S2014, BME 101L: Applied Circuits Lab 03a: hysteresis

Kevin Karplus

April 5, 2014

1 Design Goal

There are three parts to this lab:

- characterizing a Schmitt-trigger inverter.
- designing a hysteresis oscillator for use as a capacitance touch sensor.
- soldering the circuit on a PC board.

For characterizing the Schmitt-trigger, the goal is simply to determine the two input voltage thresholds at which the output switches from low to high and from high to low.

The oscillator circuit is an extremely simple one, with the Schmitt trigger inverter, one resistor, and one capacitor. The design problem is merely to select appropriate parameters for the resistor and capacitor.

The soldering practice is just that—practice at using the soldering iron on a simple enough circuit that few solder points are needed. You will also make a simple capacitance touch plate (a conductor with a thin insulator covering it), whose characteristics will affect the design of the hysteresis oscillator. Everybody needs to solder their own oscillator board—one per group is not enough.

2 Background

2.1 What is hysteresis, and why do we need it?

When we put a signal into a digital input of a computer, we want it to be treated as a simple on or off. There are lots of inputs we might want to treat this way (detecting a human input on a control panel, detecting the lack of water in a water bath, and so forth).

The electronics will deliver a voltage to a digital input pin, which the computer will interpret as a 0 or 1. You can think of this as a 1-bit analog-to-digital converter. The simplest model of a digital input is a simple step function: below a threshold voltage V_t the input is interpreted as 0, and above that voltage, it is interpreted as 1. This model is a bit too simple sometimes, as voltages near the threshold may not result in a clean 0-or-1 decision and may even damage some circuits if held for too long. A more realistic one is shown in Figure 1.

If we have noisy inputs (the usual case in the real world), the interpretation of the input as a digital signal can get difficult. See, for example, Figure 2, which shows what happens when we have a digital input using the transfer characteristic of Figure 1 interpreting a noisy input. Note that increasing the gain of the input circuit would not help by itself, as we would still see a series of pulses as the input voltage crossed back and forth across the threshold.

One solution to this problem is to have two thresholds and have the conversion from a noisy analog input to a digital output have one bit of memory. If you are currently in the high state, you have to cross the lower threshold before you go to the low state, and if you are in the low state, you have to cross the higher threshold to go to the high state. The transfer characteristic is shown in Figure 3, and how such a digital input circuit would handle the same noisy input as Figure 2 is shown in Figure 4.

Although the transfer characteristic of Figure 1 is fairly representative of normal digital inputs, which expect their inputs to stay away from the threshold as much as possible, the circuits for *Schmitt triggers*

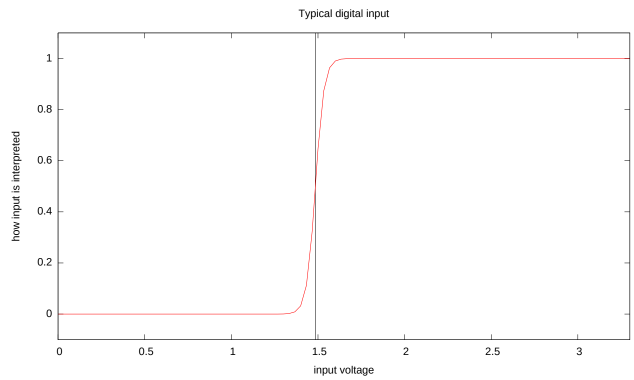

Figure 1: Representative transfer characteristic for a digital input. The *gain* of the input is the slope of the curve at its steepest point. Increasing the gain makes the 0-or-1 decision crisper, but often at the cost of other desirable properties (such as speed or power consumption). The comparator chips that we'll look at later this quarter use this approach of having a very high gain (and an extra input for setting the threshold) for converting analog signals to one-bit digital values.

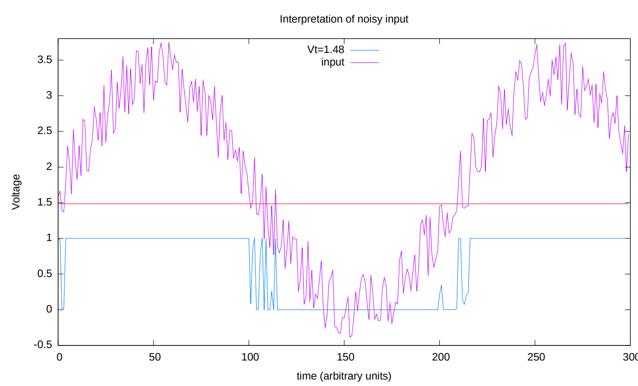

Figure 2: The results of interpreting a noisy input with the simple function of Figure 1. Note that when the input is near the threshold, the output can fluctuate wildly. If you were trying to count the events (button presses, cells in a flow sorter, . . .), you could end up counting many more than you should.

which implement hysteresis for digital inputs, have much higher gain (steeper slopes) and often have a larger separation between the two thresholds V_{IL} and V_{IH}. The separation $V_{IH} - V_{IL}$ is called the *hysteresis voltage*. The gain (the slope of the curve) is not specified on 74HC14N data sheets and was too high for me to measure easily—more than 1000, compared to a gain of only 10 in Figure 3.

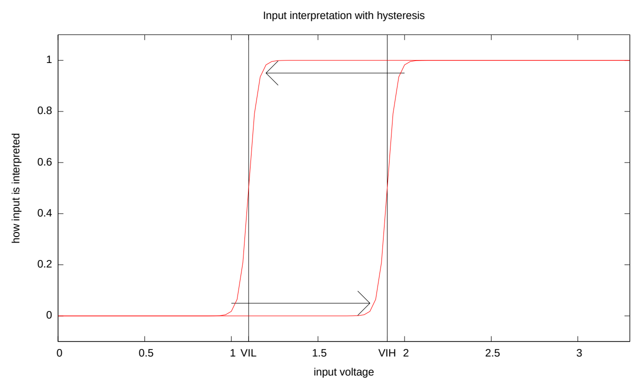

Figure 3: Representative transfer characteristic for a digital input with hysteresis. Note that the output is not a function of just the input, but of the input and the previous state. The hysteresis curve here is drawn with the same gain as Figure 1. The horizontal separation between the curves is the *hysteresis voltage* $V_{IH} - V_{IL}$.

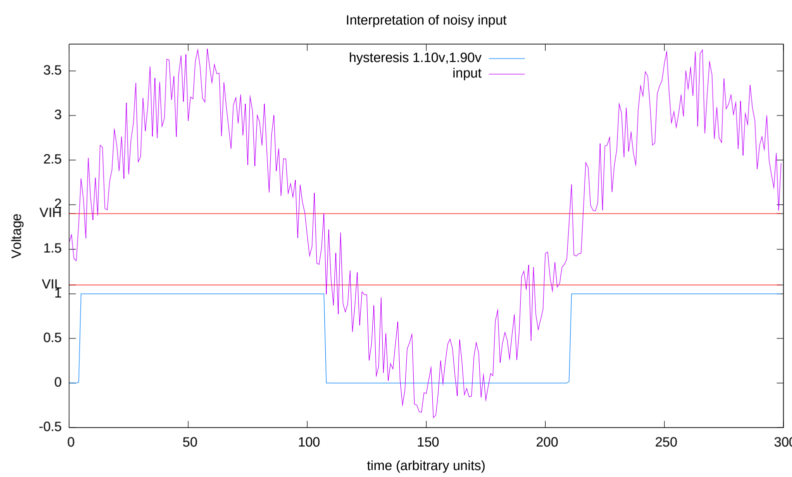

Figure 4: The results of interpreting a noisy input with the hysteresis of Figure 3. Note that the output does not wobble wildly when the input is near the threshold. One can get even cleaner transitions by increasing the gain (making the slope at the transition points steeper), and this is usually done in the design of Schmitt-trigger inputs, which are often used for converting such noisy inputs into clean digital signals.

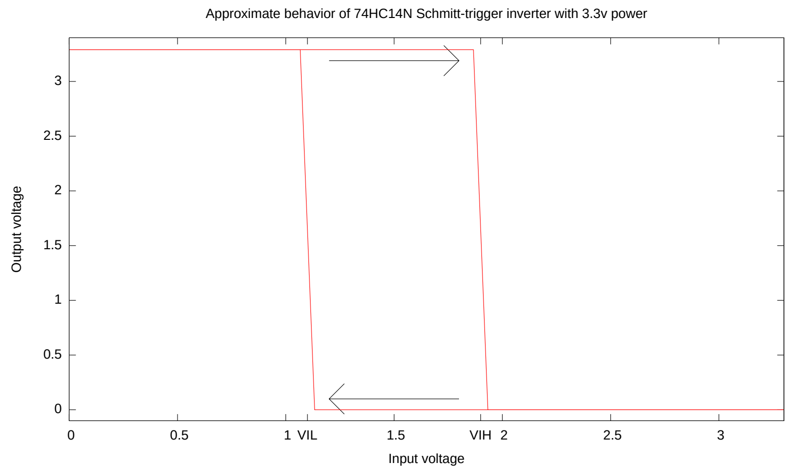

Figure 5: An approximate transfer characteristic for the 74HC14N operating with a power supply of 3.3V. On the actual device the gain is much higher, and the thresholds may be at somewhat different voltages.

The hysteresis curve for the 74HC14N does not look exactly like the curve in Figure 3. Not only are the thresholds different and the gain higher, but the 74HC14N is an *inverter*, which means that the output goes low when the input goes high and vice versa. A more realistic transfer characteristic for the 74HC14N operating on a 3.3V power supply is shown in Figure 5.

The 74HC14N is a cheap, fast part, but the threshold voltages V_{IL} and V_{IH} are not exactly constants—the spec for them allows a pretty wide range of values.

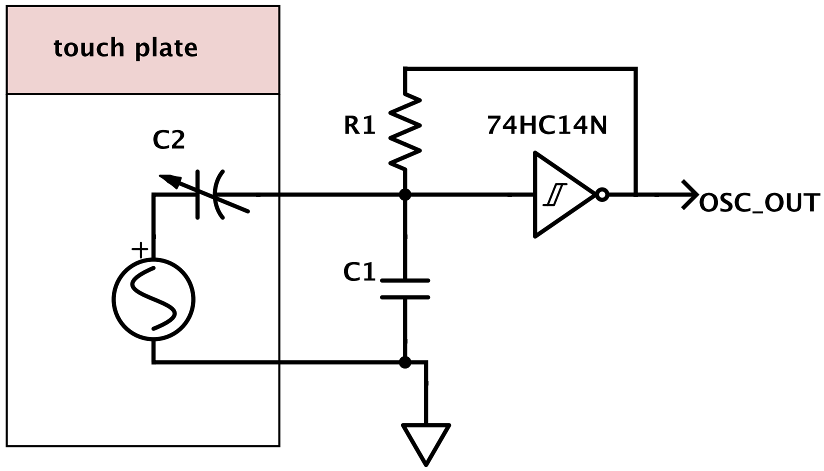

Figure 6: One way to implement an oscillator with Schmitt trigger inverters. When the output of the inverter is high, the capacitor C1 charges through resistance R1, until the voltage across C1 is larger than V_{IH} for the Schmitt-trigger inverter, then the output goes low and C1 discharges through R1 until its voltage drops below V_{IL} and the cycle repeats. The AC voltage source and variable capacitor model a person touching the capacitive touch sensor.

2.2 How a hysteresis oscillator works

Although the main use of Schmitt triggers (circuits that implement a hysteresis transfer characteristic) is to convert noisy analog inputs to clean digital inputs, they have another minor use as oscillators, which we will take advantage of in this lab to make an on-off sensor that is sensitive to changes in capacitance.

Figure 6 shows a typical circuit for a *hysteresis oscillator* (also called a *relaxation oscillator*). The principle of operation is simple: When the output of the inverter is high, the capacitor C1 charges through resistance R1, until the voltage across C1 is larger than V_{IH} for the Schmitt-trigger inverter, then the output goes low and C1 discharges through R1 until its voltage drops below V_{IL} and the cycle repeats. The time spent with the output high depends on how long it takes to charge C1 from V_{IL} to V_{IH}, with the output voltage at its high value V_{OH}. The time spent with the output low depends on how long it takes to discharge C1 from V_{IH} to V_{IL}, with the output voltage at its low value V_{OL}.

Note that the charge and discharge times depend on the voltages (V_{IL}, V_{IH}, V_{OL}, V_{OH}) and the RC time constant R_1C_1, but not on R_1 and C_1 separately. This means that we can design the hysteresis oscillator for a particular frequency with quite different component values.

What would be the effect of having a large value for R_1 and a small one for C_1, rather than a small value for R_1 and a large one for C_1? If R_1 is small and C_1 large, then the frequency will be fairly stable, not affected much by changes in C_1 or small amount of noise (like from nearby 60Hz power lines) capacitively coupled into the input. If R_1 is high and C_1 small, then the frequency will vary much more from capacitively-coupled noise, and will be very sensitive to changes in C_1.

Note that changes to the threshold voltages or to the high and low output voltages of the inverter also will result in changing charge and discharge times. These effects are probably less important than capacitively-coupled 60Hz noise.

2.3 Capacitance touch sensor

A capacitance touch sensor is used for switches on some instruments, because it has no moving parts (other than electrons) and can be put behind an easily-cleaned glass surface. It is also easy to operate with gloves on. Capacitive sensing can also be used in other contexts, to detect the presence of conductive substances without having to make direct electrical contact with them, or to measure distances between conductors.

The basic idea of this capacitance touch sensor is simple:

- A touch plate consists of one plate of a capacitor and an insulator—your finger is the other plate of the capacitor.
- The capacitance of the touch plate varies depending on how close the finger is to the touch plate and how much area of the finger is in contact.
- The varying capacitance is used to change the frequency of an oscillator.
- Because of the 60Hz energy all around, you act much like a low-voltage 60Hz voltage source, so the frequency of the oscillator is not constant, but fluctuates with 60Hz frequency modulation.
- The frequency (or period or duration of one of the pulses) of the oscillator is measured to determine whether the touch plate is currently being touched or not.

You will make a cheap capacitance touch plate out of a piece of aluminum foil (the plate of the capacitor) and a layer of packing tape (the insulator). You can connect this plate of the capacitor to the inverter input of the hysteresis oscillator, effectively putting the capacitor in parallel with C1. The period of the oscillator will then be proportional to $R_1(C_1 + C_2)$, rather than just R_1C_1, where C_2 is the capacitance of the touch plate and finger.

To detect the change in period easily, we want the change in C_2 to be fairly large relative to $C_1 + C_2$, which means we want to keep C_1 fairly small.

To detect the change in frequency of the oscillator, I wrote a small KL25Z program that you can use: `http://users.soe.ucsc.edu/~karplus/bme101/s14/freq_detector_own_isr.bin` (you can see the source code at `http://users.soe.ucsc.edu/~karplus/bme101/s14/freq_detector_own_isr.cpp`).

Since I do not know what frequency you are going to design your oscillator for, I tried to make the program adaptive and to use hysteresis!

The program expects a periodic digital signal on pin PTD4 with a frequency between about 400Hz and 800kHz. On reset, it displays a yellow light, then measures the frequency to store as the "off" frequency. You should not be touching the sensor when you reset the board.

If the frequency is out of range (say for a disconnected input), then the LED is set to red, and the off frequency checked again. Otherwise the LED is turned blue.

After initialization, if the program detects a frequency 20% less than the initial freq, it turns the light green, turning it blue again when the the frequency increases to 90% of the original frequency.

Frequency measurements are made by counting the number of rising and falling edges in one cycle of the mains frequency (1/60 sec), giving somewhat poor resolution at lower frequencies. The counting time is chosen to that frequency modulation by the mains voltages is averaged out.

As a debugging aid, the freq_detector_own_isr.bin program outputs the frequencies it uses as set points and the frequencies it observes to the OpenSDA USB port (the same USB port you use for downloading, not the native USB port used by PteroDAQ). You can look at this stream of information with the Arduino serial monitor, or any other program that can echo what is on a USB Serial connection. I used that feature to make a plot of the frequency of the oscillator with a particular set of R_1 and C_1 values (see Figure 7).

The hysteresis on the frequency detection in the program means that you need to design your capacitance touch sensor to have at least a 20% change in frequency when the touch plate is touched (more is better). You also need to design your oscillator to have a frequency between 400Hz and 800kHz when the sensor is not touched.

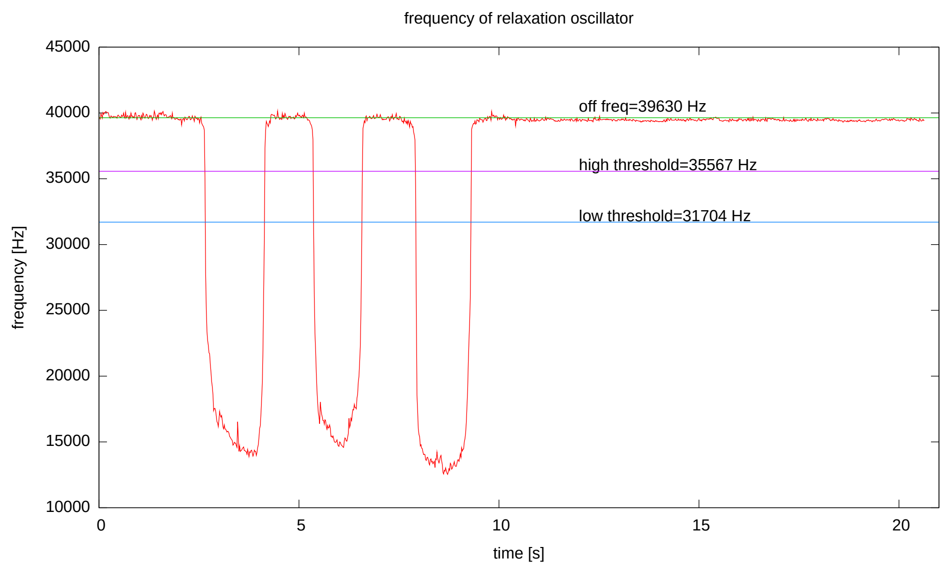

Figure 7: The noise in the frequency measurement is quite small compared with the thresholds at -10% and -20% of the off-frequency, and the change in frequency due to the touching and releasing the capacitive sensor is much larger than necessary for reliable detection.

3 Pre-lab assignment

Read the Wikipedia articles
`http://en.wikipedia.org/wiki/Capacitance`,
`http://en.wikipedia.org/wiki/RC_time_constant`,
`http://en.wikipedia.org/wiki/Relaxation_oscillator`,
`http://en.wikipedia.org/wiki/Touch_switch#Capacitance_touch_switch`, and
`http://en.wikipedia.org/wiki/Bypass_capacitor`. It is not necessary to understand everything in each of these articles, as some go into more depth than we need for this lab.

Determine how long a hysteresis oscillator will stay low given the four voltages (V_{IL}, V_{IH}, V_{OL}, V_{OH}) and the RC time constant R_1C_1. Turning this around, what range of RC time constants do you need to get a frequency of around 10kH—100kHz?

You will measure the voltages in the lab, but you might want to estimate the RC time constant range using the specs from the data sheet.

If putting your finger on the touch sensor is should double the duration of the pulse, then $C_1 + C_{2(\text{touching})}$ should be about twice $C_1 + C_{2(\text{not touching})}$ Estimate the capacitance of a finger touch on the packing-tape and foil sensor by estimating the area of your finger that comes in contact with the tape, and assume that the tape is 2mil tape (0.002" thick) made of polypropylene (look up the dielectric constant of polypropylene on line). Remember that capacitance can be computed with the formula

$$C = \frac{\epsilon_r \epsilon_0 A}{d} ,$$

where ϵ_r is the dielectric constant, $\epsilon_0 = 8.854187817E - 12F/m$ is the permittivity of free space, A is the area, and d is the distance between the plates.

Use your estimate of the capacitance of the finger touch to get initial values for R_1 and C_1. You don't have a value for $C_{2(\text{not touching})}$ yet, so for this pre-lab exercise, assume that it is small relative to C_1—this turns out to be a pretty good assumption for most values of C_1 that you are likely to choose.

4 Parts, tools, and equipment needed

Parts for this lab from kit:

- hysteresis oscillator board
- 74HC14N Schmitt trigger hex inverter
- resistor(s)
- capacitor(s)
- alligator clip
- KL25Z board
- USB A-miniB cable

Tools for this lab:

- wire cutters
- wire strippers
- long-nose pliers
- solder sucker

Equipment in lab:

- oscilloscope (to view oscillator output)
- frequency meter (optional)
- soldering iron

Parts provided by instructor:

- aluminum foil
- packing tape
- solder

5 Procedures

5.1 Characterizing the 74HC14N

To characterize the 74HC14N, we want 4 voltages: input thresholds V_{IL} and V_{IH} and output voltages V_{OL} and V_{OH}.

The output voltages are easy to measure: just connect +3.3V to pin 14 and GND to pin 7 of the chip (providing power to the chip), then connect either +3.3v or GND to pin 1 (the input of one inverter) and measure the output on pin 2 (the output of that inverter). Figure 8 identifies pins of a 14-pin DIP. Consult the spec sheet for the part to determine what the various pins do.

The input threshold voltages are a little harder to measure. One approach is to use one power supply to power the chip at 3.3v, and another to drive the input pin. Monitor the output pin with a voltmeter, oscilloscope, or LED with series resistor, and adjust the input voltage upwards until the output goes low, then downwards until the output goes high again. Sweep back and forth a few times, recording the measurements. This method takes advantage of having expensive power supplies in the lab, but requires manually recording the data.

You can use PteroDAQ to record the measurements by hooking up the input to an analog pin and the output to a digital pin, and triggering on both transitions of the digital pin. Then as you adjust the voltage up and down, PteroDAQ can record what the voltages were every time the output changed state. This method worked well for me—you don't even need a bench power supply, as you can use your 10k trimpot as a voltage divider to divide-down the 3.3v power supply from the KL25Z board to produce the input voltage. (That is, this can be done entirely at home with no bench equipment.)

Once you have the voltages V_{IH} and V_{IL}, check that your initial estimates of R_1 and C_1 based on your initial estimates of the voltages still seem reasonable. If not, adjust them.

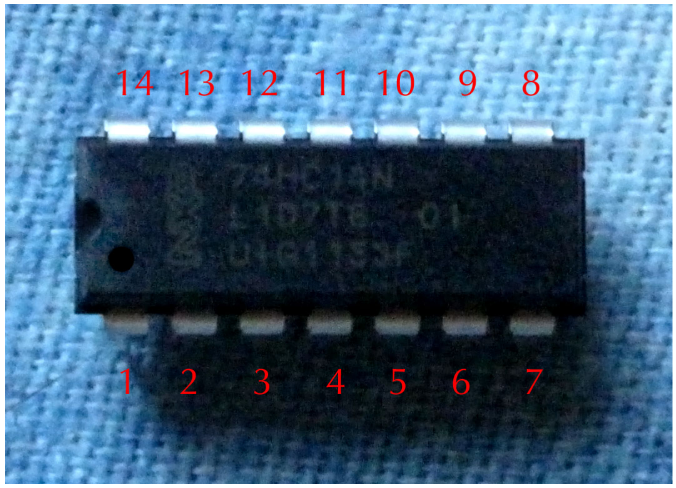

Figure 8: The numbering of pins on a dual-inline package starts by the dot and continues counter-clockwise around the package. Some packages have just a notch at the end with the lowest and highest pin numbers, but not a visible dot.

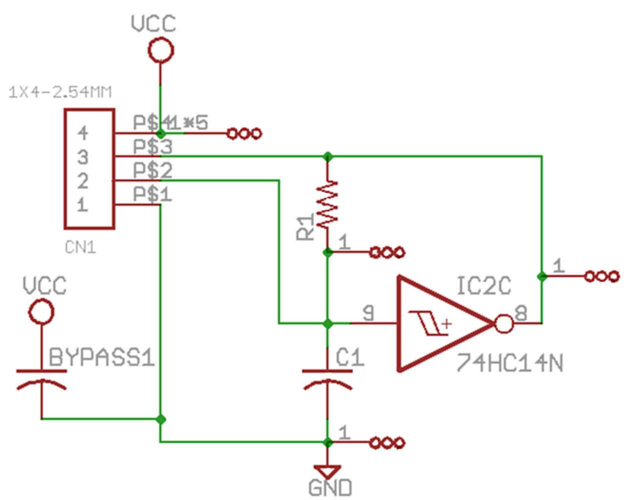

Figure 9: The oscillator schematic for the PC board, as drawn by the Eagle board-layout software. There is only one Schmitt-trigger inverter used.

5.2 Breadboarding the hysteresis oscillator

The hysteresis oscillator board that you will solder up does not have exactly the same circuit as Figure 6. The circuit that the board implements is show in Figure 9. The schematics drawn by the Eagle board-layout software are rather ugly, but I wanted to avoid the potential errors from copying the schematic using a different editor.

The circuit on the PC board has one extra component: a *bypass* capacitor to keep the fluctuations in current from the inverter chip from propagating too much noise into the power lines. See `http://en.wikipedia.org/wiki/Bypass_capacitor`. Although your circuit will work fine without the bypass capacitor, it is a good idea to get into the habit of including them—the 0.1μF capacitor (the largest of the little disk capacitors in your assortment) is a reasonable size, though I often like to use 4.7μF ceramic capacitors as the bypass capacitors.

Make your touch plate by folding a piece of aluminum foil into a neat rectangle about 2cm–3cm by 5cm–7cm. Then fold a 9cm–12cm-long piece of packing tape over the foil, so that the foil is covered with a single layer of tape

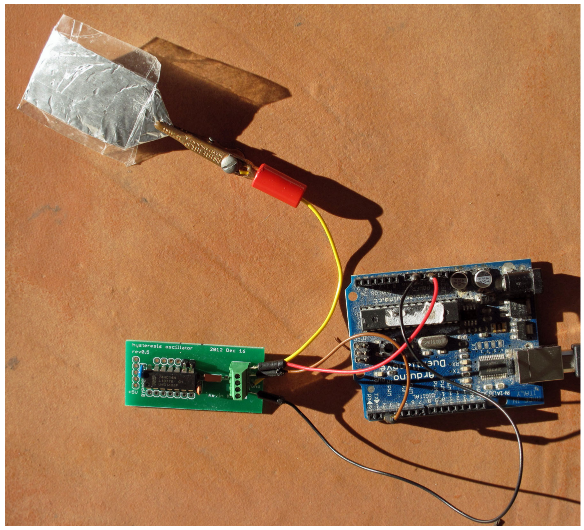

Figure 10: Touch sensor made with aluminum foil and packing tape, connected to the hysteresis oscillator and an Arduino board.

everywhere except at one end, where bare foil sticks out to provide a place for an alligator clip to attach. Having multiple layers of foil at the end that you will attach the clip lead makes the sensor sturdier.

Wire up your circuit on a breadboard (keeping the wires fairly short, to reduce stray capacitance), and connect the touch plate to the inverter input with a wire and an alligator clip. Power the circuit either from the KL25Z or from the bench power supply. For +3.3v and GND wires, remember that the standard convention is red for +3.3v and black for GND. Not following this convention will make it much harder for other people to help you debug your circuit.

Look at the output with an oscilloscope. Is it oscillating? How does the period of the oscillation change as you touch the touch sensor? If it is changing by less than 20%, try using a smaller capacitor for C_1. What is the period? If it is too short, try using a larger R_1 to increase the RC time constant. What happens if you leave the touch plate connected but remove C_1? What happens to the output of the oscillator if you look at the input to the oscillator with the oscilloscope?

Measure the frequencies or periods with different C_1 values, with the touch plate unconnected, connected but not touched, and connected but touched. (You may find the frequency meter handy for these measurements, but you can do it with just the oscilloscope.) Can you estimate the capacitance of the touch plate and the extra capacitance of the touch from these measurements?

Hook up the ground and output nodes of your circuit to GND and digital pin PTD4 of a KL25Z board with the freq_detector_own_isr.bin file downloaded. Without having your hand near the touch sensor, reset the KL25Z board by pressing its reset button. This should set the frequency thresholds in the program so that touching the touch plate changes the LED from blue to green, and releasing it changes it back.

5.3 Soldering the hysteresis oscillator

Put the components on the board in the right orientation. Follow tutorial instructions on how to solder the components in place. Be careful not to burn yourself with the iron!

Make sure that all your connections are shiny (not cold-soldered) and that no adjacent pins have been soldered together. If you do accidentally solder two points together, remelt the solder and use the solder sucker to remove the excess.

Use a multimeter to check that the +3.3V and GND terminals are not shorted together, then hook up wires to the KL25Z (or oscilloscope) and test as for the breadboard.

6 Demo and writeup

There are three checkpoints in this lab that should be demonstrated to an instructor:

1. breadboard hysteresis oscillator oscillating and displayed on the oscilloscope.
2. breadboard hysteresis oscillator working with KL25Z to control LED.
3. soldered hysteresis oscillator working with KL25Z to control LED. The instructor may wish to examine the solder joints on the back of the board.

Write up all the parameters that you measured, explaining briefly how they were determined.

Write up both your initial estimate of $C_{2(\mathrm{touched})}$ and improved estimates after you have measured the frequency of your oscillator under different conditions.

7 Design Hints

Everything in this lab comes down to setting the RC time constants appropriately for the touched and untouched sensor plate. Remember that all the times in this system (pulse widths or periods) are proportional to the RC time constant.