCTIME RateCtrl, 1, rate      ' read speed pot  
  rate = rate * / $0067 MAX 255  ' scale to 0 - 255  
  
Check_Wind:  
  IF (Windy = No) THEN  
    rate = rate / 25 + 5        ' scale to 5 - 15  
  ELSE  
    rate = rate / 10 + 25      ' scale to 35 - 60  
  ENDIF  
  
Flame_On:  
  Candles = flicker & %00111111    ' update outputs  
  PAUSE rate                    ' hold flames  
  GOTO Main                    ' start again  
  
' -----[ Subroutines ]-----
```

```

' =====
'
' File..... Candles-6x.BS2
' Purpose.... Light faux candles (with LED wicks)
' Author..... Parallax Team EFX
' E-mail..... teamefx@parallax.com
' Started....
' Updated.... 15 OCT 2005
'
' {$STAMP BS2}
' {$PBASIC 2.5}
' =====

' ----[ Program Description ]-----
'
' This program controls six faux candles. Each candle "flame" is comprised
' of three LEDs, a resistor, and a capacitor. The capacitor allows the
' flame to fade when the BASIC Stamp output goes off, creating a more
' realistic effect.
'
' This version also uses a trigger input to simulate wind, and a pot for
' fine-tuning candle flicker.

' ----[ Revision History ]-----

' ----[ I/O Definitions ]-----

RateCtrl      PIN      7           ' RCTIME circuit input
Windy         PIN      6           ' active-high button
Candles       VAR      OUTL        ' candle outputs, P0-P5

' ----[ Constants ]-----

Yes           CON      1
No            CON      0

' ----[ Variables ]-----

flicker      VAR      Word         ' random flicker value
wicks        VAR      Byte         ' to test for dark
rate         VAR      Word         ' flicker rate
idx          VAR      Nib

' ----[ EEPROM Data ]-----

```

## Column #126: Tricks and Treats with LEDs

```
' -----[ Initialization ]-----  
Reset:  
  DIRL = %00111111          ' make P0 - P5 outputs  
  
' -----[ Program Code ]-----  
Main:  
  FOR idx = 1 TO 3  
    RANDOM flicker          ' tumble random generator  
  NEXT  
  
Check_Dark:  
  wicks = flicker & %00111111    ' test value  
  IF (wicks <> %000000) THEN      ' if not dark...  
  
Get_Rate:  
  HIGH RateCtrl              ' charge cap  
  PAUSE 1  
  RCTIME RateCtrl, 1, rate      ' read speed pot  
  rate = rate * / $0067 MAX 255  ' scale to 0 - 255  
  
Check_Wind:  
  IF (Windy = No) THEN  
    rate = rate / 25 + 5        ' scale to 5 - 15  
  ELSE  
    rate = rate / 10 + 25      ' scale to 35 - 60  
  ENDIF  
  
  Candles = wicks              ' update outputs  
  PAUSE rate                    ' hold flames  
ENDIF  
GOTO Main                      ' start again  
  
' -----[ Subroutines ]-----
```