Tony Gray

Projected Capacitive Touch

A Practical Guide for Engineers

Springer

Projected Capacitive Touch

Tony Gray

Projected Capacitive Touch

A Practical Guide for Engineers

Springer

Tony Gray
Plano, TX, USA

ISBN 978-3-319-98391-2 ISBN 978-3-319-98392-9 (eBook)
https://doi.org/10.1007/978-3-319-98392-9

Library of Congress Control Number: 2018954649

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

About the Author

Tony Gray Tony has been designing, tuning, and testing projected capacitive touch panels since 2009. In his role as Director of PCAP Touch Technology first at Ocular LCD and then at Dawar Technologies, he has been responsible for over 100 custom projected capacitive touch panel design projects. Tony is highly regarded in the marketplace for his knowledge of touch systems. He has presented at several different technical conferences on projected capacitive touch technology and published numerous articles and white papers. Before working in the touch industry, Tony spent 15 years writing embedded software for a variety of devices including residential thermostats, smoke opacity meters, hospital food delivery systems, robotic assay equipment, and a system used to capture and analyze 3D neuronal structures. Tony has a Bachelor of Science in Computer Engineering from Lehigh University. He lives in Plano, Texas.

Chapter 1
Introduction

On June 29, 2007, the world changed forever. That was the day the first-generation iPhone® was released to the public. There are several reasons that the iPhone had such a huge impact. It redefined what a cell phone could do. It looked sleek, modern, and (mostly) seamless. It had a high-resolution full color display. It included a fully functional music player. It had an eco-system of custom applications. But the most important thing about the iPhone was that it could recognize two separate touches simultaneously. That may not sound like a major accomplishment today, but in 2007 it was revolutionary.

Is that picture of Grandma and the kids too small? Just use two fingers to zoom in and see the detail. Want to see how close Burkburnett Texas is to Dallas? Just use two fingers to zoom out until you can see both. Is that map of the zoo not oriented to the direction you are walking? Just use two fingers to rotate it so that it lines up.

Before the iPhone, the touch market was dominated by resistive touch sensors. Resistive touch sensors were cheap, but they also scratched easily, had to be replaced often, had to be calibrated on a regular basis, required a clunky bezel design, and only supported one touch (there were some multi-touch resistive solutions on the market, but they never worked well enough to capture significant market share). Several other touch technologies also had significant market share including surface acoustic wave and infrared, but resistive was the market leader for most applications. But once the iPhone was released, it established a new expectation for the touch experience.

I do not mean to imply that Apple® invented multi-touch. There were several multi-touch technologies available going back to the 1980s. Apple did not even invent projected capacitive multi-touch. That had been around since at least 1984 when it was used to create a 16 key capacitive touch pad, although it was not called "projected capacitive" until 1999.

Apple's true innovation was not the technology, but the use of it. The iPhone gave you a reason to need multiple touches. Once users had this capability and understood how useful and intuitive it was, they began to expect it on other devices as well from

T. Gray, *Projected Capacitive Touch*, https://doi.org/10.1007/978-3-319-98392-9_1

thermostats to exercise bikes to infusion pumps to slot machines to pretty much everything.

While the iPhone deservedly gets a lot of credit for introducing projected capacitive (PCAP) touch technology to the world, there were several other factors that also contributed to the rise of PCAP. Full color displays started getting less expensive, allowing designers to replace monochrome, segmented displays with full color bitmap displays. Processors continued to get less expensive, which was important because it is just about impossible to drive a 320 $\times$ 480 256 color TFT with an 8051 embedded processor and no operating system. Speaking of operating systems, another important factor was the continued growth of Linux in the embedded systems market. All of these changes fed into one another allowing product designers to put together inexpensive full color displays, a free operating system, and a high-end processor capable of displaying a multicolor GUI.

When all this upheaval started happening in 2007, I was working for Invensys Controls writing software for residential thermostats using a segmented LCD, a 16-bit processor, and no operating system. In 2007, our design team finally convinced marketing that we should develop a high-end thermostat with a 5.7in TFT and a resistive touch sensor. This was my first glimpse of the changes coming in product design.

In 2008, I left Invensys Controls to work for Ocular LCD. Ocular had been very successful supplying custom segmented LCDs to various embedded markets. But fairly soon after I joined the company, it became obvious that segmented LCDs were going to be replaced by full color TFTs in just a few years. My boss Larry Mozdzyn, the CTO and one of the co-founders of the company, realized that the same equipment we used to build LCDs could also be used to make projected capacitive touch sensors. Over the next few years Ocular transformed itself from an LCD supplier to a major supplier of PCAP touch sensors. And fortunately for me, Larry pulled me into the PCAP side of the business.

At the time I did not know anything about PCAP, so I did what any nerd would do: I Googled “how does projected capacitive touch work.” I found a few web sites that did not go into a lot of detail. Then I tried searching Amazon® for a book on PCAP. I was a bit surprised that I could not find one, although this was not too shocking since PCAP was just starting to become popular. I attended trainings offered by the touch controller vendors we used at the time, Cirque® and Atmel®. I asked *a lot* of questions (no doubt to the point of annoying the engineers leading the trainings). I read every bit of documentation I could get. I printed out the documents, wrote notes on each page, and then e-mailed dozens of questions to the support teams at Cirque and Atmel. I wrote a lot of utilities for the controllers including configuration apps, test apps, programming apps, and even tuning wizards that automated the tuning process. Eventually I developed an understanding of PCAP including the fundamental physics of the technology, the operating principles of a touch controller, communication protocols and driver support, accuracy and linearity testing, physical construction, and manufacturing.

Occasionally, I would do another search on Amazon looking for the definitive book on projected capacitive touch technology, but I never found one. I had written a few articles over the years, some on embedded software development and some on PCAP. Eventually, it occurred to me that if nobody else was going to write a book on PCAP, I should do it.

This book is essentially the book I wish I had found on Amazon in 2008. It explains just about everything there is to know about projected capacitive touch that is not proprietary (if you are looking for a book to explain the secrets of the *insert-name-here* touch controller, sorry but NDAs prevent me from getting into that kind of detail). This book starts with a basic explanation of how a projected capacitive touch sensor works. It explains mutual vs. self capacitance, how capacitance is measured, how a touch sensor can be modeled as an electronic circuit, optical quality metrics, transparent conductors, and so much more (check the Table of Contents for a complete list of topics). In short, this book includes just about everything I have learned about PCAP over the last 10 years that I am allowed to share. While it is targeted at mechanical, electrical, and software engineers who are considering using PCAP in a project, I am hopeful that it will also prove useful to a wide variety of people from sales and marketing to PCAP designers to physics students to nerds who just want to learn more (my people). I hope you all enjoy reading it as much as I did writing it.

Chapter 2
Projected Capacitive Touch Basics

A capacitor is an energy storage device, a very simple form of battery. We form a capacitor by placing two conductive plates very close to each other (Fig. 2.1).

One plate of the capacitor is typically connected to ground and the other plate is connected to a voltage source through an in-line resistor. Over time the capacitor charges up to the applied voltage according to the equation:

$$V_{\mathrm{CAP}} = V_{\mathrm{IN}} \left(1 - e^{\frac{-t}{RC}}\right) \tag{2.1}$$

t is time in seconds, R is the value of the resistor in Ohms, and C is the value of the capacitor in Farads. The voltage on the capacitor, V_{CAP}, charges up at an exponential rate (Fig. 2.2).

Notice that the axes on Fig. 2.2 do not have a scale. The general charging curve for a capacitor always looks like the graph shown. The Y scale is based on the charging voltage. The capacitor voltage approaches the applied voltage asymptotically (i.e., it will take an infinite time to reach the applied voltage). In theory, a capacitor can be charged to any voltage. In practice, there are physical constraints on the maximum voltage a particular capacitor can be charged to before breaking down, exploding, or failing in some other catastrophic and exciting way.

The scale of the X or time axis depends on the value of the capacitor which depends on the physical characteristics of the capacitor (it also depends on the resistor, but for now we are considering it to be a constant). In an ideal capacitor the important characteristics are the area of the plates, the distance between the plates, and the dielectric of the material between the plates. Think of the dielectric as a measurement of how well an electric field propagates through the material. Some materials do a better job of propagating electric fields and thus have a higher dielectric. Some materials do a worse job and have a lower dielectric.

The capacitance can be calculated from the area, distance, and dielectric (ε) using Eq. 2.2.

T. Gray, *Projected Capacitive Touch*, https://doi.org/10.1007/978-3-319-98392-9_2

Fig. 2.1 Capacitor

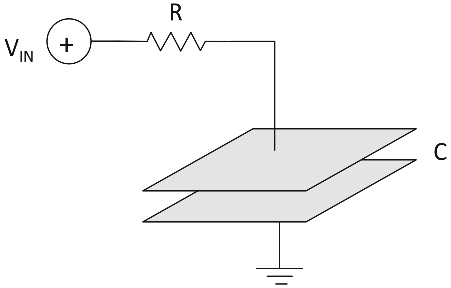
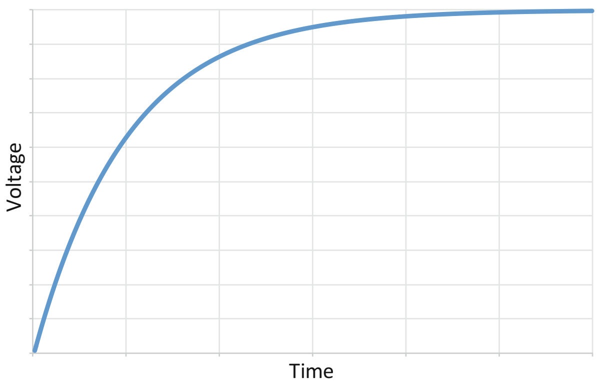


Fig. 2.2 Capacitor charging



$$C = \varepsilon \frac{A}{d} \tag{2.2}$$

The units for capacitance are Farads. One Farad is defined as the capacitance needed to store one Coulomb of charge with one volt across the plates.

A smaller capacitor charges very quickly; a larger capacitor charges very slowly. Most capacitors used in electronics have very small values in the micro Farad (10^{-6}) to nano Farad (10^{-9}) range. In projected capacitive touch sensors, we have to measure capacitances in the pico Farad range (10^{-12}).

A useful analogy is to think of the capacitor as a coffee mug and the current from the voltage source as coffee being poured into the mug (Fig. 2.3).

The applied voltage (V_{IN}) is the height of the mug, and the current height of the coffee in the mug is the current voltage on the positive plate of the capacitor (V_{CAP}). As you pour the coffee into the cup, you naturally begin to slow the rate of pouring as the cup gets more full. The capacitor is fully charged when the cup is completely filled.

If the mug is small, it gets filled very quickly. If the mug is large, it gets filled more slowly. The resistor in the circuit controls how fast the coffee comes out of the pot. Once the mug is full, you cannot fill it up any further and the flow of coffee stops. To extend the analogy a bit further, there is always some small leakage current from a capacitor. Think of this as a tiny hole in the bottom of the mug that drips

Fig. 2.3 Charge time analogy

coffee out very slowly. If you stop pouring coffee in the mug and wait long enough, the mug will eventually empty by itself. The same thing happens to capacitors. Once you charge a capacitor, it will slowly discharge over time.

As we pour coffee into the mug, we can measure the pour rate (current), the height of the coffee in the mug (voltage on the positive plate), and how long it takes to fill the mug (charging time). And of course we already know the maximum amount of coffee we can fit (applied voltage).

We said before that the capacitor never quite gets to the applied voltage level (i.e., the mug never completely fills up). So how do we determine when it is full enough? We need some standard way to define how full our capacitor is. To simplify matters, it would be great if the thing we use to measure how full it is could be ratiometric, like 50% or 70% of the full charge. That will allow us to have a standard metric for our discussions that we can all agree on, but that does not depend on the value of the capacitor or the voltage we are using.

The metric we use is called the time constant and is identified by the Greek letter Tau (τ). The time constant is defined as:

$$\tau = RC \tag{2.3}$$

The units for the time constant, τ, is seconds. That implies that multiplying resistance times capacitance gives time, which can be a bit surprising if you have never seen this equation before. We can see how this works out if we look at resistance (Ohms) and capacitance (Farads) in fundamental units:

$$\text{Ohms} = \frac{\text{kg} * \text{m}^2}{s^3 * A^2} \tag{2.4}$$

Table 2.1 Time constant levels

Tau	Equation	Charge level
1	$1 - e^{-1}$	63%
2	$1 - e^{-2}$	86%
3	$1 - e^{-3}$	95%
4	$1 - e^{-4}$	98%
5	$1 - e^{-5}$	99%

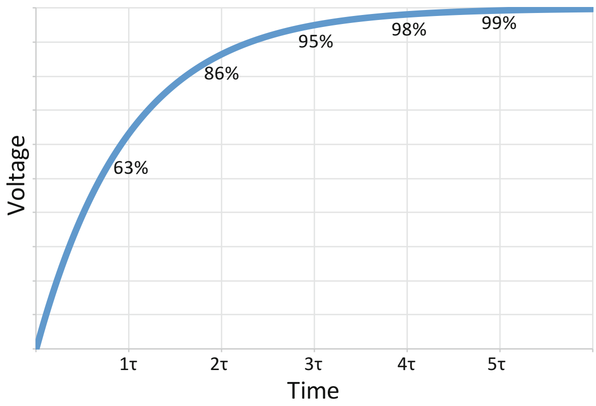


Fig. 2.4 Charge levels for τ

$$\text{Farads} = \frac{s^4 * A^2}{\text{kg} * \text{m}^2} \tag{2.5}$$

$$\text{Ohms} * \text{Farads} = \frac{\text{kg} * \text{m}^2}{s^3 * A^2} * \frac{s^4 * A^2}{\text{kg} * \text{m}^2} = s \tag{2.6}$$

Looking back at the equation for charging a capacitor:

$$V_{\text{CAP}} = V_{\text{IN}} \left(1 - e^{\frac{-t}{RC}}\right) \tag{2.7}$$

It is clear that the shape of the charge graph is defined by the values R and C. For that reason, τ is often called the RC time constant and the charging equation is often written like Eq. 2.8:

$$V_{\text{CAP}} = V_{\text{IN}} \left(1 - e^{\frac{-t}{\tau}}\right) \tag{2.8}$$

The standard way of defining the charge time is to use a multiplier of τ, as in 2τ, 3τ, 4τ, etc. Since τ is RC, one τ is $1*RC$, two τ is $2*RC$, and so on. So 2τ means when $t = 2 * RC$, or when two times RC seconds has passed. Table 2.1 shows the correspondence between different multipliers of τ and the actual charge level as a percentage of the applied voltage.

Figure 2.4 shows the charge graph for charge levels 1τ to 5τ.

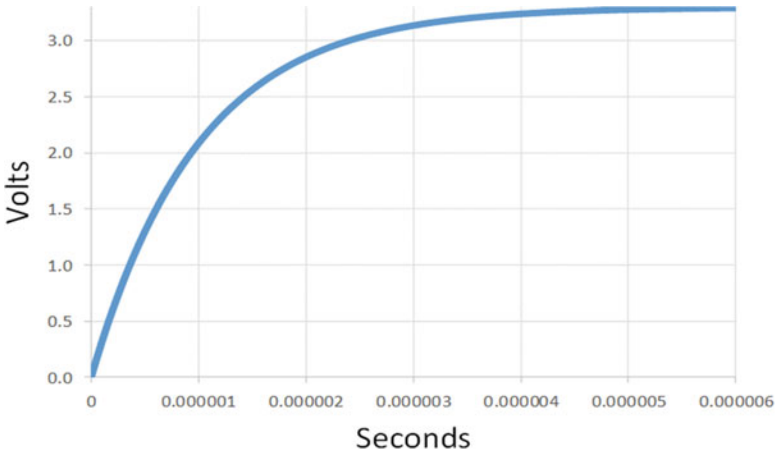


Fig. 2.5 Charge time for 10 pF and 100 kΩ

Let's start using a real example and come up with some real numbers. We will learn later that the capacitance values we are interested in are on the order of 1–10 pF (pico is 10^{-12}), so we will use a capacitor of 10 pF with a resistance of 100 kΩ and a charging voltage of 3.3 V. For this example, the charge graph looks like Fig. 2.5.

A capacitor is generally considered fully charged at 5τ or 0.9913 * 3.3 V = 3.27129 V. Our 5τ time for this circuit is 5 * 10 pF * 100 kΩ = 0.000005 s or 5 µs.

Now imagine that we do not know the capacitance and we want to determine it. This is an important point because the whole idea behind projected capacitive touch sensors is that we need to measure an unknown capacitance. The easiest way to do that is to use the charge time of the capacitor to determine its value. There are a few practical ways we can do this. One way would be to apply the 3.3 V, measure the amount of time it takes to charge up to 5τ, reverse Eq. 2.8, and use it to calculate the capacitance:

$$C = \frac{-t}{R * \ln\left(1 - \frac{V_C}{V_{IN}}\right)} \tag{2.9}$$

As an example, let's use the same numbers we found above for when our 10 pF cap is charged to 5τ:

$$C = \frac{-0.000005}{100000 * \ln\left(1 - \frac{3.27129}{3.3}\right)} = 10 * 10^{-12} = 10\text{ pF} \tag{2.10}$$

While the math for this method is simple, the implementation is difficult. With the various setups, multiple sampling, averaging, etc. that we need to do, this measurement might take 1 µs or it might take 100 µs depending on the capacitance. It is

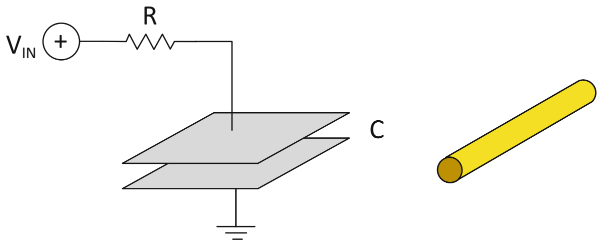


Fig. 2.6 Capacitor with grounded rod

difficult to write reliable embedded software when the time for a measurement cycle is unknown. And since we are eventually going to use this to locate fingers on a touch sensor, we do not want the time between our touch reports to vary based on the distance of the touch from one of the capacitors.

Another option is to charge the capacitor for a set time, then measure the voltage on the charging plate. This has the benefit of always taking the same amount of time. And we can solve it using the same equation as before. For example, if we measure the voltage on the capacitor 1.85 μs after connecting the voltage source to the plate, we measure 2.78 V. Our calculation becomes

$$C = \frac{-0.00000185}{100000 * \ln\left(1 - \frac{2.78}{3.3}\right)} = 10 * 10^{-12} = 10\text{ pF} \tag{2.11}$$

While measuring the voltage at the positive plate works from a theoretical point, it does not really work with the physical construction of a touch sensor. Once we have discussed how a touch sensor is actually built, we will come back to the issue of how we actually determine a capacitor's value.

Now that we have a basic understanding of how to measure the value of a capacitor, let's put it to work to do something useful. We will start with a question: What happens if we place a grounded copper rod next to the capacitor? (Fig. 2.6).

When we begin dumping charge into the capacitor, some of that charge is going to couple to the grounded rod and disappear into ground. Going back to the coffee mug analogy, the grounded rod essentially acts like a second coffee mug. Most of the coffee goes into the first mug, but some is diverted into the second mug (Fig. 2.7).

Because some of the charge is diverted, it takes longer to charge up the capacitor. And since it takes longer to charge up the capacitor, the charge graph now looks like Fig. 2.8.

If we use the same measurement technique to determine the capacitance, we find that the value of the capacitor has increased (i.e., the coffee mug appears bigger than it actually is). What is really going on is that we have added two new capacitors. The first is from the positive capacitor plate to ground, and the second is from the bottom capacitor plate to ground. This will be explained further when we get into the equivalent circuits involved in a touch sensor. For now, just think of it as the capacitor value increasing.

Fig. 2.7 Capacitor with grounded rod analogy

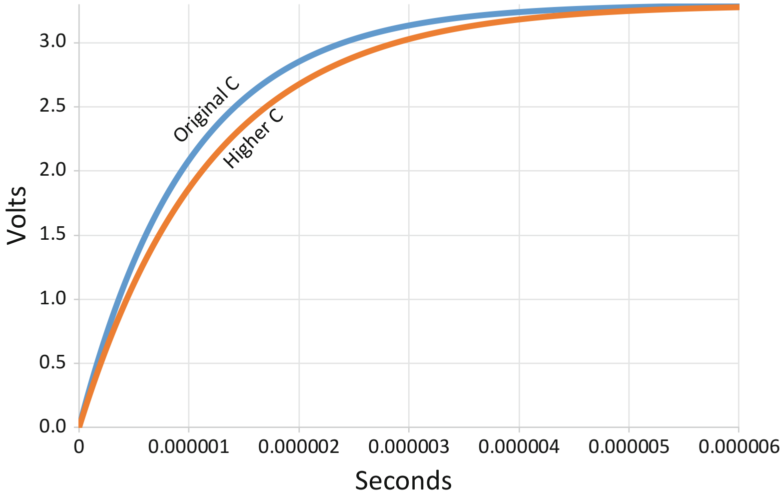


Fig. 2.8 Charge time with grounded rod

When the rod is closer to the capacitor, more charge is lost to ground, it takes longer to charge up the capacitor, and the measured capacitor value increases. When the rod is further away, less charge is lost and the measured capacitor value decreases. In theory, we can use this change in capacitance to calculate the distance between the metal rod and the capacitor (Fig. 2.9).

We just need to know the value of the capacitor without the metal rod in place, then the value of the capacitor with the metal rod in place. In practice, this is actually

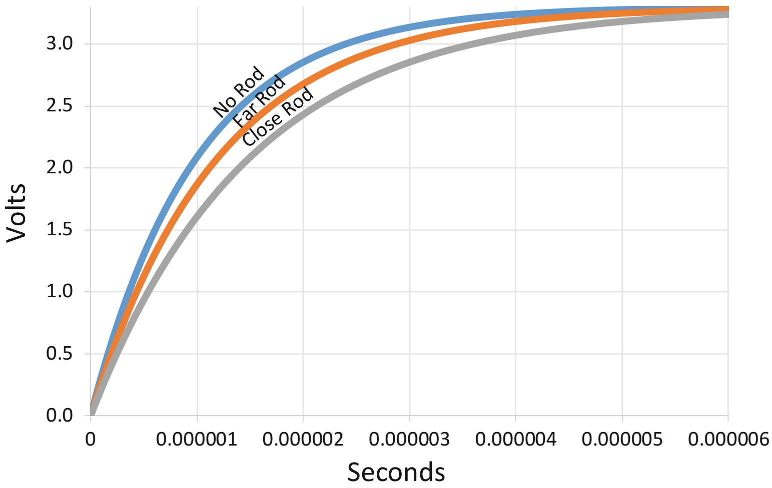


Fig. 2.9 Charge time varies by distance

quite difficult due to the complex interaction of the electric fields involved. But the basic idea, that we can use the change in capacitance to determine the relative distance between the capacitor and the rod, is true regardless of the messiness of the math.

When we say the metal rod is grounded, what does that mean? When something is truly grounded, it is literally inserted into the Earth. The reason for this is that, in theory, the Earth can supply a nearly infinite number of electrons, or act as a sink for a nearly infinite number of electrons (which can also be thought of as a nearly infinite number of electron holes). In reality, of course, the number of electrons or holes is not infinite, but it is much, much larger than the typical electrical circuit needs. In essence, by saying something is grounded we are saying that the ground can supply all the electrons or all the holes needed for the circuit to do whatever it is going to do.

When a printed circuit board is built, it typically has a ground plane which is a large copper area. This copper area is usually not connected to Earth ground (it may be connected to a grounded outlet which is connected to a power framework in the building which is connected to a terminal box which is eventually connected to the Earth, but from a practical stand point the copper is so far from Earth that it does not really matter). Even though the copper plane is not connected directly to Earth, it has way more electrons and holes than the circuit could ever possibly need assuming the circuit is operating normally. To the circuit, the copper plane looks like ground. This is called a virtual ground because it can supply all of the electrons or holes needed even though it is not actually Earth ground.

Now for the interesting part: a human body can also act as a virtual ground. For small enough circuits, the human body can supply enough electrons or holes that it appears to be ground. And since the body is a virtual ground, it can also couple to a circuit through an electric field. In other words, if we take away the metal rod and put

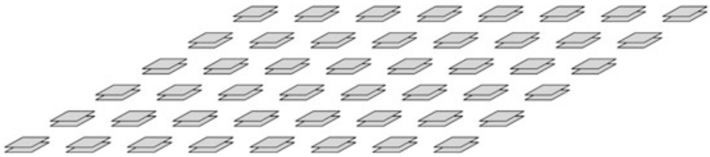


Fig. 2.10 Grid of capacitors

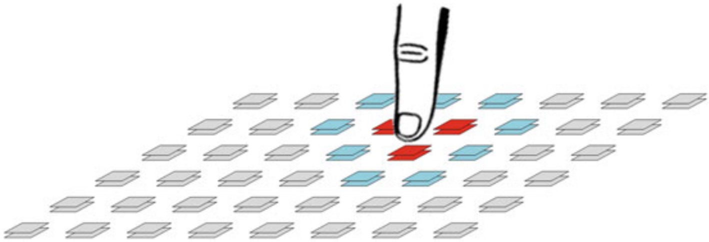


Fig. 2.11 Grid of capacitors with finger

our finger near the capacitor described above, some of the applied voltage will couple to our finger which changes the measured capacitance. We can use that change in capacitance to calculate the distance from the capacitor to our finger.

Next, imagine that we create a grid of capacitors as shown in Fig. 2.10 (the resistors and wires are omitted for clarity).

Each capacitor will have a slightly different value due to physical differences, proximity to the other capacitors, etc. The capacitors on the edges will have very different values since they are not surrounded by other capacitors. This will be important later when we talk about edge effects on touch sensors.

We measure the capacitance of each capacitor in the grid. Then we place a finger somewhere near the grid. Capacitors that are very close to our finger will have large changes in values (red in Fig. 2.11). Capacitors slightly further away will change less (blue). Capacitors still further from the finger will not be affected at all, or at least not much compared to those near the finger (gray). By measuring the changes in capacitances, and using a little interpolation, we can determine which capacitors the finger is closest to.

Our next task is to turn that into a useful coordinate. Let's assume that our rows and columns of capacitors are 6 mm apart (i.e., 6 mm from the center of one capacitor to the center of the next capacitor). We could define our active area, that is, the area in which we can measure a change in capacitance, as aligning with the centers of the outer capacitors (Fig. 2.12).

However, we know that if a finger is outside of the last row or column, it will still have an effect on the outer capacitors it is closest to. To make things simple, let's assume we can use half the pitch around the edges, meaning that we can detect a finger up to 3 mm away from the center of our outer capacitors.

So our new active area is now defined like Fig. 2.13.

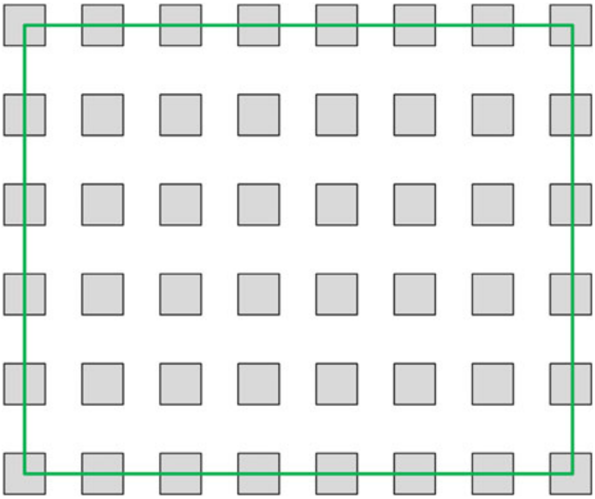


Fig. 2.12 Active area centered on outer capacitors

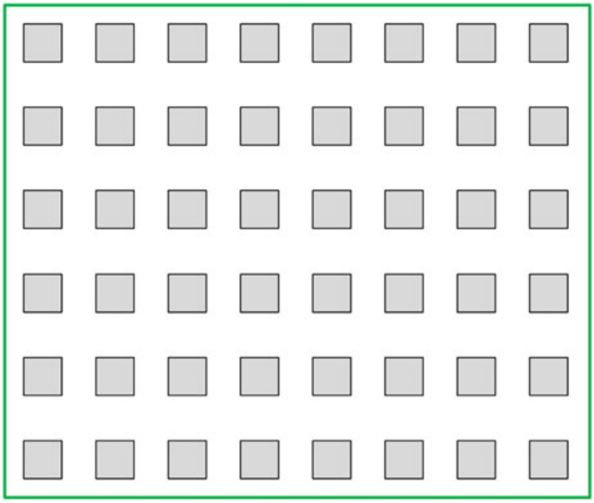


Fig. 2.13 Active area outside outer capacitors

We need to apply a coordinate system to this area. We will follow the LCD coordinate convention since eventually we will want to compare our touch location with the location of some widget on the screen. The upper left corner is (0, 0). To keep things simple for the software engineers who prefer powers of 2, we will define the bottom right corner as (1023, 1023). To clarify later discussions involving multiple coordinate systems, we will call these coordinates touch pixels.

We have 8 rows and 6 columns, each with a pitch of 6 mm for a total active area of 48 mm by 36 mm. Those lengths are mapped into our 1024 $\times$ 1024 coordinate system (Fig. 2.14).