

1 Equations for the Breathing LED Indicator

When the lid of an Apple Macintosh laptop is closed, an LED indicator light pulses with the rhythm of human breathing. On December 2, 2003, Apple was awarded US Patent number 6,658,577 for an *Breathing status LED indicator*. The goal of this exercise is to develop an Arduino program to imitate the Apple LED indicator.

The goal of this exercise is to develop an Arduino program to imitate the Apple LED indicator. The Arduino will run a program that changes the effective voltage supplied to the LED. To get the visual effect of a “breathing” LED, the voltage (and ultimately the power) supplied to the LED has to vary in a cyclic pattern. In these notes, we discuss the procedure for obtaining the equations that describe the cyclic pattern, that in turn controls the brightness of the LED as a function of time.

In practice, the brightness of an LED cannot be specified directly with an Arduino command¹. Instead, the LED brightness is controlled indirectly by changing duty cycle of the PWM output that supplies power to the LED circuit. Focusing on the PWM output is a helpful simplification and is sufficient to create the breathing effect.

The PWM output is increased and decreased in a regular pattern during each breath cycle. The time variation of the PWM output is specified with a piecewise continuous mathematical function defined by two compact formulas. There are alternative strategies, e.g., table look-up, but using a piecewise continuous function provides a compact representation that can be easily modified.

1.1 Learning Objectives

These notes develop some simple formulas for the shape of the PWM curve used to drive the breathing LED. The learning objectives for this exercise are

1. Be able to define the characteristics of a piece-wise continuous function.
2. Be able to sketch the shape of the PWM output curves.
3. Be able to obtain the coefficients of an exponential function, given two data pairs that the function must pass through.
4. Be able to specify the two exponential functions that define the inhale and exhale portions of the breathing LED cycle.

These exercises presume that you already understand

- The meaning of exponential and logarithmic functions.
- How to use algebra to solve for two equations in two unknowns.

The details necessary to converting these equations to a working Arduino program are left to another document.

¹In fact “brightness” is not a rigorous term. In lighting technology the terms “luminance” (or luminosity) and “radiance” have rigorous definitions. Wikipedia has a useful introduction to these terms.

2 Mathematical Model of the Breathing LED

The diagram in Figure 2 represents the pattern of PWM output to be used in the final implementation of the Arduino program. The shape of the curve corresponds to the growing brightness and then dimming of the LED. We will use this pattern to specify the duty cycle of the PWM output as it is controlled by the Arduino `analogWrite` command.

The breathing pattern has three phases: inhale, pause, and exhale. The inhale and exhale phases are modeled with exponentially increasing and exponentially decreasing functions of time. The pattern repeats every $t_4 - t_1$ units of time. The inhale phase ends at t_2 , and the exhale phase begins at t_3 .

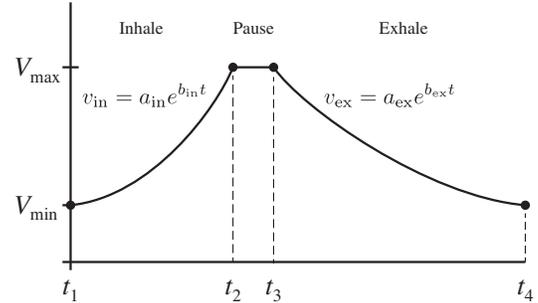


Figure 1: One breathing cycle.

2.1 Piecewise functions

The breathing pattern repeats indefinitely, as suggested by the diagram in Figure 2.1. We only need to define the function for the first complete cycle of inhale-pause-breath. The PWM output function is *piecewise-continuous*. The function is continuous because at no time is there a sudden change in the *value* of the function. Clearly the slope changes abruptly at each point marked by a solid dot. The function is called *piecewise* because the mathematical definition of the function (which has not yet been specified) changes along the t axis. In other words, not only is v a function of time, the definition of $v(t)$ is different for different time intervals. The boundary for the function definitions are marked by vertical dashed lines and solid dots on the $v(t)$ curve.

The piecewise function does not create any special difficulty for an Arduino program. The Arduino program needs a way to keep track of time so that the appropriate function for each interval can be selected.

The equations for the inhale and exhale phases are, respectively

$$v = a_{\text{in}} e^{b_{\text{in}} t} \quad v = a_{\text{ex}} e^{b_{\text{ex}} t} \quad (1)$$

where v is the value sent to the PWM output. Remember that the values used in the `analogWrite` function must be limited to the range $0 \leq v \leq 255$.

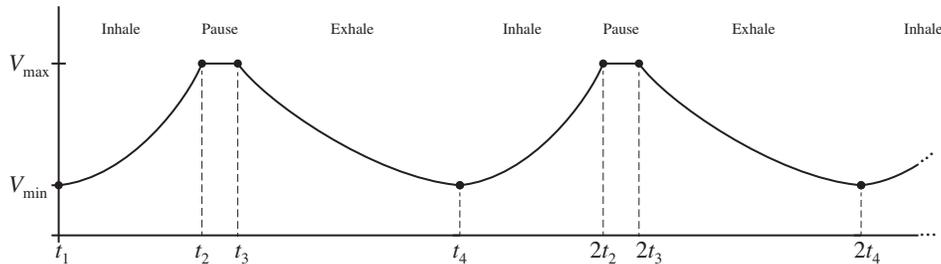


Figure 2: The continuous pattern of breathing is obtained by joining segments for individual breathing cycles. The power (or brightness) at the end of one cycle ($t = t_3$) should be equal to the power at the start of the next cycle.

The goal of these notes is to obtain formulas for a_{in} , b_{in} , a_{ex} and b_{ex} given specified values of v and t along the inhale and exhale curves. The equation for the pause phase is simple: $v = V_{\text{max}}$, where V_{max} is a constant.

2.2 Using Two Points to Define the Coefficients of $v(t)$

The inhale and exhale functions are of the form

$$v = ae^{bt} \quad (2)$$

We require that this function pass through two points (t_A, v_A) and (t_B, v_B) . Substituting these data pairs into Equation (2) gives

$$v_A = ae^{bt_A} \quad (3)$$

$$v_B = ae^{bt_B} \quad (4)$$

Points A and B can be any two pair of (t, v) values along the inhale and exhale curves. In practice, these points will be chosen as the endpoints of the inhale curve and the endpoints of the exhale curve.

Whereas Equation (2) is the $v(t)$ curve for any point along either the inhale or the exhale curve, Equations (3) and (4) apply to two specific data pairs. We assume that (t_A, v_A) and (t_B, v_B) are known. This leads us to interpret Equations (3) and (4) as conditions that determine the values for a and b . Therefore, by solving for the a and b that satisfy Equations (3) and (4), we obtain the coefficients of the function that passes through (t_A, v_A) and (t_B, v_B) .

2.2.1 Linear Version of the Equation for Unknowns a and b

Finding the values of a and b that satisfy Equations (3) and (4) is the basic problem of solving two equations for two unknowns. The equations are non-linear, but in this case that slight problem is simplified by using the properties of the natural logarithm and the exponential function. Recall that if

$$z = xy$$

then

$$\ln(z) = \ln(x) + \ln(y).$$

In words: the logarithm of a product is the sum of the logarithms of the terms being multiplied. Also recall that if

$$r = e^{st}$$

then

$$\ln(r) = st.$$

In other words $\ln(\bullet)$ is the inverse of $e^{(\bullet)}$, and $e^{(\bullet)}$ is the inverse of $\ln(\bullet)$. If we take the logarithm of Equation (2) and apply the rules for manipulating logarithms of products, we get

$$\ln(v) = \ln[ae^{bt}] \quad \longrightarrow \quad \ln(v) = \ln(a) + \ln[e^{bt}] \quad \longrightarrow \quad \ln(v) = \ln(a) + bt$$

Applying that transformation to Equations (3) and (4) gives

$$\ln(v_A) = \ln(a) + bt_A \quad (5)$$

$$\ln(v_B) = \ln(a) + bt_B \quad (6)$$

These are two *linear* equations for the two unknowns, $\ln(a)$ and b . Now that we have transformed the system of nonlinear equations to a system of linear equations, it is a matter of simple algebra to solve for the a and b . Note that once we have a value of $\ln(a)$ we compute a with the exponential function, i.e., $\exp[\ln(a)] = a$.

2.2.2 Solving for a and b

Subtract Equation (6) from Equation (5) to get

$$\ln(v_A) - \ln(v_B) = b(t_A - t_B) \quad (7)$$

Since t_A , v_A , t_B and v_B are known, we can solve for b

$$b = \frac{\ln(v_A) - \ln(v_B)}{t_A - t_B}. \quad (8)$$

Now that the formula for b is known, we can substitute Equation (8) into either Equation (5) or Equation (6) to solve for a . This involves a bit more algebraic work, but the steps are straightforward

$$\begin{aligned} \ln(v_A) &= \ln(a) + \left[\frac{\ln(v_A) - \ln(v_B)}{t_A - t_B} \right] t_A \\ \ln(a) &= \ln(v_A) - \left[\frac{\ln(v_A) - \ln(v_B)}{t_A - t_B} \right] t_A \\ &= \ln(v_A) \left[\frac{t_A - t_B}{t_A - t_B} \right] - \left[\frac{\ln(v_A) - \ln(v_B)}{t_A - t_B} \right] t_A \\ &= \frac{\ln(v_A) [t_A - t_B] - [\ln(v_A) - \ln(v_B)] t_A}{t_A - t_B} \\ &= \frac{t_A \ln(v_B) - t_B \ln(v_A)}{t_A - t_B} \end{aligned}$$

where, in the last step the $+t_A \ln(v_A)$ and $-t_A \ln(v_A)$ terms in the numerator cancel each other. Applying the exponential function to both sides of the preceding equation gives the formula for computing a

$$a = \exp \left[\frac{t_A \ln(v_B) - t_B \ln(v_A)}{t_A - t_B} \right]. \quad (9)$$

3 Applying the Model Equations

In Section 2, the features of the breathing pattern were translated into mathematical formulas for brightness as a function of time. To use this model in an Arduino program we need to do some setup work to find the a_{in} , b_{in} , a_{ex} and b_{ex} coefficients. You are free to adjust V_{min} , V_{max} , t_2 , t_3 and t_4 until you get a pleasing display. It's best to assume that you will have to iterate these steps until your breathing pattern looks right.

1. Choose appropriate values of V_{min} , V_{max} , t_2 , t_3 and t_4 . These are somewhat arbitrary *design* choices that you make to achieve a desired look to your inhale and exhale functions. Note that $t_1 = 0$.
2. Use Equations (9) and (8) to compute a_{in} and b_{in} .
3. Use Equations (9) and (8) (again) to compute a_{ex} and b_{ex} .

For a given set a_{in} , b_{in} , a_{ex} and b_{ex} values, it is a good idea to add this step

4. Plot the $v_{\text{in}}(t)$ and $v_{\text{ex}}(t)$ functions (say, with Excel, MathCAD, or MATLAB) to make sure you do not have an error in your algebra.

Once the variation of $v(t)$ looks good on paper, convert the formulas into Arduino code and test the system with an LED.