CS 341 – Lab 8
A Low Cost Frequency Meter

Background Information

This lab is based on an article “A simple, low cost, precision frequency meter uses only two pins of a PC parallel port” written by Radovan Stojanovic published in the Design Ideas section of the EDN magazine. His setup is shown in Figure 1:

![Figure 1](image1.png)

The square wave generator (f_{IN}) puts out a pulse as shown in Figure 2 and feeds it to a pin of the line printer port of a computer.

![Figure 2](image2.png)

The pulse has a width (the level when the output is HIGH) and a period (the time between cycles). A perfect square wave is generated when width = Period/2. The frequency in hertz (or cycles per second) of the square wave is calculated using the formula:

\[
\text{frequency} = \frac{1 \times 10^6}{\text{Period}} \quad \text{for Period in microseconds (usec or 10^{-6} sec)}
\]

The computer runs a program to read the pin level of the line printer port. By measuring the time when the pin is high, the program computes the pulse
width. By measuring the total time when the pin is high and when the pin is low in one cycle, the program computes the period and the frequency.

In this lab, we set up the computer to read the level of an I/O pin that is connected to a pulse generator. The pulse width is measured using a polling method that checks whether the level of the I/O pin has changed according to the following logic:

```c
//start the measurement when the pin level just turns high
get current_time_1;
here: read the pin level; if pin is high, loop here;
get current_time_2;
width= current_time_2 – current_time_1;
```

Use a similar method to measure the time when the I/O pin level is low. The period is obtained by adding the pulse width (when I/O pin is high) and the duration when the I/O pin is low.

**The Code**

This lab requires you to modify the [starter code](#).

**Program the Arduino**

1. With the UnoArduSim simulator, find the square wave generator (labeled “pulser”).
2. Enter 10 in the top right box of pulser; 20 milliseconds (msec or $10^{-3}$ sec) in pulse width and 100 ms as a period. Click on pin 10 to see a digital waveform that looks like:
3. Write a program to measure the pulse width and the period.
4. In the program, set up pin 10 as an input.
5. Use the function micros() to read the current_time_1 in microseconds (usec). Please note: 1 msec = 1000 usec.
6. Read the level of pin 10. If the pin is high, loop.
7. When the pin goes low, call the function micros() again to get the current_time_2 in usec.
8. Pulse width in usec = current_time_2 - current_time_1.
9. Check against the pulse width setting in step 2 to see if this measurement is right. You can change the pulse width in step 2 to another value. Restart the program and measure the pulse width again.

10. Program more instructions to measure the time when pin 10 is low using steps similar to that outlined in steps 5-8. Calculate the period by adding the 2 numbers together and then the frequency using the above formula.

**Lab Report**

Please answer the following questions in the “Results” section of your report:

1. With the setting in step 2, what is your measured pulse width, period and frequency?
2. What is the accuracy of this measurement technique? What happens when you reduce the pulse width to 10 usec with a period of 50 usec?
You can submit your lab report by email to jack.davis001@umb.edu. Please cc: all group members so that his replies reach everyone.