



Column #140, December 2006 by Jon Williams:

Celebrating the Season

At the risk of repeating a comment I think I make every December... I love this season. Sure, I could live without the cold weather, but the smell of fireplaces burning and holiday decorations everywhere I look just makes it all okay – even if I do have to put on long sleeves and a jacket before I go out. While many of us think about Christmas this time of the year, our Jewish friends enjoy a special event as well: Hanukkah (the Festival of Lights). A key element in the Hanukkah ritual is the Menorah, a special candelabra used to commemorate the eight days of the celebration. This month we're going to build a Menorah, an electronic version, that is.

Some of you will remember the faux candles project we did in October of 2005. Well, the faux candles generated more e-mail and phone calls than anything I've ever written about in this column – ever! The response has been amazing, and continues to this day. In the scheme of the prop-related programming I've done over the last year and a half, I think more than half of those projects (requested by others) have involved faux candles. One of the requests was for a "magic candles" prop that would light candles in a specific sequence, and that lead me to what we're doing this month.

The reason for the electronic Menorah is a bit personal. No, I'm not Jewish, but one of my dearest friends, Jenny, is – or, more accurately, is becoming. Religion is a deeply individual matter and I have great respect for any person who practices their religion faithfully while letting others do the same, and I stand in absolute awe of those who make the conscious decision to change. Religious conversion is a very serious undertaking, and my gift to Jenny

this December is a decorative electronic Menorah to commemorate the commitment to her new religion.

Let's get right to it. I've made this very simple because Jenny, while a wonderful actress, really doesn't like to fuss with high-tech gadgets. What this means is that she'll simply have to plug it in at the beginning of each evening of Hanukkah and unplug it before she goes to bed. The program will advance the day automatically, so there are no controls or settings to worry about (she'll like that!). Another choice I made, after consulting a Jewish friend, is that my Menorah has all candles installed all the time. In the actual Hanukkah ritual, one candle is added each day and then lit in a specific sequence. This design is not really intended to be used in religious services, but rather as a commemorative decoration. If you decide to make a version to be used in services, you might consider adding plugs to the candles with matching sockets on the Menorah.

Flicker Away

Simulating candle flicker – using just an LED – is not terribly easy with the BASIC Stamp due to the instruction fetch (from external EEPROM) and decoding, this makes using the PWM instruction a bit fruitless. My friend John came up with the candle circuit (see Figure 140.1) that cheats a bit by using a capacitor. When the control line is connected to ground through a device like the ULN2003, the LEDs will light and the capacitor will charge. When the control line is released the capacitor will discharge through the LEDs providing a slow fade, eliminating the digital nature of the controller output, giving more of a candle-like effect. Note that the control line needs an open-collector type control; do not turn the “wick” off by making the control line high as this will defeat the operation of the capacitor in the circuit. We'll use two ULN2003 Darlington arrays to control the candles. The schematic for the candle driver is shown in Figure 140.2.

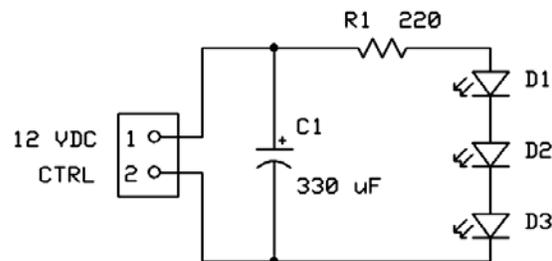


Figure 140.1: The Electric Candlewick Schematic

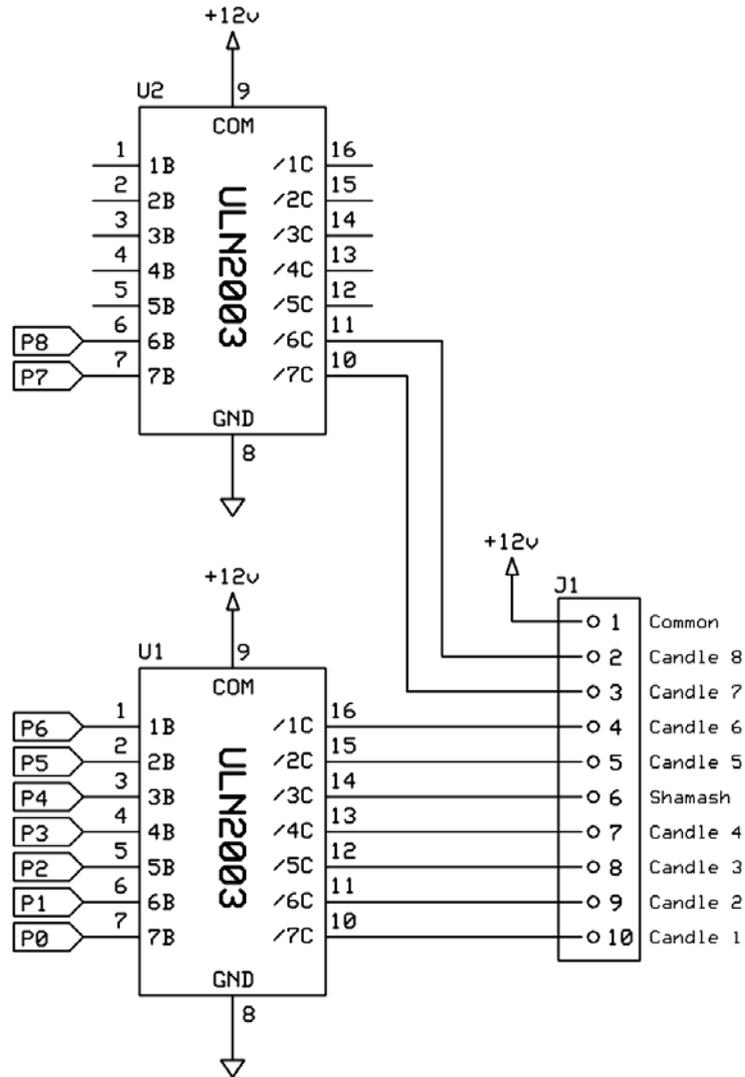


Figure 140.2: The Candle Driver Schematic

One of the aspects I enjoy most about prop programming is the simulation of real events. In the Hanukkah ceremony the “servant” (shamash in Hebrew) candle is lit first and then used to light the other candles in a specific sequence. Candles are inserted into the Menorah from right-to-left, but actually lit from left-to-right. As I stated earlier, my Menorah has all of the candles pre-installed so the only thing to consider is the lighting sequence. Figure 140.3 shows the lighting sequence for the eight days of Hanukkah.

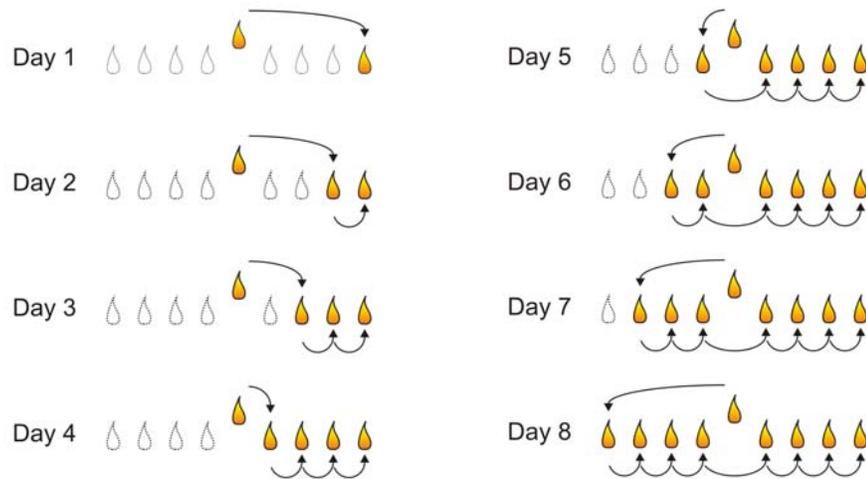


Figure 140.3: The Hanukkah Candle Lighting Sequence

There are any number of ways we could approach the coding of this project, but I tend to lean toward the idea that simple is best. To that end I decided to put the lighting sequence into a set of DATA tables and pull them out as needed for the current day in the cycle.

Let’s have a look at the sequence for day eight:

```

Wicks8      DATA      Word  %000010000
            DATA      Word  %100010000
            DATA      Word  %110010000
            DATA      Word  %111010000
            DATA      Word  %111110000
            DATA      Word  %111111000
            DATA      Word  %111111100
            DATA      Word  %111111110
            DATA      Word  %111111111
            DATA      Word  %111111111
    
```

As you can see there is one table entry for each candle lit. We start with the shamash (BIT4) and then work left-to-right from candle 8 to candle 1. We'll control the timing between each line to simulate a person moving a match from one candle to another.

At the top of the program we'll set our candle pins to outputs and then read the current day of the cycle from an EEPROM location.

```
Reset:
  DIRS = $01FF
  READ Day, dyNum
```

Once we drop into the main body of the program the day number is used to fetch the address of our candles table using LOOKUP.

```
Main:
  LOOKUP dyNum-1, [Wicks1, Wicks2, Wicks3,
                  Wicks4, Wicks5, Wicks6,
                  Wicks7, Wicks8], addr

Light_Candles:
  FOR cycle = 0 TO (dyNum * 2) STEP 2
    READ addr + cycle, Word mask
    GOSUB Flicker
  NEXT
```

A simple loop iterates through the table entries, moving the values from the table into a variable called mask. Note that we're multiplying the day number by two and using FOR-NEXT step value of two as the mask values in the tables are Words (two bytes each). Once the mask has been read for the sequence element this is passed on to the Flicker subroutine where it is used to selectively enable specific candles.

```
Flicker:
  timer = 0
  DO
    FOR idx = 1 TO 3
      RANDOM wicks
    NEXT
    Candles = wicks & mask
    delay = wicks // 26 + 25
    PAUSE delay
    timer = timer + delay
  LOOP UNTIL (timer >= 1000)
  RETURN
```

Column #140: Celebrating the Season

The purpose of this subroutine is to light the selected candles for one second, simulating the time to move from one candle to the next as if lighting real candles with a match. At the top of the subroutine the RANDOM function is called in a small loop.

Why do this – the result is random, right? Well, sort of. Remember that the RANDOM function uses a Linear Feedback Shift Register and if we don't call RANDOM at least three times we can actually see the movement of the bits through the LFSR in the parallel output of the candles. For pure numeric use, this isn't necessary, but if we're dealing with parallel outputs like the candles in this project it's best to call RANDOM at least three times. I've experimented with other values, and three seems to work best.

Now that we've got an apparently random set of bits we can do a bitwise AND with the mask to disable those candles that are not supposed to be lit at this moment. A small, somewhat randomized delay is derived from the random number and this value is added to an accumulator. Once we have "lit" candles for about a second we can return to the caller.

From time-to-time I get questions on creating a random value between X and Y. First, we need to call RANDOM, and doing it frequently is best. The general formula, then, works out like this:

```
value = random_number // (X + Y + 1) + X
```

This works because the modulus operator returns a value between zero and the divisor minus one – this is why we add one to the span for the formula.

Back in the main section we can increment the day pointer and write it back to the EEPROM for the next cycle. As you can see, we use our friend the modulus operator to wrap the day value from eight back to one, making the Menorah ready for next year.

```
Update_Day:
  dyNum = dyNum // 8 + 1
  WRITE Day, dyNum

  DO
    GOSUB Flicker
  LOOP
```

The end of the program we enter an infinite loop that flickers the day's candles – and that's all there is to it. This is a very simple program, and when combined with the beauty of the electronic wicks it results in a beautiful display.

Putting It Together

Since the Menorah control section is a one-off project I decided to make things easy by assembling the ULN2003 output drivers and terminal blocks on the Parallax BASIC Stamp Super Carrier. This board will hold any 24-pin BASIC Stamp module, the Javelin Stamp, and even the BS1-IC if you like (which won't work for this project because we need nine outputs). The board has a 2.1 mm barrel connector for power, a beefy five-volt regulator, and a standard serial connection for programming the BS2 or Javelin. For \$20 (which does not include BASIC Stamp) this seems like a really good deal, especially when one considers the time and energy to layout a custom PCB, or hand wire the whole works.

The user section of the Super Carrier has convenient dog-bone pads that make adding the two ULN2003s a breeze. I got a bit lucky in that I could just fit five, two-position terminal blocks (found them at RadioShack) into the board for the common plus nine wick control outputs. Everything is connected with point-to-point wiring using #26 gauge solid wire. I kept all connections on top, save one, as the easiest place to grab the Vin voltage was from the AppMod header. I removed the solder from this pin and made the V+ connection there. Figures 140.4 and 140.5 show the top and bottom of the completed control board.

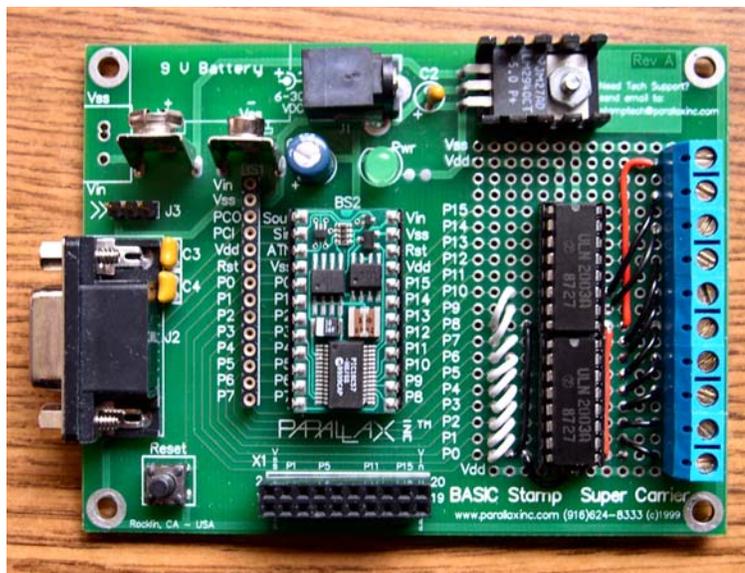


Figure 140.4: The BASIC Stamp Super Carrier Board, Top View

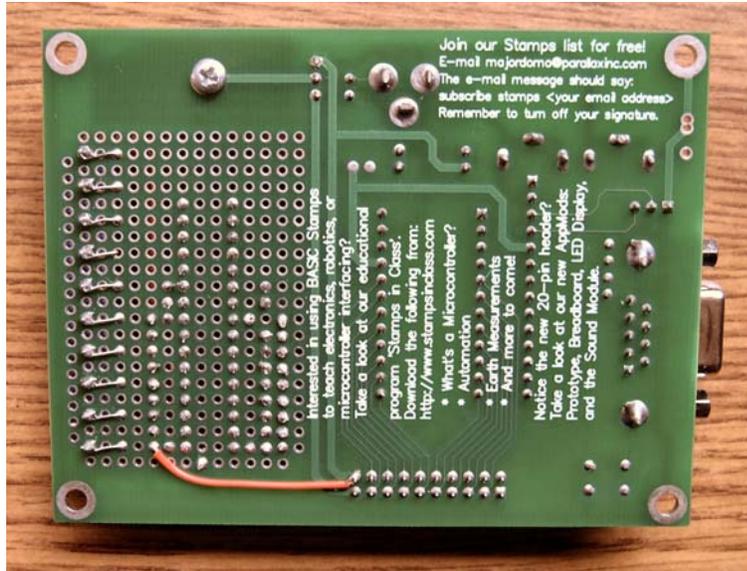


Figure 140.5: The BASIC Stamp Super Carrier Board, Bottom View

I just said that I wouldn't layout a board for the control section, but for the wicks I sing a completely different tune. I have hand-wired wicks in the past and believe me, it is not any fun at all. If you're going to do just one small project, that would be fine, but if you are going to build more than one project with the faux candles, you can either get them pre-built or build your own wick PCB.

I've saved you the trouble of laying out the wick PCB. In the download file you find a panelized PCB in ExpressPCB (www.expresspcb.com) format. ExpressPCB offers great service and very good prices, and I use them frequently when prototyping devices for my clients. The wick PCB is small, and as you can see in Figure 140.6, 21 wick boards will fit onto a "mini board" PCB. What this means is that for standard order of three mini boards, you'll end up with 63 wicks – that's a lot of candles, and a possible opportunity to split the board cost amongst friends!

The wicks are installed in short, pillar-style candles because these are the easiest to work with. After burning them for an hour or so to create a bit of a well, a 1/2" hole is drilled through the centerline to make room for the electronic wick. Use a wood boring bit and do go slowly; I learned the hard way that getting too aggressive can cause the candle to split. After coring the

candles I did a little additional sculpting with a knife and then smoothed everything with a butane torch. The wicks are installed from the bottom side of the candle so that the tops of the LEDs nearly protrude from the core.

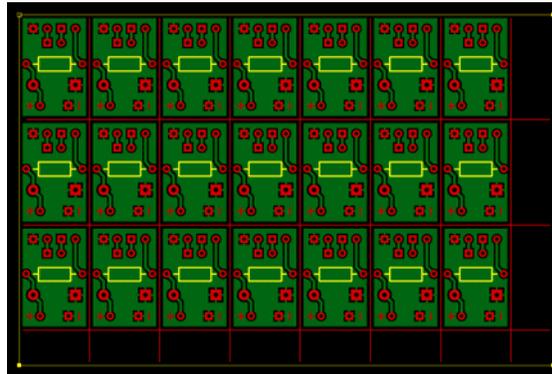


Figure 140.6: The BASIC Stamp® Super Carrier Board, Bottom View

Please, please, please... be safe when burning your candles and do make sure you do this in an area clear of flammable debris. And do not leave them to burn while unattended.

Finally, I asked around and found there is no specific requirement for the construction of the actual Menorah other than the central (shamash) candle being higher than the others, so I turned to master woodworker, Jeremy Black, for help. I wanted simple elegance, and Jeremy gave it to me in spades. My Menorah is a gorgeous oak platform that is large enough to accommodate nine, three-inch pillar-style candles. Since the platform is flat, I used a six-inch tall candle for the shamash.



Figure 140.7: The Completed Menorah

Column #140: Celebrating the Season

The only thing I didn't specify for Jeremy – which nearly got me – was the size of the PCB I intended to mount underneath the platform. As luck would have it, the Super Carrier just fit when I mounted it vertically. I think that you'll agree the final piece is beautiful, and I expect Jen will get many years of enjoyment from it.

Hopefully this project will give you ideas for additional holiday projects. For those that are fans of the Rogue uMP3 player (we used that last December), a fun project might be a “dancing candles” display that lights candles in sync with your favorite holiday song. After starting the audio you can read the “candle dance” out of a DATA table as we did in this project. Hmm, where's my uMP3 player?....

Happy Holidays

Well, that about wraps it up for another month and indeed another year. From my family to yours, let me wish you a very happy holiday season filled with peace and love, and a terrific new year for you and those in your life.

Until 2007... Happy Stamping!

```
' =====
'
'   File..... Hanukkah_Menorah.BS2
'   Purpose... Electronic Hannuka Menora
'   Author.... Jon Williams, EFX-TEK (www.efx-tek.com)
'   E-mail.... jwilliams@efx-tek.com
'   Started...
'   Updated... 05 OCT 2006
'
'   {$STAMP BS2}
'   {$PBASIC 2.5}
'
' =====
'
' -----[ Program Description ]-----
'
' Simulates the lighting of a Hanukkah Menorah. At the end of each day
' the program self updates to point to the next day of the cycle.
'
' -----[ Revision History ]-----
```

The Nuts and Volts of BASIC Stamps 2006

```
' -----[ I/O Definitions ]-----
Candles          VAR      OUTS          ' P0..P8

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

' -----[ Variables ]-----
idx              VAR      Nib              ' loop controller
dyNum            VAR      Nib              ' current day
cycle            VAR      Byte             ' controls lighting cycle
delay            VAR      Byte             ' delay for flickering
addr             VAR      Word             ' address of daily sequence
mask             VAR      Word             ' maks for lit candles
timer            VAR      Word             ' flicker timer
wicks            VAR      Word             ' lit wicks

' -----[ EEPROM Data ]-----
Day              DATA     1              ' current day

'
'                               +----- Shamash candle
'                               |
Wicks1           DATA     Word %000010000
                 DATA     Word %000010001

Wicks2           DATA     Word %000010000
                 DATA     Word %000010010
                 DATA     Word %000010011

Wicks3           DATA     Word %000010000
                 DATA     Word %000010100
                 DATA     Word %000010110
                 DATA     Word %000010111

Wicks4           DATA     Word %000010000
                 DATA     Word %000011000
                 DATA     Word %000011100
                 DATA     Word %000011110
                 DATA     Word %000011111

Wicks5           DATA     Word %000010000
                 DATA     Word %000110000
                 DATA     Word %000111000
                 DATA     Word %000111100
                 DATA     Word %000111110
                 DATA     Word %000111111
```

Column #140: Celebrating the Season

```
Wicks6      DATA      Word %000010000
            DATA      Word %001010000
            DATA      Word %001110000
            DATA      Word %001111000
            DATA      Word %001111100
            DATA      Word %001111110
            DATA      Word %001111111

Wicks7      DATA      Word %000010000
            DATA      Word %010010000
            DATA      Word %011010000
            DATA      Word %011110000
            DATA      Word %011111000
            DATA      Word %011111100
            DATA      Word %011111110
            DATA      Word %011111111

Wicks8      DATA      Word %000010000
            DATA      Word %100010000
            DATA      Word %110010000
            DATA      Word %111010000
            DATA      Word %111110000
            DATA      Word %111111000
            DATA      Word %111111100
            DATA      Word %111111110
            DATA      Word %111111111

' -----[ Initialization ]-----

Reset:
  DIRS = $01FF          ' P0..P8 are outputs
  READ Day, dyNum      ' get current day

' -----[ Program Code ]-----

Main:
  LOOKUP dyNum-1, [Wicks1, Wicks2, Wicks3,
                  Wicks4, Wicks5, Wicks6,
                  Wicks7, Wicks8], addr ' light candles for day
                                      ' get address of sequence

Light_Candles:
  FOR cycle = 0 TO (dyNum * 2) STEP 2 ' light candles for this day
    READ addr + cycle, Word mask      ' get lit candles mask
    GOSUB Flicker                     ' update candle wicks
  NEXT

Update_Day:
  dyNum = dyNum // 8 + 1              ' point to next day
```

```

WRITE Day, dyNum          ' save to EEPROM

DO
  GOSUB Flicker          ' stay lit until shutdown
LOOP

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

' Flicker lit candles for one second
' -- pass mask value for lit candles

Flicker:
  timer = 0              ' clear timer
  DO
    FOR idx = 1 TO 3     ' tumble random value
      RANDOM wicks
    NEXT
    Candles = wicks & mask ' "light" selected candles
    delay = wicks // 26 + 25 ' random delay, 25 to 50 ms
    PAUSE delay
    timer = timer + delay ' update timer
  LOOP UNTIL (timer >= 1000) ' done? if not, keep going
  RETURN

' -----

```