NXP Semiconductors
Application Note

Document Number: AN12190
Rev. 1, 03/2019

QN908x Capacitive Sensing Design Guide

Contents

1 Introduction

This document details the Capacitive Sensing (CS) interface of QN908x. It includes QN908x key CS features introduction, sensing basics, supported patterns, hardware design consideration, SDK porting, and key parameters tuning.

2 Key Features

- Self-capacitance sensing.
- Active mode operation capable of up to eight input channels with integrated hardware scan.
- Configurable oscillation frequency to calibrate drive current.
- Interrupts available for when a channel scan for each enabled channel is completed, FIFO is not empty, half full, or full.
- Dedicated LP mode that operates on the 32 kHz clock. When using the LP mode, a single channel can still be scanned while in power-down mode. An interrupt can be generated to wake up the part when the channel scan returns values that meets the programmable threshold and the de-bounce value.
- Support NXP touch library and debugging tool.

© 2018 NXP B.V.

3 Capacitive Sensing Methods

Capacitive sense interface (CS) provides touch sensing detection with the capacitive touch sensors. The capacitive touch sensor is a copper pad on the PCB and the sensor electrodes are connected to CS input channels through the I/O pins of QN908x.

QN908x CS supports the self-capacitive mode only, as showing in the Figure 1 below.

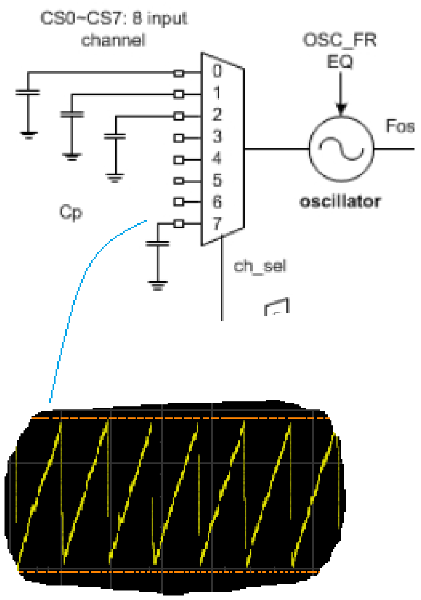


Figure 1. Self-capacitive mode

3.1 Capacitive Sensing Methodology

The capacitive sense determines proximity and touch detection based on the self-capacitance. When a finger contacts a touch pad, the change in capacitance is measured between the input channel and the ground. This is accomplished by utilizing a relaxation oscillator that increments a counter every cycle. When there is no contact on the touch pad, the relaxation oscillator will oscillate at a user-configurable frequency based on the parasitic capacitance of the system. When a finger makes contact, the total capacitance increases, causing the frequency of the oscillator to drop. It causes a noticeably low counter output when there is no contact with the touch pad. Touch events are determined by observing the change in the counter output.

Counting the frequency of the relaxation oscillator is the method the capacitive sense utilizes to detect a touch event. This frequency changes depending on the capacitance of the input channel and the amount of current being driven on the pin. The estimated value of the parasitic capacitance is illustrated in figure 2, by which we can decide the frequency of the relaxation oscillator.

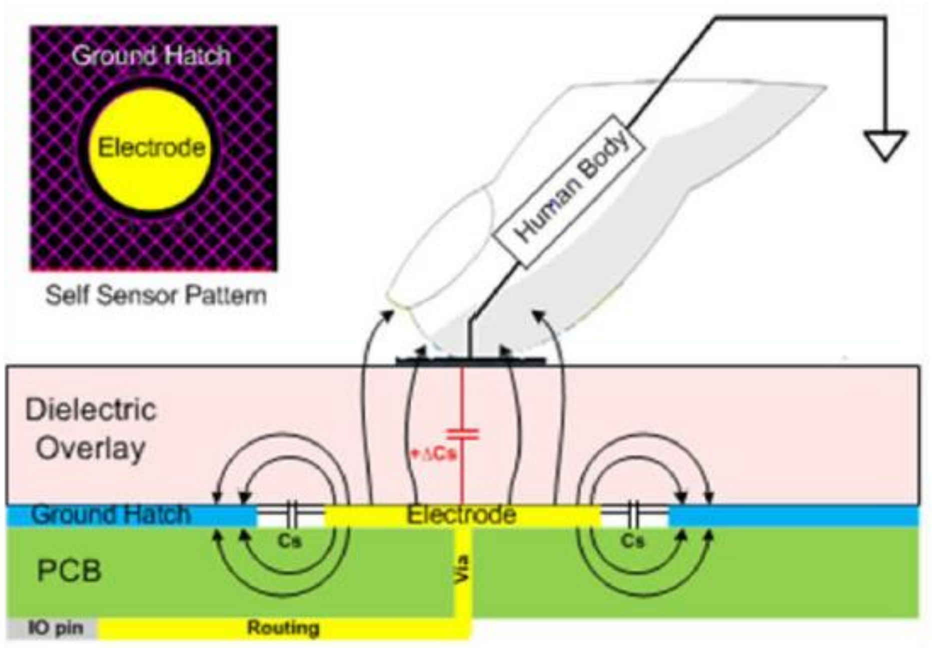


Figure 2. Self-capacitive equivalent circuit

The figure illustrates the touch-sensor structure of the self-capacitive mode:

- Cs — Intrinsic self-capacitance. 10 pF ~ 50 pF
- ΔCs — Touch generated self-capacitance. 0.3 pF ~ 2 pF
- Sensitivity of sensor — ΔCs/Cs. 1% ~ 10%

3.2 Parameter Settings

The drive current and frequency are a function of the programmable OSC_FREQ bit field in the CTRL0 register. The actual total amount of current being driven to the input channel is approximately

$$0.4 + 0.2 \times (\text{OSC\_FREQ} + 1)\ \text{uA},$$

and the current being driven to the touch pads is

$$0.2 \times (\text{OSC\_FREQ} + 1)\ \text{uA}.$$

The frequency is approximately

$$2 \times (\text{OSC\_FREQ} + 1) / RC\ \text{Hz},$$

where the resistance is about 5 M.

For example, if OSC_FREQ = 0 and the input channel capacitance is 10 pF, the current being driven to the pin is 0.2 uA with a relaxation oscillator frequency of approximately 40 kHz. Though the calculation of the frequency is a rough estimation and will drift in frequency, a touch event should be distinguishable by detecting a significant change of signal in the counter output.

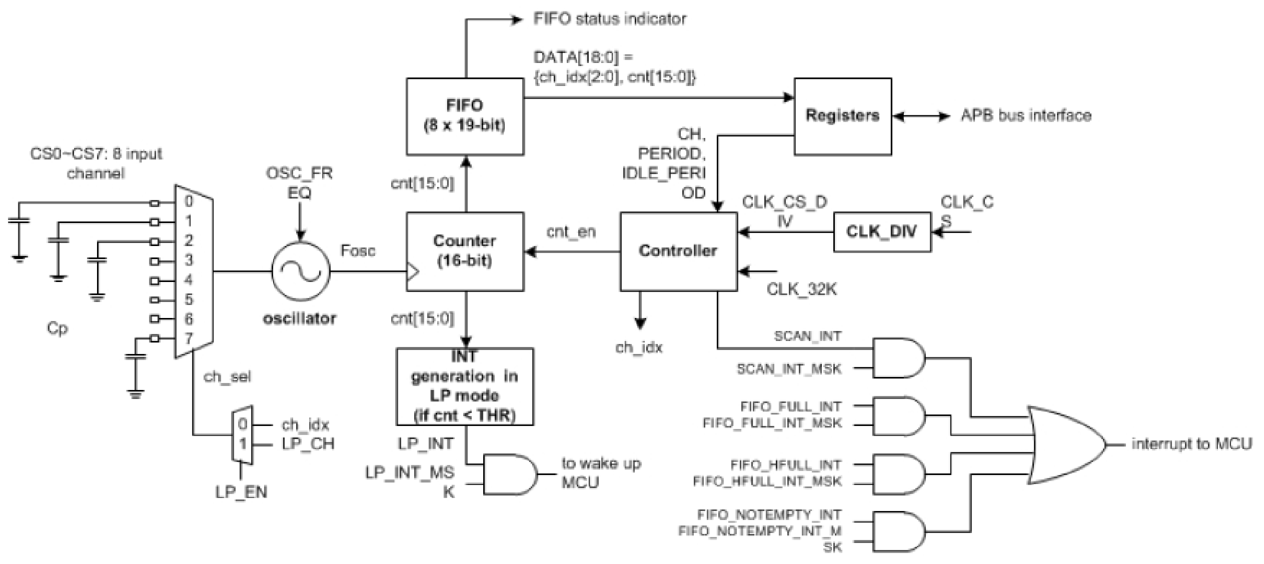


Figure 3. Capacitive Sense block diagram

The capacitive sense supports up to eight input channels. The capacitive sense scans the enabled channels starting from the lowest index to the highest.

During a scan, a counter increments every clock cycle of the relaxation oscillator. The hardware scans based on the capacitive sense clock source, CLK_CS. In active mode, the CLK_APB is divided by CLK_DIV (in CTRL0) plus one to produce the clock for the capacitive sense, CLK_CS.

$$CLK\_CS = CLK\_APB / (CLK\_DIV + 1)$$

The programmable PERIOD bit field in the CTRL1 register contains a value that indicates how many CLK_CS clock cycles to scan for. Between scans, it is possible to set a specific amount of idle time to save power. The IDLE_PERIOD register indicates how many CLK_CS clock cycles to wait before starting another scan.

The following figure shows the timing of CS measurement for the self-capacitive mode.

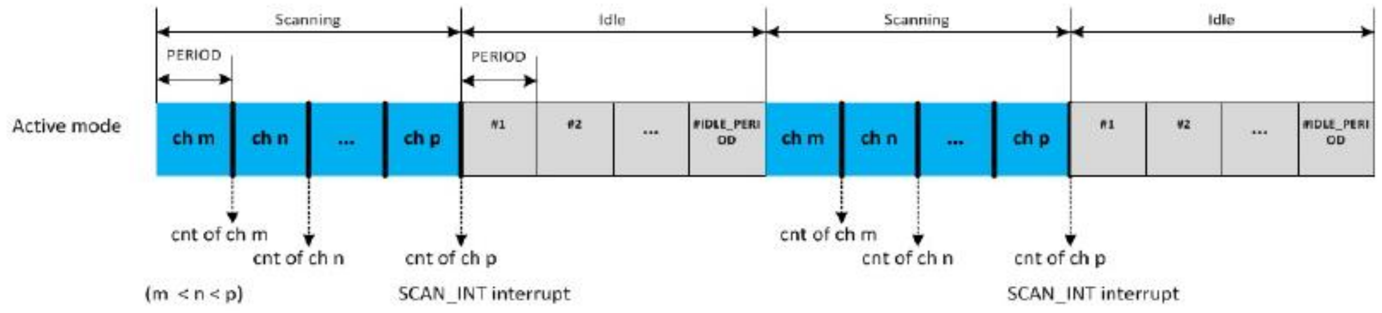


Figure 4. Operation modes

The scan window for each channel is PERIOD / CLK_CS.

The counter value of each channel is approximately equal to the scan window times the oscillator frequency. The basic formula of the self-capacitive mode is listed below, COUNTER is the result of CS single scan:

$$\text{COUNTER} = 2 \times (\text{OSC\_FREQ} + 1) / RC \times \text{PERIOD} / \text{CLK\_APB} \times (\text{CLK\_DIV} + 1)$$

- OSC_FREQ — Configurable, 0x00 ~ 0x3F
- R — Rresistance is about 5 M
- C — Self-capacitance, Cs (release) or Cs + ΔCs (touch)
- PERIOD — How many CLK_CS clock cycles to scan a channel for, 0x00 ~ 0xFFFF
- CLK_APB — APB clock，16M or 32M
- CLK_DIV — Configurable, 0x00 ~ 0x1FF

4 Supported Patterns

The capacitive sense interface supports pattern, keypad, analog rotary, and analog slider.

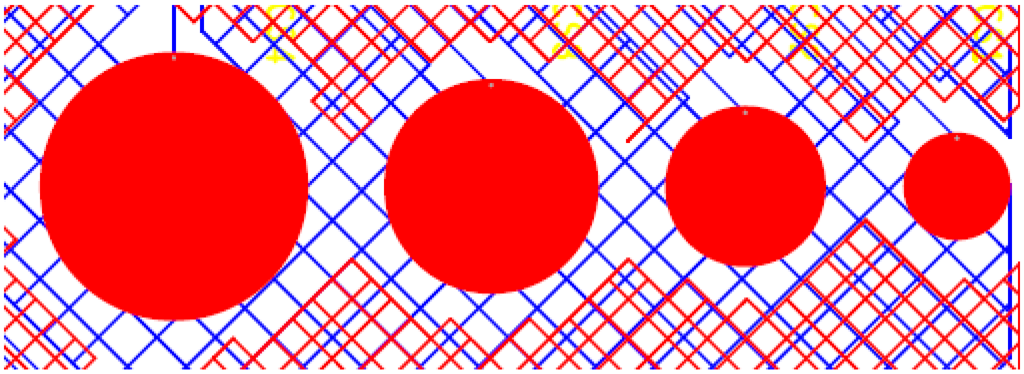


Figure 5. Single keypad

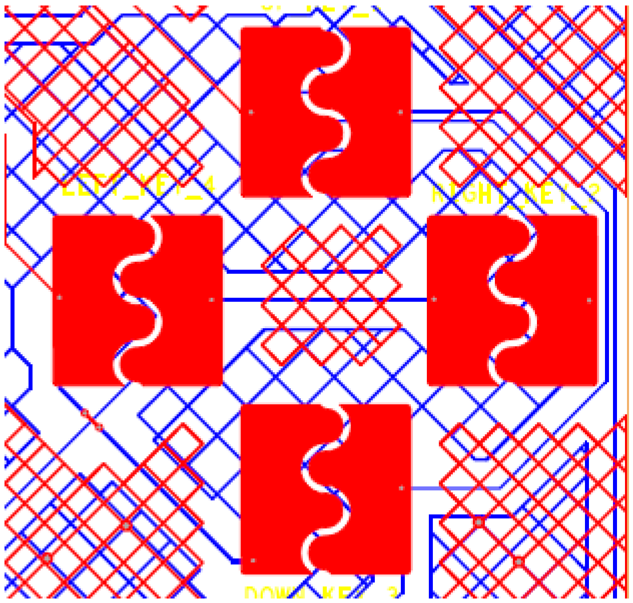


Figure 6. Matrix keypad

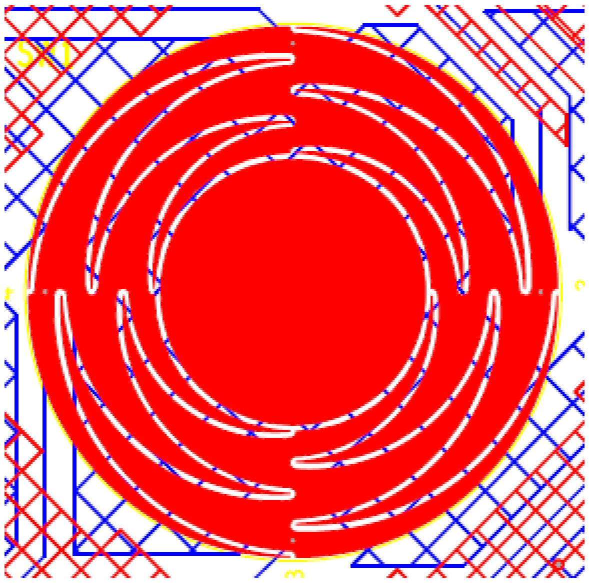


Figure 7. Analog rotary

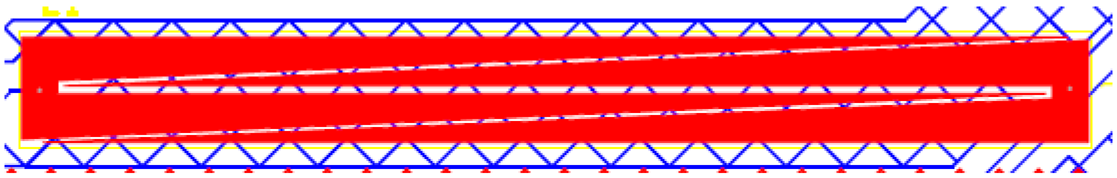


Figure 8. Analog slider

5 Hardware Setup

The hardware setup in this example uses a QN9080 DK and a touch-sense board, shown in the Figure 9 and Figure 10.

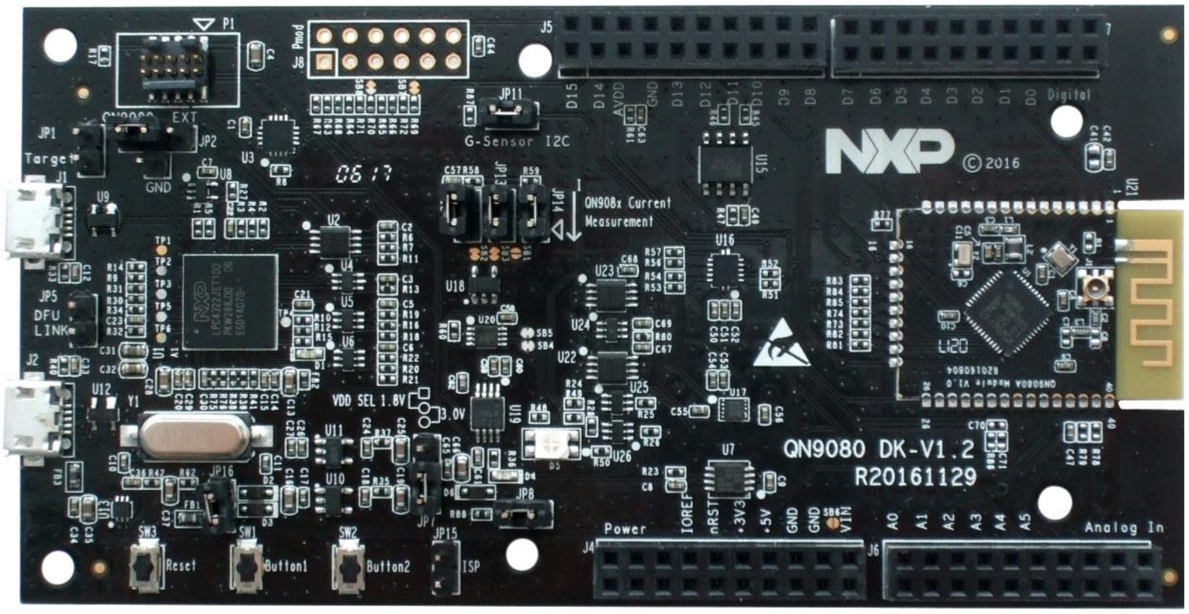


Figure 9. QN9080 DK

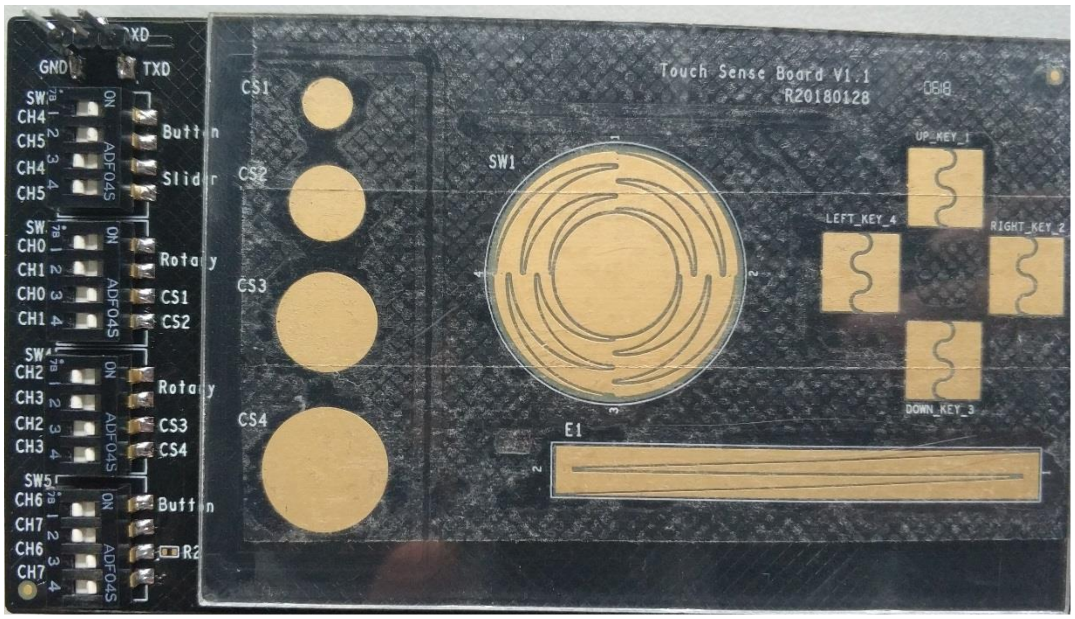


Figure 10. Touch Sense Board

5.1 Board Configuration

The capacitive sense is a sensitive to the change of capacitance, make sure the following setting is correct.

Table 1. Jumper settings

Name	Configuration	Info
JP11	not populated	disconnect the PA14(CS0) from other circuits.
JP8	not populated	disconnect the PA16(CS2) and PA17(CS3) from other circuits.
R25	remove	disconnect the PA25(CS7) from other circuits.
R81	remove	disconnect the PA24(CS6) from other circuits.
R82	remove	disconnect the PA25(CS7) from other circuits.

The capacitive sense channel CS2 and CS3 are multiplexed with the UART pins PA16 and PA17. The PA4, and PA5 pins are used for debugging UART. Connect these two pins to a USB-UART converter via J8 of the touch sense board as shown below.

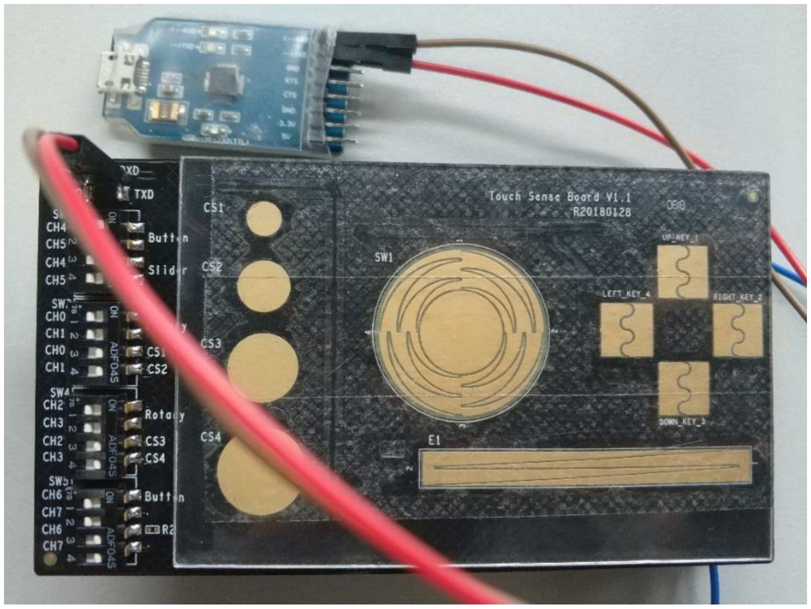


Figure 11. Serial debugging pins

5.2 Toggle switch channel configuration

Table 2. On-board switch setting

Channel	Switch	Pattern
PA14_CS0	SW3 - 1	Arotary-1
	SW3 - 3	Single keypad-1
PA15_CS1	SW3 - 2	Arotary-2
	SW3 - 4	Single keypad-2
PA16_CS2	SW4 - 1	Arotary-3
	SW4 - 3	Single keypad-3
PA17_CS3	SW4 - 2	Arotary-4
	SW4 - 4	Single keypad-4
PA18_CS4	SW2 - 1	Aslider-1
	SW2 - 3	Matrix keypad-1
PA19_CS5	SW2 - 2	Aslider-2

	SW2 - 4	Matrix keypad-2
PA24_CS6	SW5 - 1	Matrix keypad-3
	SW5 - 3	Reserved
PA25_CS7	SW5 - 2	Matrix keypad-4
	SW5 - 4	Reserved

6 Hardware Design Consideration

6.1 Electrode shapes

To maximize the electrodes area from the capacitor plates, it is recommended that the size of the electrode is comparable to a human finger (10×10mm is a good size).

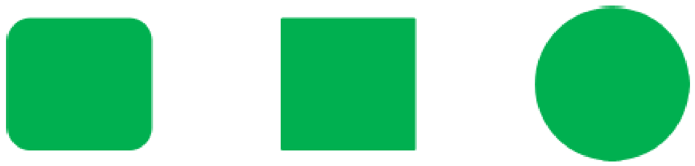


Figure 12. Electrode shapes

6.2 PCB trace routing

The following are the recommendations for correctly routing the traces of capacitive electrodes.

- Width — Keep traces width as thin as possible. 5-7 mil trace is recommended. A 5mil trace has half the capacitive coupling with the planes compared to a 10mil trace.
- Length — As short as possible. Trace length must be less than 300mm, minimize trace length from TSI pins to touch pads in order to optimize signal strength.
- Clearance — To ensure signal integrity, leave a minimum clearance of 10 mils for the lines that run parallel to each other in the same layer, and route perpendicularly the ones running in adjacent layers. Good design practice is to keep traces separated as much as the design allows. At the sensor's end, where typically the pitch is lower than 10 mils, a bottleneck mode connection is recommended as shown in Figure 13. The figure below is an example for maintaining adequate clearance in touch sensing traces.

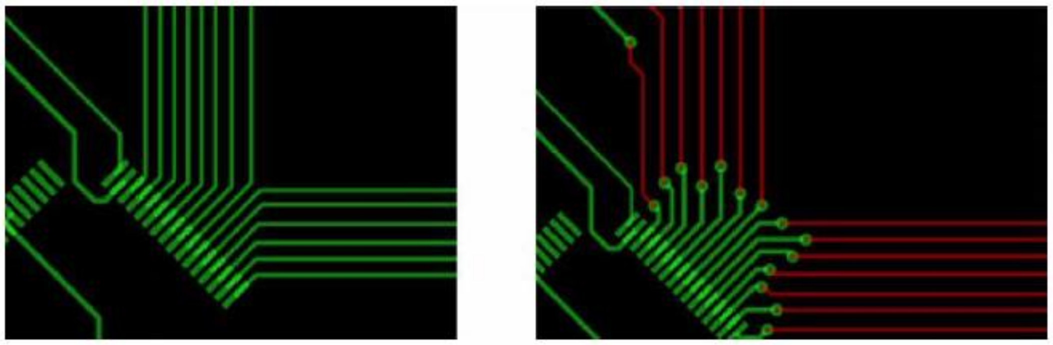


Figure 13. PCB trace routing

- Avoid crossover with another signal.
- Avoid routing under touch electrode, do not route traces directly under any touch pad.

6.3 Ground plane

A ground plane prevents coupling of external electromagnetic interference to the touch sensing electrodes, and acts as a shield for undesired electric fields. However, too many ground planes or simply filling open areas with ground planes might affect the sensor's sensitivity.

The following are a few recommendations and best practices for ground planes usage.

- Use X-hatch pattern on the top layer, 25% ground fill, 7mil line, 45mil spacing.
- Use X-hatch pattern on the bottom layer (e.g. underneath the electrodes area), 17% ground fill, 7mil line, 70mil spacing.

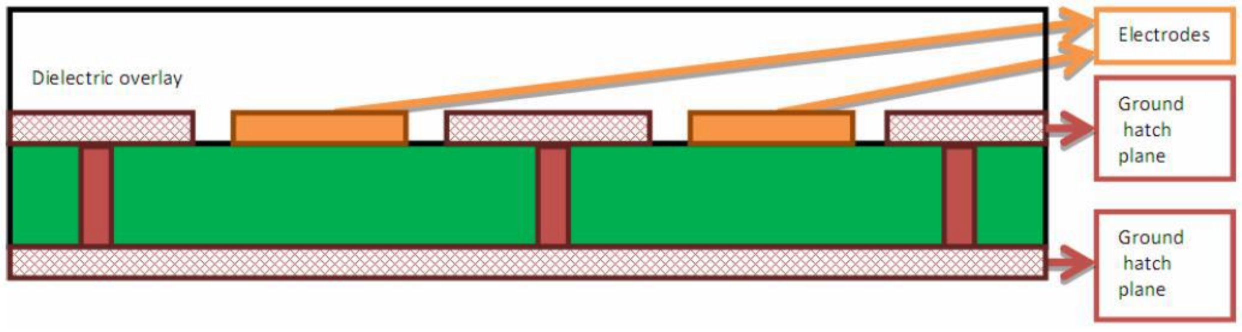


Figure 14. Ground plane

6.4 Electrodes placement

The following are recommendations for placing the touch sensing electrodes on a PCB or Flex-PCB.

- All touch electrodes should be placed as close to the MCU as possible. As the long trace loops in layout causes extra intrinsic capacitance and easily coupled noise, placing touch electrodes closer to the chip is always better.
- Components underneath electrodes — It is not recommended to place any component underneath the touch sensing electrode's area, especially in two layers board.
- Keep electrodes far away power module, RF antenna, etc.

6.5 Hardware checklist

The following is a checklist based on the recommendations in this application note. Before having a board, film, ITO, and the touch sensing board built, make sure the design follows all or most of these rules:

- GND return path is provided per specifications (GND hatch below or at least around the electrode keypad).
- No pull-ups present in CS-enabled (touch sensing input module) pins.
- Series resistors in cases where series current protection is desired should be lower than 100ohm.
- Make sure no signals that are not for touch sensing run parallel to the touch sensing signals. If signals need to go through the touch sensing traces, have them go in a different layer and perpendicular.
- Make sure to fill in ground between groups of traces (analog, digital, and touch), if possible, fill in ground between touch sensing traces.
- Traces as thin as the PCB or film technology will allow.
- Short traces (<300 mm. from electrode to MCU, ideally < 50 mm.)

Electrode shape corners as rounded as the layout will allow.

7 CS Software Configurations

Capacitive sense interface (CS) supports NXP touch library, the keypad, analog rotary and analog slider are implemented.

In NXP touch lib, the following configuration parameters need to be tuned based on specific design.

```
/* USAFA keydetector settings */
const struct nt_keydetector_usafa keydec_usafa =
{
/* Electrodes */
    .signal_filter = {2},
    .base avrg = {.n2 order = 12},
    .non activity avrg =  {.n2 order = NT FILTER MOVING AVERAGE MAX ORDER},
    .entry_event_cnt = 4,
    .signal to noise ratio = 4,
    .deadband cnt = 4,
    .min_noise_limit = 100,
};
```

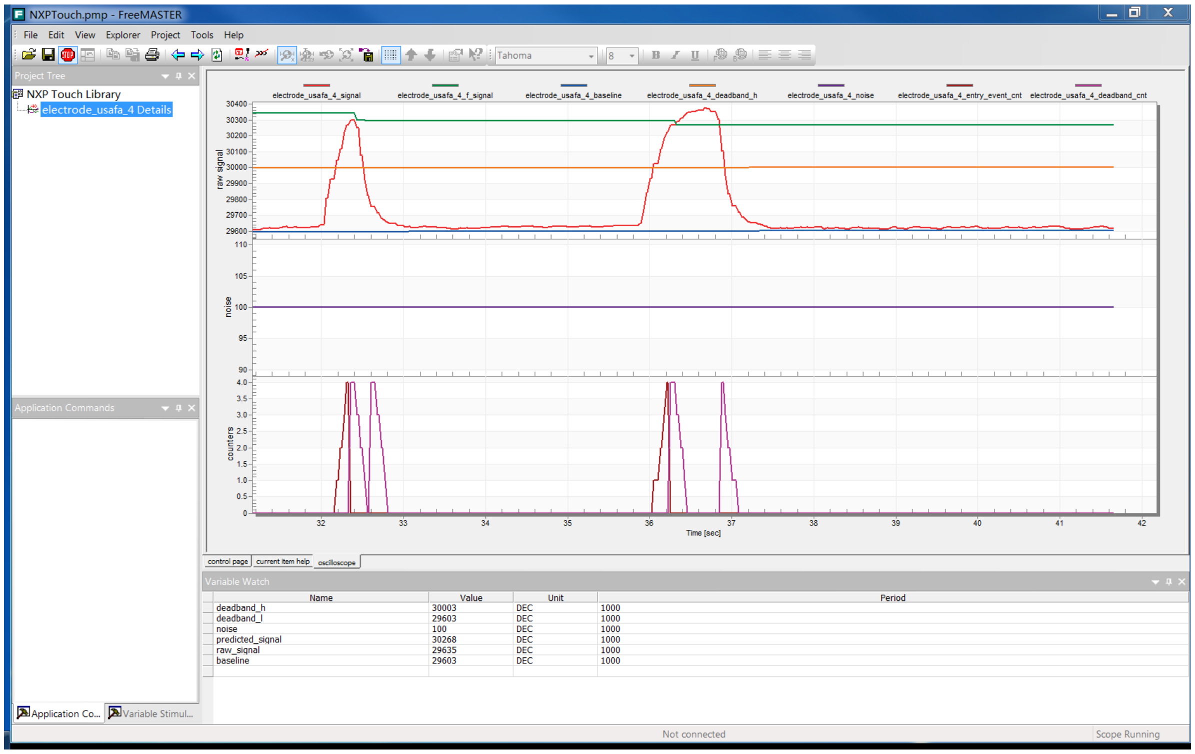
```
/* Self-cap config */
const cs_config_t hw_config =
{
    /* For acitve mode */
    .activeChannelEnable = CS_ACTIVE_CHANNEL_ENABLE,
    .activeClockDivider = CS_ACTIVE_CLOCK_DIVIDER,
    .activeDetectPeriod = CS_ACTIVE_DETECT_PERIOD,
    .activeIdlePeriod = CS_ACTIVE_IDLE_PERIOD,
    .activeOscFreq = CS_ACTIVE_OSC_FREQ,

    /* For low power mode */
    .lowPowerChannelNum = CS_LOWPOWER_CHANNEL_NUM,
    .lowPowerThreshold = CS_LOWPOWER_THRESHOLD,
    .lowPowerDebonceNum = CS_LOWPOWER_DEBONCE_NUM,
    .lowPowerIdlePeriod = CS_LOWPOWER_IDLE_PERIOD,
    .lowPowerOscFreq = CS_LOWPOWER_OSC_FREQ,
};
```

```
const struct nt_system system_0 = {
    .controls = &controls[0],
    .modules = &modules[0],
    .time_period = 50,
    .init_time = 400,
};
```

Where, the time_period should be greater than the channel scan and idle time.

The following is the debug information for FreeMaster, which can be used to calibrate the above parameters.



Name	Value	Unit	Period
deadband_h	30003	DEC	1000
deadband_l	29603	DEC	1000
noise	100	DEC	1000
predicted_signal	30268	DEC	1000
raw_signal	29635	DEC	1000
baseline	29603	DEC	1000

Figure 15. FreeMaster debugging information

If you need to print debug information from a serial port, modify the following macro definition:

```
#define NT_FREEMASTER_SUPPORT 0
#define FMSTR_DISABLE           1   /* To disable all FreeMASTER functionalities */
```

8 References

The following references are available on NXP website.

1. Designing Touch Sensing Electrodes (Document: AN3863)
2. QN908x user manual (Document: UM11023)
3. NXP Touch Library Reference Manual (Document: NT20RM)
4. NXP Touch Software Website
5. FreeMASTER Software Website

9 Revision History

Table 3. Revision history

Revision number	Date	Substantive changes
0	05/2018	Initial revision
1	03/2019	Changed the Application Note name from Capacitive Sensing Interface of QN908x to QN908x Capacitive Sensing Design Guide.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2018-2019 NXP B.V.

Document Number: AN12190
Rev. 1
03/2019