Hello and welcome to course 411C related to Capacitive Touch User Interface Solutions. My name is Nelson Quintana.
Mr. Nelson Quintana

- Senior Manager, Product Marketing
  - Business Development for White Goods and Small Appliance Market Segments heavily focused on User Interface Applications
  - Renesas R8C/3xT Capacitive Touch MCU Roadmap Definition
  - Responsible for Renesas 8-/16-bit MCU Strategy in the Americas

PREVIOUS EXPERIENCE:
- Over nine years at NEC Electronics America as a Field and Technical Application Engineer for 8-bit and 16-bit Microcontroller products and Development tools

- BSEE in Digital Design with concentration in Microprocessor/Microcontroller systems from San Francisco State University

Just a little background about me –
I have been with Renesas for over four years. My primary responsibilities include business development for home appliance market segment, and product marketing for Renesas 8- and /16-bit MCU products in the Americas. I’ve been personally involved in the definition of Renesas R8C/3xT MCU Touch solution which are tailored for a number of applications including appliances.
Before Renesas, I spent over 9 years at NEC Electronics America as an Applications Engineer for 8- and 16-bit MCUs.
In the session 110C, Renesas Next Generation Microcontroller and Microprocessor Technology Roadmap, Ritesh Tyagi introduces this high level image of where the Renesas Products fit. The big picture.
This is where our session, 411C, 'Capacitive Touch User Interfaces and Hardware-based Solutions' is focused within the 'Big picture of Renesas Products
Here are the MCU and MPU Product Lines, I am not going to cover any specific information on these families, but rather I want to show you where this session is focused.
The Renesas capacitive touch solution is based on the R8C MCU Family.
Technology innovation brought by devices such as the iPhone are literally transforming the expectation from consumers for richer and better human-machine interfaces. The use of capacitive touch detection technology is rapidly expanding to just about any device containing some type of input interface required to operate it. Examples shown here include home appliances, portable consumer electronics, office equipment and medical equipment.
Touch controllers that fit your products best

Renesas’ hardware-assisted touch sensing MCU is ‘ahead of its class’ for providing full system control and overall lower power consumption without sacrificing touch performance.

Capacitive-based touch technology is not new, but Renesas provides you with an innovative solution enabling better control.

Renesas hardware-assisted touch sensing MCU is ahead of its class for providing full system control and overall lower power consumption without sacrificing touch performance.
This course will cover:

- Key Reasons to Use Capacitive Touch-based User Interfaces
- Typical uses of capacitive touch interfaces
- Capacitive touch detection methods and general challenges
- MCU-based solutions available in the market today
- Overview of Renesas Touch solution including:
  - Key features
  - Benefits of hardware-assisted touch
  - Development Tools
  - Target applications, and Roadmap
- Q&A
Key Takeaways

By the end of this session you will be able to:

- Know about market trends in HMI technologies
- Understand types of capacitive touch solutions available
- Identify pros and cons of hardware and software-based solutions
- Recognize the benefits of Renesas’ touch solution

By the end of this session you will be able to:

- Know about market trends in HMI technologies
- Understand types of capacitive touch solutions available
- Identify pros and cons of hardware- and software-based solutions
- Recognize the benefits of Renesas’ touch solution
Let’s start by exploring reasons why capacitive-based touch technologies are being adopted in various applications.

The first reason is, Better Aesthetics
Here is an example of a real product (IP Phone) which has been re-designed with keypad most likely using capacitive touch I/F. The new design looks more Innovative and “modern” compared to traditional button I/F. In addition, the layout of the keys and related functions can be accommodated easier to make a more intuitive interface. (These are probably the same reasons why products such as the iPhone have been very popular).

The second, yet very important reason to use cap touch is for Reliability purposes.
For example: The buttons and knobs typically used to control a home appliance contribute to the unit’s overall reliability. Due to their electro-mechanical nature, these controls are higher-prone to wear and tear, which can cause failures. In addition, the design can be more complex, and the manufacturer also needs to take into account the added logistics to get the components. The same appliance re-designed with capacitive touch interface, doesn’t just look better, but it eliminates the need for the electro-mechanical components, making the design easier and more reliable.
Another reason why capacitive touch is being adopted is related to day-to-day maintenance. I am showing here an example of an infusion pump device. This type of equipment is usually exposed to environments which can be contaminated by chemicals or blood. Imagine having how much works it would take to maintain this device clean. The same device designed with a touch interface offers an easier-to-clean surface.

The last key reason for using touch is in fact related to the overall benefit of cost reduction. Engineers can potentially use one common board for a platform of products having different user interfaces. Because of the physical design of a touch board, variations of interfaces can be easily accomplished by just changing the overlay panel. Here is an example of one of the Renesas touch boards with two different overlay designs. One overlay gives access to all keys and display while the other one masks or hides some of this interface components through the graphics on the panel.
So, what are some of the most common capacitive-based touch interfaces we find being used today. Discrete keys is probably the most popular form of touch interface. Designers typically make use of a key implementation to replace single buttons or a full array of mechanical switches. Shown here is a photo of a notebook with touch keys for volume and LCD backlight functions.

Another very popular interface is what is many times referred to as a touch Wheel. iPods were probably the first products on which this type of interface was widely used. There could be different shapes and forms for wheels, but in most cases, designers use them to replace a mechanical knob (such as those use to adjust volume settings on an audio equipment).

Sliders can be used in similar fashion as wheels, but they are linear in shape and are mostly used when an adjustable setting has upper and lower limits.

Depending on the application requirements, designers can easily define the actual operation performed by wheels and sliders, so the same interface can have completely different behavior from one product to another.

The last common use for capacitive-based touch interfaces, which I would like to mention briefly is in LCD screens. Cell phones, tablet PCs, ATM machines are examples of products which can make use of this type of interface. The touch solution currently provided by Renesas is geared mainly for keys, wheel and slider implementations, therefore the touch screen application is not covered in this material.
Let’s now look at methods to detect a touch. Actually, there are two main principles used by most solutions available today. The first method is commonly known as mutual capacitance-based detection, or some times referred to as the “driven method”.

In Mutual capacitance, a sensor (or a touch element) is made up of one Transmission Node and One Reception Node. When an electric current is applied to transmission node, an electromagnetic field is generated and detected on the reception node. When a person’s hand gets close to the sensor area, some of the field gets absorbed thereby decreasing the amount of energy detected by the reception node. Due to its pulse-generating structure this method typically has a significant influence on overall power consumption and noise emission. In addition, this method may allow for shielding which can provide good noise tolerance.

The second method referred to as Self-capacitance requires only one electrode which holds a “floating capacitance”. The floating capacitance is influenced by parasitic capacitance between the electrode and surrounding electric conductors. Since the human body is a conductor, when a finger is placed close to the electrode, the value of the floating capacitance increases and can thereby be detected through a measurement terminal. This method is typically simpler to implement, requires no extra power and there is no noise emission associated with it. Although, in the Self-capacitance method shielding is generally not allowed, noise immunity depends on the actual implementation and overall circuit design practices.
An now, let’s have a question. I have some giveaways for the correct answer.

Q: What are typical challenges with touch design?

This leads into the next foil, which is...
Design Considerations for a capacitive-based touch system:

All touch designs require specific PCB layout guidelines. This include many aspects such as electrode size, separation from sensor to sensor, trace length and width, and even sensor shape.

Most designs make use of a surface overlay panel, which can be made of different types of material such as glass and plastic. The panel’s composition can affect various aspects of the touch system, including signal-to-noise ratio, durability, ESD resistance, as well as accuracy. The panel’s own capacitance characteristics influenced by the material’s dielectric constant and overlay thickness will play an important role in touch sensitivity.

During development, a system needs to be tuned and calibrated to meet the specific application requirements such as sensitivity and response time. Even when using the exact same solution in two different boards, this process is required to optimized touch performance.

Environmental influences such as temperature and humidity, and even contamination can affect the capacitance characteristics of a system. To ensure consistent and reliable touch detection, the system must be capable to compensate for these drifting effects, which can many times handled in software.

To prevent false touches and guarantee appropriate touch behavior, designers must take into account and test for electrical noise generated from within the system, or from outside sources. Engineers typically incorporate filtering techniques in software and/or hardware to handle issues which are many times specific to each system/design.

The last consideration I would like to discuss is patents. As a designer you should be aware that there are many patented technologies and solutions in this field, including detection mechanisms, hardware implementations and software techniques. The good news is that Renesas solution is based on a patented detection technology originally developed by OMRON. Renesas has licensed the core technology and incorporated it into MCU products.
### MCU-based Capacitive Touch Solutions

<table>
<thead>
<tr>
<th>MCU Suppliers</th>
<th>Technology “Names”</th>
<th>Main Detection Method</th>
<th>Hardware-Assisted Detection</th>
<th>Types of Solutions Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Renesas</td>
<td>Series Capacitance Voltage Division</td>
<td>Self-capacitance</td>
<td>Yes</td>
<td>Touch MCU</td>
</tr>
<tr>
<td>Atmel</td>
<td>Charge Transfer Technology (QTouch and QMatrix)</td>
<td>Self- and Mutual-capacitance</td>
<td>Yes (MaxTouch)</td>
<td>Touch MCU; Sw Library</td>
</tr>
<tr>
<td>Cypress</td>
<td>1)CapSense Successive Approximation (CSA) 2)CapSense Sigma-Delta (CSD)</td>
<td>Mutual-capacitance</td>
<td>No</td>
<td>Touch MCU</td>
</tr>
<tr>
<td>Freescale</td>
<td>1)Advanced Touch Logic (ATL) 2)Capacitive Turbo Sensing (CTS)</td>
<td>Self-capacitance</td>
<td>No</td>
<td>Sw Library</td>
</tr>
<tr>
<td>Microchip</td>
<td>Relaxation Oscillator</td>
<td>Self-capacitance</td>
<td>No</td>
<td>Touch MCU; Sw Library</td>
</tr>
<tr>
<td>ST</td>
<td>RC Acquisition</td>
<td>Self-capacitance</td>
<td>No</td>
<td>Sw Library</td>
</tr>
</tbody>
</table>

1 Information based on information available through public websites as of Sep 2010
2 Comparable to Renesas MCU-based solution

There are many capacitive-based touch solutions available. Let me give you quick comparison of some of the MCU-based solutions.

This table shows six MCU supplier including Renesas and their respective touch solutions available (as of Sep 2010). Suppliers use different names for their technologies and they may offer them in different forms. I’d like highlight only two sections on this table – The first one is detection methods. As you can see, most solutions including Renesas are based on self-capacitance method, which as I mentioned earlier provides great benefits such as lower power consumption. The second are to focus here is the hardware-assisted detection column. Renesas is the only supplier today that offers an MCU-based solution with specialized hardware to assist in touch detection. The only other supplier that offers similar technology is Atmel, but the solution is a mostly a sensor-only implementation geared for touch screens.
Let’s jump right into the key benefits of the Renesas capacitive touch solution. I’ll start with a summarized list and then cover each area in more detail.

Starting with #1 – Renesas offers a single-chip MCU-based solution. This means that designers can use one chip to perform capacitive touch and system control.

The 2nd benefit is hardware-assisted touch detection featured I just mentioned. The Renesas MCU incorporates an innovative hardware block called Sensor Control Unit to perform touch detection functions resulting in minimum CPU overhead.

The touch detection mechanism has also been architected to minimize overall power consumption in addition to the standard low power modes available in the MCU.

Benefit #4 is easy to remember – the solution is simply designed to achieve low system cost.

Finally, Renesas realizes that it is not just about the chip. A full infrastructure consisting of advanced development environment, optimized firmware and essential design support such as application notes.
Renesas touch solution is based on the R8C MCU family, specifically on the R8C/3xT Series ("T" meaning touch). The significance of this is that the touch MCU is fully compatible to many other general purpose R8C devices which have been used in many UI applications over the years. Designers can easily move from a non-touch R8C to the R8C/3xT series with minimum effort.

Let’s look at a cook range as an example – the design may require display control, a buzzer, communication to another device and perhaps even temperature monitoring. The 3xT can easily perform all these function using onboard peripherals as shown here. For the touch portion, the 3xT incorporates a dedicated hardware block called Sensor Control Unit (or SCU) which can be connected to discrete keys as required by this UI panel. The same block can interface to a touch sliders, or a wheel configuration to replace the mechanical knobs shown in this example. This mixture of hardware blocks makes the R8C/3xT very attractive because it can perform the functions traditionally done with two ICs.
Let's discuss briefly how the 3xT actually detects a touch using the Sensor Control Unit (or SCU).

Here is a simplified connection inside the SCU. CHA, B and C in the diagram are referred to as touch system channels. These channels are connected to a pair of capacitors and a resistor which are used for charging/discharging and reference. CH0 is one of the touch sensor inputs. As discussed earlier the 3xT solution uses the self-capacitance method to detect touch. Cx in the diagram is the parasitic capacitance associated with the electrode connected to channel 0. So, basically when a touch occurs, Cx will increase. However, the actual touch is detected by an analog comparator circuit. As shown by this formula, Voltage VCHxA which is fed into the comparator is inversely proportional to the Cx.

To measure changes in capacitance accurately, the following steps are performed.

1. Charge Cc by connecting the CHxB pin to the voltage supply.
2. Discharge Cc by connecting CHxΩ to common.
3. After discharging Cc for a short period of time, the SCU measures voltage at VCHxA
4. This voltage is compared to a internal voltage threshold Vth. If VCHxA is higher than Vth, the discharge and measurement process is repeated
5. Once the threshold is hit, the comparator will trigger a timer to capture a running counter

This discharge count will be different if an electrode is touched. This difference in counts is used to determine if there is a valid touch or not.
Let’s zoom out on the SCU to understand what functions are provided by it.

The SCU contains blocks to handle full configuration of options such as clock source, touch triggering mechanism, enabling of channels and so on. Error checking or flags are also available to manage the operation of the SCU.

The 2nd main function provided by the SCU is all the actual measurement mechanism along with all the timing, sequencing and even channel scanning selection.

There are blocks within the SCU which can be used for noise countermeasures, such as the secondary counter.

Finally, once the measurement data is collected, the SCU can use a dedicated sensor DMA engine to data transfer data to SRAM for post-processing. Alternatively, the data can also be transferred using the Data Transfer Controller unit built into the R8C devices.

-----------------------------------------------

DMA used in Standby
DTC used in Active Mode
As an example of one of the SCU features, let’s look at the automated scanning function.

The SCU can be configured to start a touch scan by using a register bit, a timer or an external signal as a triggering mechanism.

Each touch channel can be enable or disable independently.

The scanning sequence for the channels can also be controlled automatically by hardware, so that depending on the application requirements, the code can scan in ascending or descending order.

At the end of the touch scanning process, the SCU can also generate an interrupt for efficient processing.
All these features in the SCU make it possible for the 3xT to perform touch sensing with minimum CPU bandwidth.

Regardless of how the touch detection is implemented, when a solution is software-based, the CPU must spend a significant amount of time performing all the related operations. In an application where the MCU also needs to carry out system functions, the CPU’s time must be distributed to achieve both tasks – here is an illustration of such a scenario. After touch measurements are performed most solutions will require post-processing to determine a valid touch, especially if additional filtering is needed.

In case of the R8C/3xT, the user has two options. The first one, as shown, disables the CPU completely while the Sensor Control Unit performs all the touch related functions. The CPU is activated only for the post-processing portion.

Alternatively, if the system requires the MCU to perform system functions such as updating a display, or communicating to a host processor, the R8C/3xT can take advantage of the SCU to perform all touch functions while the CPU is 100% available for application-related operations. In this scenario, the time required for the CPU to the touch post-processing does not exceed 15% of the total time.

---

Scan Time: 4ms  
Decoding Time: 500us (@ 20MHz)  
Total Time: 4.5ms  
% of CPU used = 500us/4.5ms = ~11%
How does this co-processing capability offered by the SCU helps to reduce power consumption.

Let’s look at a digital photo frame as an example. This particular one is battery operated and has multiple touch keys including a standby function to enable/disable a sleep mode to save power.

To minimize current draw, the process can be put in standby mode (or Wait mode) for a period of time. A timer interrupt can then turn the SCU block on to perform a touch scan and then post processing is performed with the CPU to determine if a key has been touched. The application can cycle through this sequence until a key pressed is detected.

The R8C/3xT’s power consumption for each of these phases is shown in the table.

If we use approximately 100ms for each cycle, which would be acceptable time to respond to user input, the average power consumption would be under 16uA to detect a touch on the standby key.

If more channels are to be scanned only touch detection time would be slightly increased, so the impact on the average current would be minimum.

Renesas has an app note and sample code that shows this in detail.

Matrix: Decoding 32 keys or less will typically take less than 150 uS
Let’s now discuss how the R8C/3xT can help you reduce total system cost.

Let’s start with the basics. Here is a diagram showing a typical MCU with standard blocks. Depending on the application requirements, there is a number of external IC’s that may be required such as DC-DC converter and a crystal resonator.

The R8C/3xT eliminates this need because of all the on-board components and integrated functions. A wide operating voltage range makes it easier to handle power supply requirements. An internal 40MHz oscillator provides accurate timing even for touch sensing.

In terms of touch, as discussed earlier, only two capacitors and one resistor regardless of the number of keys in the system.

Just as in most touch systems, a limiting current resistor connected to each touch input is recommended for ESD protection.

When keys are arranged in a matrix configuration such as the one shown (a 3x3), only one pin per row and one per column is required. A maximum of 4 keys can be connected to each line. This configuration allows for maximum use of I/O which can be used for other functions in the system eliminating I/O expander ICs.

Finally, as with most R8C devices, the R8C/3xT can drive up to 20mA through all the output ports available. This feature eliminates external drivers used for LED’s.

---

Cr: 5 to 30 pF typical  
Rc: 1 kΩ to 30 kΩ typical  
Cc: 0.1 uF  
Matrix: Decoding 32 keys or less will typically take less than 150 μS
Let’s change gears a bit and look at the software architecture that Renesas has put in place to support touch applications.

At the bottom of the software stack, we have the hardware interface layer. This is all the software associated with the configuration of the SCU block (i.e.: the drivers).

Above the drivers, we have the Sensor Layer. This is basically where the processing and touch decision is made. The touch or no-touch logical output from this layer can then be used by what we call the Functional Implementation Layer.

In this upper layer, is where the touch output can be further interpreted as a valid touch on a wheel or slider configuration depending on how it is implemented on a specific application.

The User Application can then determine how to use the data to perform UI functions.

The bottom two layers make up what we call the Renesas Touch API. We have optimized this to minimize code to a bare minimum, but most importantly it gives engineers a very quick way to integrate into a real application without the need to worry about the SCU specifics. In addition, The Touch API is provided in source code providing full transparency for user customization if required. Renesas also provide a set of examples for the Functional Implementation blocks.
Before I move to other aspects of the Renesas Solution, let me briefly summarized a comparison between software-based and a hardware-based touch solution based on the R8C/3xT information I have presented.

• In a software solution the MCU must spend a significant amount of time to perform all the touch sensing operations such as measurement, and scanning. (Just think about the all port settings and charging/discharging cycles which are required for many of the touch detection mechanism).

• During this control sequence the MCU is not available to do other tasks

• In addition, to ensure the touch detection/processing is properly performed, processing of other interrupts is limited, which may affect the ability to respond quickly to other system functions, or a reliable touch detection can be compromised.

In contrast, a Hardware-based solution such as the one offered by the R8C/3xT, the critical touch operations are fully automated by the Sensor control unit. This allows for the CPU to be fully available for system control functions. In addition, because processing is performed by a dedicated and autonomous hardware block, the system can consume much lower power compared to system where the CPU or an A/D converter is required to be operating. Finally, less CPU requirement also means less code being executed – that is one of the reasons why the full Renesas touch API is less than 2.5KB of code.

So, Why would you want to do this?..., when you can have this?...
What about development tools?

A number of Renesas and 3rd party tools are available for development of General purpose R8C MCUs. Most of these also apply to the R8C/3xT Touch MCUs. In addition however, Renesas offers a tool called Touch Workbench which is designed to help engineers quickly evaluate and tune hardware and software for optimum touch performance.

The tools is simple to operate, yet very powerful and intuitive. Within Workbench, there are various utilities available. Parameters used by the Touch API, such as reference values, thresholds and drift compensation can be modified on the fly through the Parameter Setting utility to tune each channel in real time. The measurement at each touch channel can be monitored graphically via the Measurement tool. Finally, Workbench also includes a Circuit Modeling utility which provides recommended values for the capacitor and resistor components based on physical parameters of the electrode and overlay characteristics entered by the user.

Specifics about this tool will be covered in the lab.
So, how can you get your hands on the Renesas Touch Solution?

We have developed a complete evaluation system based on the R8C/33T, which is a 32-pin device. The kit includes a touch board with keys, wheel and slider interfaces. The MCU drives a display and other peripheral components.

Users can measure the MCU power consumption.

The kit comes with an E8a debug emulator which is used for debugging and programming. A CD contains the Touch API, sample code, the Workbench tools, a 64KB Free R8C C compiler and all the documentation including application notes to help users with touch design.

The kit’s part number is shown above and will soon be available for order through our website.
Question

What are the top benefits of Renesas Touch Solution?

Answer:

• Hardware assisted detection for low CPU bandwidth: Sensor Control Unit
• Highly Integrated: Single-chip solution for touch and system control
• Low Power: ~20uA average while touch detection
• Low cost features: Minimum external components + optimized used of pins, Data Flash, POR, internal oscillator
• Complete Support: Full API (source code), Self-contained evaluation system, application notes

Here is another question:

What are the top benefits of Renesas Touch Solution?

Answer:

• Hardware assisted detection for low CPU bandwidth: Sensor Control Unit
• Highly Integrated: Single-chip solution for touch and system control
• Low Power: ~20uA average while touch detection
• Low cost features: Minimum external components + optimized used of pins, Data Flash, POR, internal oscillator
• Complete Support: Full API (source code), Self-contained evaluation system, application notes
Let’s look at the Touch MCU roadmap based on the R8C/3xT Series. The first device already available in production is the R8C/33T which comes in a 32-pin package and provides 18 SCU channels with up to 32KB. The R8C/3JT, also in production, offers 4 additional SCU channels and comes in a 5mm sq. QFN package making it ideal for designs with limited physical space.

The R8C/3xT Series is being expanded to include a 64-pin device with 28 SCU channels and up to 128KB of Flash memory. A 36-SCU channel version is also being developed in a 80-pin package to offer additional I/O. Renesas is also considering to develop a low-end R8C/3xT device in a 20-pin package targeted for simple touch applications.
The Renesas solution is geared for any application which requires a touch key, wheel or slider interface. Some examples of those include exercise equipment, kitchen appliances, office equipment, portable devices, audio equipment and security keypad amongst others.
Q&A
Now that you have seen the benefits of the Renesas Touch solution, I am sure you will find ways to take advantage of it in future designs.

Who knows, perhaps this innovative solution will encourage you to explore new and exciting frontiers....
Thank you all for joining us today.
Appendix
Main Points:
Full on-chip debug capabilities of R8C/3xT available

All R8C MUCs incorporate dedicated hardware for debugging. This debug unit provides features such as hardware breakpoints for execution and data match as well as tracing capability. High-performance versions also include a DMA engine built-in the debug unit which allows real-time memory access during debug operation.

In-system debugging can be accomplished as easy as connecting one of Renesas's on-chip debug emulators such as E8a to the R8C dedicated MODE pin. Additional connections to power supply and reset signal may be required depending on system configuration. This same interface can also be used for flash memory programming which makes it very simple and convenient for firmware upgrades.