Eclectronics Project 1
Logan Sweet

14 September 2017

Introduction

The purpose of this lab was to design a circuit that flashed within 15% of 1Hz. To do this, I evaluated the accuracy of my circuit design in LTspice and MATLAB, then layed it out in KiCad. All parts were selected from the provided BOM, and will be soldered to the board upon arrival.

Calulating τ, Capacitor, and Resistor Values

Instead of starting with values for R_1 and R_2, I chose a value of τ and a pair of resistors that would create the correct frequency along with that τ value. I chose $\tau = 1$ for simplicity, which I created with a $10M$ resistor and a $0.1 \times 10^{-7}\mu F$ capacitor. Using the equations we derived in class along with my selected τ value, I was able to find a relationship between R_1 and R_2, as seen below.

When V_{-} is falling from V_2 to V_1, it has the equation

$$(e^{\frac{-t}{\tau}}) \times (3.3 \times \frac{R_1}{R_1 + R_2} + \frac{3.3}{2} \times \frac{R_2}{R_1 + R_2}$$

And its final value at V_1 before it reverses direction will be

$$\frac{3.3}{2} \times \frac{R_2}{R_1 + R_2}$$

Since I have already selected a value of $\tau = 1$ and we want to force a frequency of 1 Hz, I set $\tau = 1$ and $t = 0.5$. I can then set the above equations equal to one another to find the R_1 and R_2 relationship that will make the transition fit that timescale and have that τ value.

$$(e^{\frac{-.5}{1}})(3.3 * \frac{R_1}{R_1 + R_2} + \frac{3.3}{2} * \frac{R_2}{R_1 + R_2}) = \frac{3.3}{2} * \frac{R_2}{R_1 + R_2}$$

$$(e^{-.5}) * (\frac{3.3 * R_1}{R_1 + R_2} + \frac{\frac{3.3}{2} * R_2}{R_1 + R_2}) = \frac{\frac{3.3}{2} * R_2}{R_1 + R_2}$$

$$(e^{-.5})(R_1 + \frac{1}{2}R_2) = \frac{1}{2}R_2$$

$$(R_1e^{-.5} + \frac{1}{2}R_2e^{-.5}) = \frac{1}{2}R_2$$

$$R_1 = \frac{R_2(1 - e^{-.5})}{2 * e^{-.5}} \approx 0.324361 * R_2$$

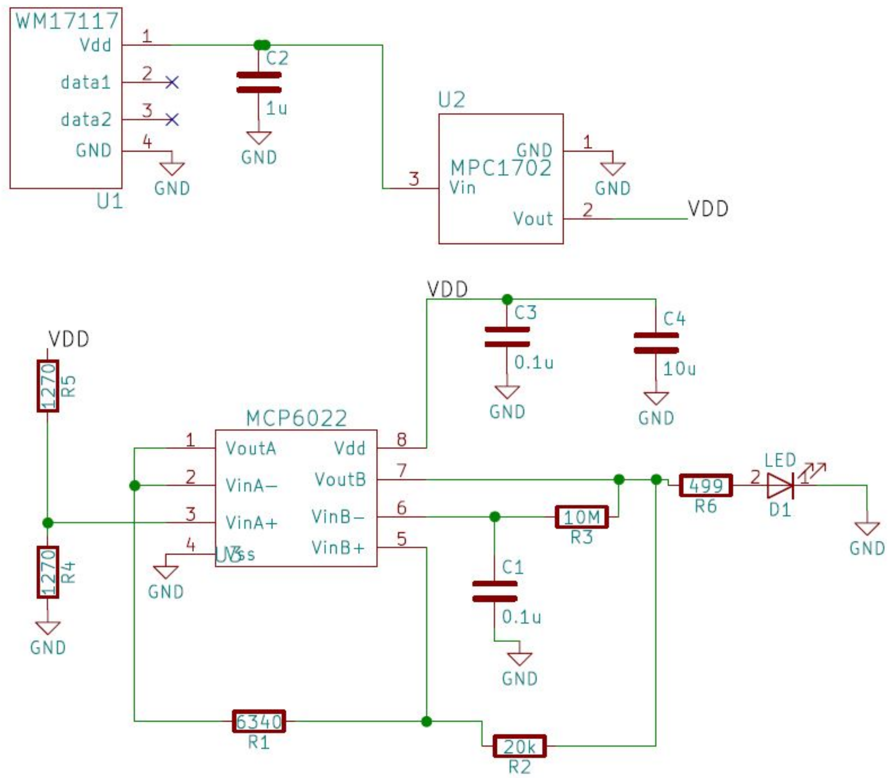
I then used excel to make a sheet with all resistor values in the first row and column. I used the $= \$A2/C\$1$ command on the first available cell, and dragged it to fill the whole sheet to make a sort of "division table" of all resistor values. I found a calculated value that was close to my R_1R_2 relationship: $20K\Omega$ and 6340Ω $(20000\Omega \times 0.324361 = 6487\Omega)$

I simulated the circuit in LTspice, and using the cursor I was able to measure a frequency of 1.03 Hz, which I expected to be within the 15% tolerance given in the assignment after I took into account calacitor and resistor tolerances.

Validating Tolerance

I used the "worse case" function as found on the youtube video here: `youtube.com/watch?v=5_ZzX88GzwE`. This produced a variety of initial capacitor charging times, so I collected data from 57 to 60 seconds after the beginning of the simulation. I exported all simulations in a single file and MATLAB to process the data. After removing the lines with text (manually, ugh) I shifted the voltage up so that it crossed the x axis, and recorded the times when it crossed. This sometimes recorded two points of the same crossing, so I calculated the period of the oscillation for a couple differences of times. Finally, I saved a 1 in a matrix for each time that was within tolerance, and a zero for ones that were out of tolerance. Out of 87 trials only one simulation was out of tolerance, so I feel fairly optimistic that my PCB will turn out fine. The MATLAB code that I used folows at the end of this document.

Circuit Schematic



Bill of Materials

Component Name	Part Description	Quantity
USB connector	CONN PLUG USB 4POS RT ANG PCB	1
Voltage Regulator	IC REG LIN 3.3V 250MA SOT23A-3	1
Op-Amps	IC OPAMP GP 10MHZ RRO 8TSSOP	1
LED	LED BLUE CLEAR 0805 SMD	1
10 μ Capacitor	CAP CER 10UF 16V X7R 0805	1
1 μ Capacitor	CAP CER 1UF 25V X7R 0603	1
.1 μ Capacitor	CAP CER 0.1UF 50V X7R 0603	2
499 Ω Resistor	RES SMD 499 OHM 1% 1/10W 0603	1
1270 Ω Resistor	RES SMD,1.27K OHM 1% 1/10W 0603	2
6340 Ω Resistor	RES SMD 6.34K OHM 1% 1/10W 0603	1
20k Ω Resistor	RES SMD,20K OHM 1% 1/10W 0603	1
10M Ω Resistor	RES SMD 10M OHM 1% 1/10W 0603	1

Conclusion

I liked building a circuit that is not breadboard-wired
And hope that my files include all that is required.
It was very interesting to learn KiCad,
But late on Monday night the learning curve made me sad.
Though my PCB is small, the aesthetics leave much to be desired.

MATLAB Code

```

load EclectronicsCSV.csv;
shift = EclectronicsCSV(:,2) - 1.65 ;        % dc offset to ensure voltage passes zero
times = EclectronicsCSV(:,1);
simrow = 1;
dcount = 1;
datacol = 1;
checks = 1;

while simrow < 36780                         % number of rows of data

    if (-0.23 < shift(simrow)) && (shift(simrow) < 0.23)        % if the voltage crosses zero
        data(dcount, datacol) = times(simrow);                   % record the time when it happened
        dcount = dcount + 1;
        simrow = simrow + 1;
    else
        simrow = simrow + 1;
    end

    if times(simrow+1) < times(simrow)       % if you reach a new simulation's data, start a new column
            datacol = datacol+1;
            dcount =  1;
    end
end

while checks < 88                   % subtract the times from one another. adjacent times were sometimes half a period, so the nest time was also calcuated
    timediff(checks,1) = data(2,checks) - data(1,checks) ;
    timediff(checks,2) = data(3,checks) - data(1,checks) ;
```

```
    checks = checks + 1;
end

checks = 1;

while checks < 88           % put a 1 in the finalcheck matrix if one of the calculated values was within the tolerance
    if (timediff(checks,1) > 0.85) && (timediff(checks,1) > 0.85)
        finalcheck(checks,1) = 1;
    elseif (timediff(checks,2) > 0.85) && (timediff(checks,2) > 0.85)
        finalcheck(checks,1) = 1;
    else
        finalcheck(checks,1) = 0;
    end
    checks = checks + 1;
end


hertz = -1*(1 ./ timediff ) ;           % out of 87 trials , only one was out of tolerance
```