Capacitive Touch Sensor of a Simple RC Circuit

Ryan Edwards   Cameron Karabin   Chloe Li   Shreyas Musuku   Jake Schatz

***Abstract*—In this experiment, we designed and created a capacitive touch sensor using a graphite-drawn resistive strip, a 2MΩ resistor, and a human fingertip acting as a capacitor to charge the RC (resistor-capacitor) circuit and turn on an LED. Initially, we tried our circuit on a breadboard and used an Mbed microcontroller and Arduino Uno, but our design did not function correctly due to either code or faulty connections in our circuit. After being unable to successfully make the capacitive touch sensor using our hardware in the classroom, we pivoted to a Python simulation of the RC circuit and simulated the circuit as it was intended. In the simulation, touching the sensor increased charging time, allowing for threshold detection of turning on the LED. The LED was programmed to turn on when the graphite was touched, and this functioned successfully. While we were unable to make a functional capacitive touch sensor using hardware in the classroom, we successfully simulated the circuit using Python and validated the principles of capacitive sensing, emphasizing the role of RC time constants in real-world applications.**

***Index Terms*—RC Circuit, Capacitor, Time Constant, Python, Threshold Detection**

I. Introduction

A. Theory

A circuit is a connection of electrical components, transferring electrons in cyclic patterns. Driven by a voltage difference created by an electric field, electrons are pushed through conductive wiring in order to power electrical components throughout a closed circuit. In an RC (resistor-capacitor) circuit, electrons charge a capacitor over a period of time. The time constant of an RC circuit ($\tau = RC$) controls how fast a capacitor charges or discharges through a resistor. When a voltage source is applied to these circuits, the voltage across the capacitor follows an exponential curve described by the formula: $V(t) = V_0(1 - e^{-t/\tau})$. This behavior demonstrates how a capacitor charges over time and it show how voltage at certain times is relative to the time constant of the RC circuit. Changes in capacitance, such as when we placed our finger on the conductive graphite strip of our circuit, can be detected in the circuit, causing the LED to turn on. In our case, the human body acts as additional capacitance, boosting the system's total capacitance and increasing the time constant. By timing how long it takes the capacitor to charge, we were able to detect when someone touched the graphite strip and turn on the LED as a result of the strip being touched.

How

How much

B. Hypotheses

We hypothesized that the charging time of the RC circuit would increase when a finger touches the graphite due to the added capacitance from the human body. We also hypothesized that we would be able to find a threshold to determine whether the finger was a part of the circuit or whether the finger was not touching the graphite by comparing the measured charging time to this threshold value. To test these hypotheses, we first attempted to make the circuit using hardware from the classroom: an Mbed microcontroller and Arduino Uno. However, we faced issues with our hardware and eventually simulated the design of our RC circuit in Python. We measured and modeled the voltage across the capacitor while adjusting capacitance to imitate touching the graphite strip with a finger. By analyzing these simulated charging times and finding a detection threshold, we demonstrated how the characteristics of RC circuits were reflected in our capacitive touch sensor design.

not as important for a "design project" type article

For a design project needs to be very precise & repeatable

C. Materials

The materials used to set up this experiment were a paperclip, a breadboard, a large-value resistor, an Arduino, and an LED sensor. To create a conductive surface, we used a pencil and paper. For the second part of this experiment, we utilized a computer software program to simulate the expected phenomena.

II. Methods

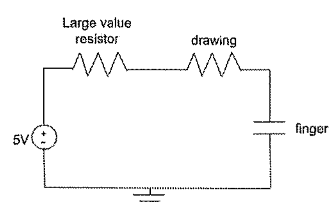


Figure 1: Circuit

could be a good possible pattern

A. Setup

To set up this experiment, we first drew an image on a piece of paper, using the pencil. At the bottom of the page, we drew a small rectangular box-like shape in order for the paperclip to be used as a conductive surface, connecting it to the fully interconnected drawing we made.

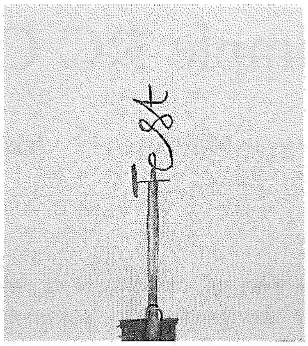
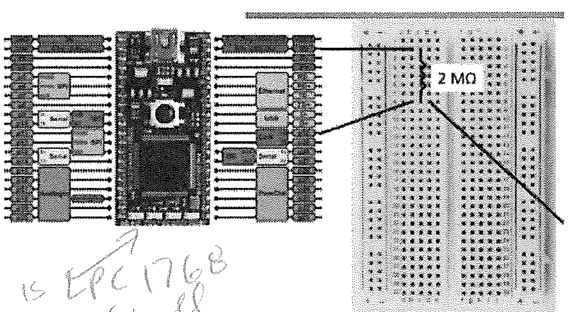


Figure 2: Pencil Drawing

B. mBed

Next, we were ready to build the circuit. At first, we attempted to build the circuit by using an mBed microcontroller. Using the TinkerCAD model in Figure 3, we created the circuit, connecting the voltage and CAN outputs to a resistor. We then connected the wire attached to the paperclip to the output of this simple circuit. We then incorporated an LED into this system to check if it would respond to the touch of our fingers. When we started to write the code, we ran into issues. We were under the assumption that since both the mBed and Arduino are both compatible with C++, they would be nearly identical. However, even though we were familiar with Arduino coding, we were not successful in making it, mainly because of the different library compatibility mBed offers. This caused us to switch to the Arduino setup.

X



This is LPC1768 specifically

Figure 3: Diagram of the setup

need proper schematic

C. Arduino

As can be seen in Figure 4, we attempted to pivot to using an Arduino, as we were all familiar with Arduino code. When attaching to the pins, we made the mistake of trying to find the compatible pins to the mBed system, rather than doing the Arduino pins from scratch. So, the model was completely built and coded, but when it came to testing, it did not prove to be successful because we believe that we converted the pins from the mBed to the Arduino faultily.

Digital out
+ Digital In / interrupt capable timer pin

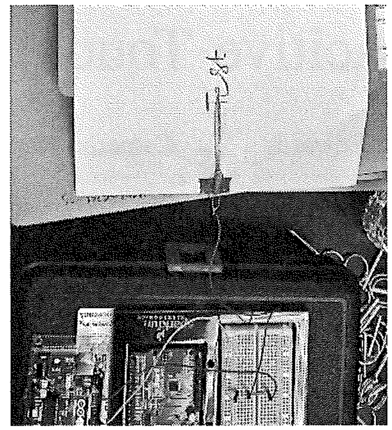


Figure 4: Original Arduino Set Up

D. Python

```
# Define the new detection threshold
threshold_voltage_adjusted = 4.8

# Step 1: Simulate with a 5ms time range
time_ms_5 = np.linspace(0, 5, 100)  # 0 to 5ms, 100 points

# Compute time to reach 4.8V with a 5ms window
time_to_threshold_no_touch_5ms = time_to_threshold(V_s, R, C_no_touch, time_ms_5, threshold_volt
time_to_threshold_with_touch_5ms = time_to_threshold(V_s, R, C_with_touch, time_ms_5, threshold_

# Step 2: If "With Touch" doesn't reach 4.8V in 5ms, extend to 10ms
if time_to_threshold_with_touch_5ms is None:
    time_ms_10 = np.linspace(0, 10, 150)  # Extend to 10ms, 150 points
    time_to_threshold_no_touch_10ms = time_to_threshold(V_s, R, C_no_touch, time_ms_10, threshol
    time_to_threshold_with_touch_10ms = time_to_threshold(V_s, R, C_with_touch, time_ms_10, thre
    time_ms_final = time_ms_10
    time_to_threshold_no_touch_final = time_to_threshold_no_touch_10ms
    time_to_threshold_with_touch_final = time_to_threshold_with_touch_10ms
else:
    time_ms_final = time_ms_5
    time_to_threshold_no_touch_final = time_to_threshold_no_touch_5ms
    time_to_threshold_with_touch_final = time_to_threshold_with_touch_5ms

# Generate simulated data for the final time range
data_final_threshold = {
    "Time (ms)": time_ms_final,
    "Voltage (No Touch)": simulate_capacitor_charge(V_s, R, C_no_touch, time_ms_final),
    "Voltage (With Touch)": simulate_capacitor_charge(V_s, R, C_with_touch, time_ms_final),
}
```

Figure 5: Code

code?

To overcome these challenges, we implemented a Python-based simulation to model the expected behavior of the capacitive touch sensor. The simulation followed the exponential charging curve of the capacitor, allowing us to analyze how voltage changed over time. We incorporated different capacitance values to represent both the "No Touch" and "With Touch" conditions, demonstrating how the human body increased the overall capacitance of the system. By measuring the time required for the capacitor voltage to reach 4.8V, we established a reliable detection threshold for touch sensing. We then plotted the simulation results to provide a visual representation of voltage trends over time and determine the relationship between capacitance and charging delay.

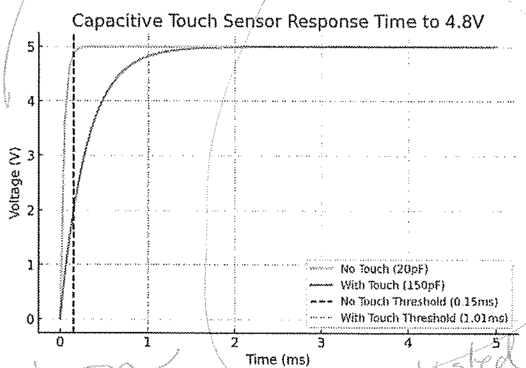
III. Results

Time (ms)	Voltage (No Touch)	Voltage (With Touch)
0	0	0
0.5050505051	4.999983577	4.071387527
1.01010101	5	4.827535775
1.515151515	5	4.967969514
2.02020202	5	4.994051218
2.525252525	5	4.998895177
3.03030303	5	4.99979461
3.535353535	5	4.999961892
4.04040404	5	4.999992922
4.545454545	5	4.999998686
5	5	4.999999711

Figure 6: Data Table

To create the graphs measuring different voltages based on the capacitance, with and without our fingers touching the for this logic circuit, what is H voltage?

model, we measured the voltage gain over time, analyzing the smoothness of the curves and the 'No Touch Threshold'.



Unlisted

Figure 7: Graph of Capacitive Touch Sensor Response Time

Figure 7 shows the charging behavior of the capacitor in the RC circuit under two different conditions: No Touch (20pF capacitance) and With Touch (150pF capacitance). The x-axis represents time in milliseconds (ms), while the y-axis represents voltage (V) across the capacitor.

This visual shows us how the potential changes as it gets further from the charge. The electric field closer to the origin has a greater magnitude and it exponentially decreases as it gets further away. The change in potential will have the same magnitude but the opposite sign, which means it increases exponentially as it gets closer to the origin. The No Touch curve (yellow) shows that when there is no external capacitance from a finger, the capacitor charges very quickly, reaching the detection threshold of 4.8V in approximately 0.15ms (marked by the blue dashed vertical line). In contrast, the With Touch curve (orange) demonstrates that when a finger touches the graphite strip, the increased capacitance slows down the charging rate, causing the capacitor to reach 4.8V at approximately 1.01ms (marked by the orange dashed vertical line). This delay is a key feature of capacitive touch sensing, as it provides a measurable distinction between touch and no-touch conditions.

IV. Discussion

A. Graph Analysis

The exponential nature of both curves in Figure 7 follows the expected RC charging equation, confirming that the circuit behaves as predicted by theoretical models. The significant difference in response times between the two conditions validates the feasibility of using this approach for touch detection, as a microcontroller can reliably measure the delay and determine when a finger is present on the sensor. The experimental data visualized in the graph demonstrates the fundamental principles of capacitor charging and how capacitance affects response time. According to the differential equation: $V_c(t) = V_S(1 - e^{-t/RC})$, the voltage across the capacitor rises exponentially toward the voltage source, growing based on the RC time constant. The graph underlines the fact that capacitance increases due to the simulated human touch. The charging curve becomes significantly slower, meaning it takes longer for the voltage to reach a given threshold. This is because a larger capacitance requires more time to accumulate charge. The threshold lines further illustrate this effect: in the no-touch case, the voltage reaches the threshold in approximately 0.15 ms, whereas in the touch case, it takes around 1.01 ms. This difference in response time is a key factor in capacitive touch sensing, as the system detects touch events by measuring variations in the time it takes for the capacitor to charge. Thus, the data effectively demonstrates the simulation's accuracy when it comes to depicting the physical concepts of exponential charging, capacitance, and resistance in an electrical circuit in touch detection technology.

B. Hardware Failure Analysis

Looking at the simulation results and how it seemed to succeed, we found certain inconsistencies in our own design. First, for the resistance, we believe that we used too small of a resistance value. Even though we aimed to use a $2M\Omega$ resistor, that was unavailable to us, so we tried significantly smaller resistance values like $10K\Omega$, $100K\Omega$, and $200K\Omega$. be specific. We believe that this may have caused a faulty circuit because even though the resistance value was high, it did not meet the requirements the circuit demanded it to. It caused issues with how fast the capacitor charged, which reduced the sensitivity of the sensor to human touch, how much current was flowing from the power source, and difficulty for the Arduino to tell between a touched and untouched sensor. Additionally, we believe that the switch from the mBed hardware and software to the Arduino hardware and software was a major issue. Rather than trying to implement a new hardware wiring system with the Arduino, we tried to convert the connections from the mBed hardware system. Because of this, the Arduino software we programmed was compatible to work with our mBed-designed Arduino hardware.

V. Conclusion

Through this experiment, we explored the principles of RC circuits and capacitive sensing by attempting to design a capacitive touch sensor using a graphite-drawn resistive strip, a $2M\Omega$ resistor, and a human fingertip as additional capacitance. Although our initial efforts to create a capacitive touch screen were unsuccessful, we were able to simulate the same results using Python. The simulation confirmed our hypothesis that the charging time of the RC circuit increased when a finger touched the graphite strip, allowing for threshold-based detection of touch. This demonstrated the fundamental relationship between capacitance and charging time in an RC circuit, reinforcing the concept of time constants in capacitive sensing applications. Although we could not achieve a working physical prototype, our simulation illustrated the core principles of capacitive touch sensing, emphasizing the importance of RC time constants in real-world electronic applications.

Contributions

The contributions of each author to this work are as follows: Jake was responsible for the abstract and introduction.

Cameron assisted in drafting the abstract and methods. Ryan contributed to the methods, discussion.. Shreyas created the simulation and gathered the results. Chloe contributed to the conclusion and discussion sections. All authors reviewed and approved the final paper.

REFERENCES

[1] "Capacitor - an Overview — ScienceDirect Topics." Www.sciencedirect.com, www.sciencedirect.com/topics/materials-science/capacitor.

[2] Ahmad, Amna. "Understanding RC Circuit Operation and Time Constant." Eepower.com, EEPower.com, 31 Mar. 2023, eepower.com/technical-articles/understanding-rc-circuit-operation-and-time-constant/.