

Code (text):

```
//
// Emulates a Candleflicker-LED
// ported from CandeflickerLED.c
//
device = 18F4520
clock = 8

// LED connected to PB0 (set this to match your hardware)
dim LEDPIN as PORTB.0

// main flicker clock = 440*16 Hz
const FLICKER_DELAY as word = ((1000000/440)/16)

//
// 32 Bit maximum length LFSR
// see http://www.ece.cmu.edu/~koopman/lfsr/index.html
// Using inverted values so the LFSR also works with zero initialisation
//
dim rand_z as longword

function random() as byte
    rand_z = rand_z >> 1

    if (rand_z.bits(0) = 0) then
        rand_z = rand_z xor $7FFFF159
    endif

    result = rand_z.byte0
end function

dim pwm_ctr as byte    // 4 bit-Counter
dim frame_ctr as byte  // 5 bit-Counter
dim pwm_val as byte    // 4 bit-Register
dim nextbright as byte // 4 bit-Register
dim rand as byte        // 5 bit Signal
dim randflag as boolean // 1 bit Signal

// initialization
rand_z = 0
pwm_ctr = 0
frame_ctr = 0
pwm_val = 0
nextbright = 0
rand = 0
randflag = false
```

```

low(LEDPIN)

while (true)
    delayus(FLICKER_DELAY)

    // PWM
    pwm_ctr = pwm_ctr + 1
    pwm_ctr = pwm_ctr and $0f // only 4 bit

    if (pwm_ctr <= pwm_val) then
        LEDPIN = 1
    else
        LEDPIN = 0
    endif

    // frame
    if (pwm_ctr = 0) then
        frame_ctr = frame_ctr + 1
        frame_ctr = frame_ctr and $1f

        // generate a new random number every 8 cycles
        // In reality this is most likely bit serial
        if ((frame_ctr and $07) = 0) then
            rand = random() and $1f
            if ((rand and $0c) <> 0) then
                randflag = true // only update if valid
            endif
        endif

        // new frame
        if (frame_ctr = 0) then
            pwm_val = nextbright // reload PWM
            randflag = true // force update at beginning of frame
        endif

        if (randflag) then
            if (rand > 15) then
                nextbright = 15
            else
                nextbright = rand
            endif
            randflag = false // not in original code
        endif
    endif

end while

```