

---

**Atmel AVR3001: QTouchADC Measurement and Tuning**

---

**8-bit Atmel Microcontrollers****Introduction**

---

QTouchADC is the newest addition to the range of sensing algorithms available in the Atmel® QTouch® Technology. It is implemented by oversampling a standard ADC and requires only one pin per channel. The only external component required is a series resistor on the ADC input pin. Compared to standard QTouch technology, QTouchADC offers faster acquisition times with shorter burst lengths, resulting in lower power consumption. This acquisition method is available as part of the QTouch Library for selecting Atmel AVR®s. It has also been implemented on certain touch sensor ICs. This application note aims to familiarize the reader with QTouchADC measurement technology and sensitivity tuning techniques.

## Table of Contents

---

1. QTouchADC basics .....	3
2. Sensitivity tuning .....	4
2.1 Ensure full charge transfer .....	5
2.1.1 Coin-scope method.....	5
2.1.2 CSD adjustment method.....	7
2.2 Adjust pulse-scale factor .....	7
2.2.1 Pulse factor.....	7
2.2.2 Scale factor.....	8
2.2.3 Setting the pulse-scale factor.....	8
3. Salient features .....	9
3.1 Single pin per channel.....	9
3.2 Simple design.....	9
3.3 No external $C_s$ .....	9
3.4 Noise immunity.....	9
3.5 Faster acquisition .....	9
3.6 Proximity sensing .....	9
3.7 Resilient to $AV_{CC}$ changes.....	9
4. References.....	9

## 1. QTouchADC basics

The QTouchADC acquisition method works by sharing charge between the ADC's internal sample-and-hold capacitor ( $C_s$ ) and the sense electrode capacitance ( $C_x$ ).

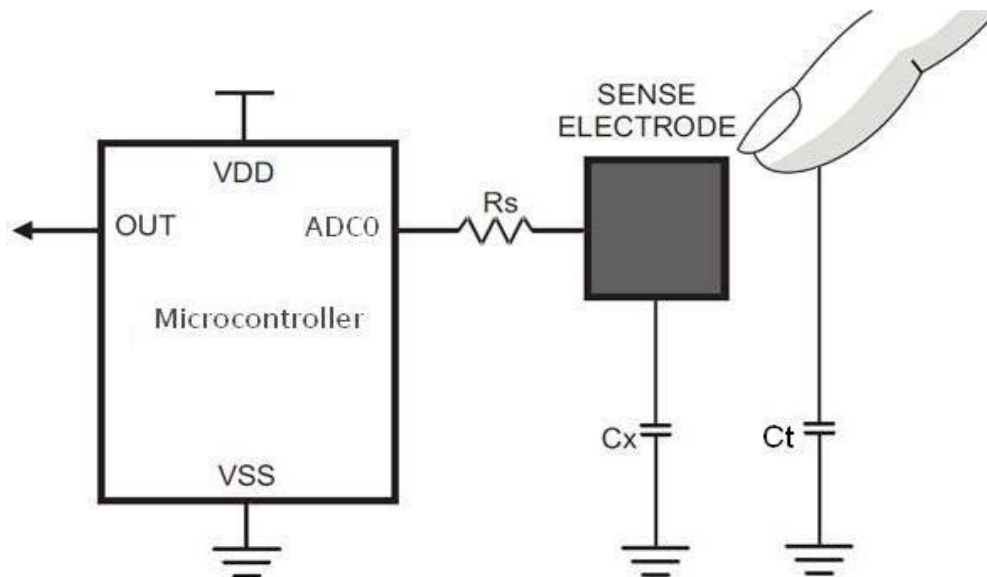
The charge sharing is carried out in the following sequence:

- Step 1.**  $C_x$  is charged and connected in parallel with  $C_s$ . This causes charge to flow between the capacitors until the charge is distributed proportionally between them.
- Step 2.** The resulting voltage across  $C_s$  is digitized by the ADC.
- Step 3.**  $C_s$  is charged and connected in parallel with  $C_x$ . The charge is shared between the capacitors again.
- Step 4.** The voltage across  $C_s$ , due to the remaining charge, is digitized by the ADC.

The two ADC values obtained are averaged and this gives the signal on the key. The long term average of the signal without touch is called the reference.

When the sensor is touched the effective capacitance of the sensor electrode increases and becomes  $C_x + C_t$ . This affects the amount of charge shared between the capacitors. When pre-charging  $C_x$  and sharing with  $C_s$ , charge transferred to  $C_s$  increases on touch and ADC input voltage increases. When pre-charging  $C_s$  and sharing with  $C_x$ , charge remaining on  $C_s$  decreases on touch and ADC input voltage decreases. But the resulting signal from the averaged ADC values increases on touch. If the difference between signal and reference is greater than the user-determined threshold (delta), a touch is reported.

Figure 1-1. QTouchADC block diagram.



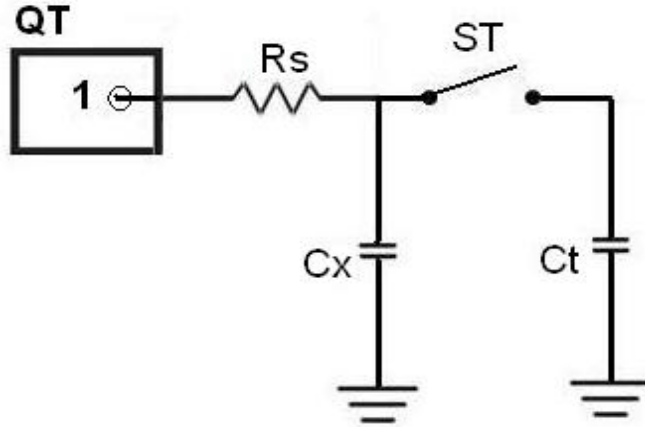
Since an internal sampling capacitor is used, only a single physical pin is required per channel. The differential nature of the measurement provides a level of low frequency noise rejection.

It should be noted that  $C_x$  and  $C_t$  are not physical capacitors.  $C_x$  is the effective capacitance of the sense electrode and  $C_t$  is the effective capacitance of the human finger touching the sensor.

## 2. Sensitivity tuning

The touch sensor circuit can be represented as shown in Figure 2-1.

Figure 2-1. Equivalent charge sharing circuit.



The touch sense pin (QT) is connected through series resistor  $R_s$  to the sensor electrode capacitance, represented by  $C_x$ . The switch ST represents a finger touching the key. The capacitance introduced by the finger is represented as  $C_t$ . When the key is touched,  $C_t$  is switched into the circuit forming a parallel capacitor with  $C_x$ , changing the effective sensor capacitance to  $C_x + C_t$ . In the following discussion  $C_{xt}$  is used to represent the sum of  $C_x + C_t$ . The value of  $C_x$  should be close to that of  $C_s$ . If  $C_x$  is significantly larger than  $C_s$  then the ADC will saturate. For best performance it is recommended that  $C_{xt}$  should not be greater than  $\sim 60\text{pF}$ .

The series resistor  $R_s$  is nominally  $1\text{k}\Omega$ , but it may be increased to higher values to improve the noise immunity of the circuit. The value of  $R_s$  should be increased in steps to find the lowest value that provides adequate noise immunity. Resistance values of up to  $100\text{k}\Omega$  have proven to be useful in extremely noisy environments.

The time constant  $\tau$  to charge  $C_{xt}$  to 63% through  $R_s$  is given by:

$$\tau = R_s \times C_{xt}$$

In the extreme case given above, the time constant is

$$\tau = R_s \times C_{xt} = 100 \times 10^3 \times 60 \times 10^{-12} = 6 \times 10^{-6} \text{ that is, } \tau = 6\mu\text{s}$$

As the voltage across  $C_{xt}$  rise asymptotically<sup>1</sup>, the charging current, and therefore the rate of transfer of charge through  $R_s$  falls asymptotically. After a charge time delay of  $4\tau$  the voltage across  $C_{xt}$  will be 98% of the applied voltage and the charge will be 98% of  $V \times C_{xt}$ .

The specific parameters and operating conditions of the key will vary from design to design. A certain amount of tuning will be required to optimize the performance of the key in each design. The main steps involved in tuning a sensor are ensuring full charge transfer on  $C_x$ , setting the appropriate gain on the signal and setting the averaging factor.

Note: 1. An asymptote is a line whose distance to a give curve tends to zero. An asymptote may or may not intersect its associated curve.

## 2.1 Ensure full charge transfer

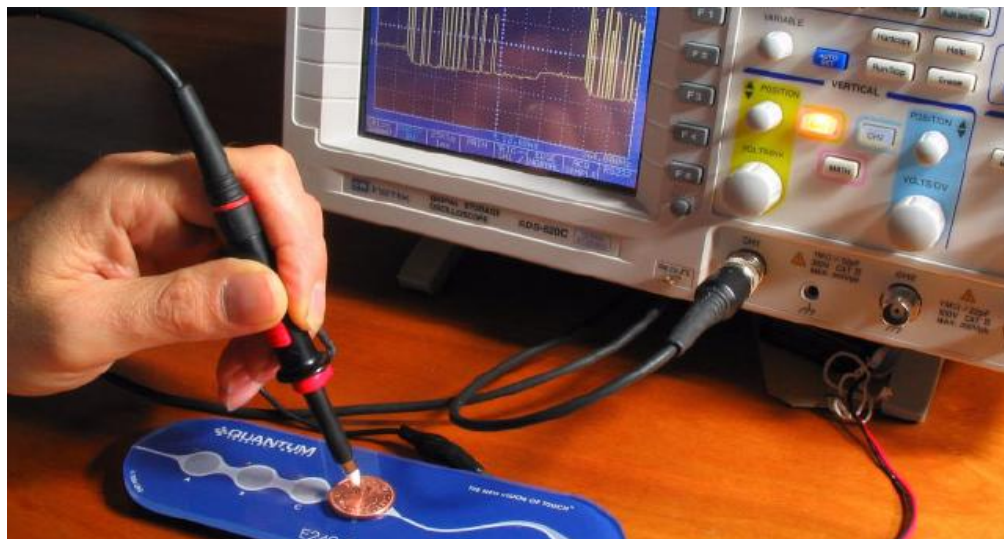
In order to ensure reliable operation, it is essential to ensure that full charge transfer is achieved at all the relevant stages of the measurement procedure. The largest time constant to be considered during a measurement cycle is the charging of  $C_{xt}$  through  $R_s$  in step 1. This will be a function of  $C_{xt}$  and  $R_s$  as described above. The time constant associated with the charge sharing phases of the measurement will be a function of  $R_s$  and the series combination of  $C_{xt}$  and  $C_s$  (that is  $(C_{xt} \times C_s)/(C_{xt} + C_s)$ ).

Hence with every change in  $R_s$ , the charging of  $C_{xt}$  needs to be measured to ensure full charge transfer. Two procedures are described to ensure that full charge is achieved at the sensor electrode. It is recommended that both methods are used to double-check that full charge transfer is achieved.

### 2.1.1 Coin-scope method

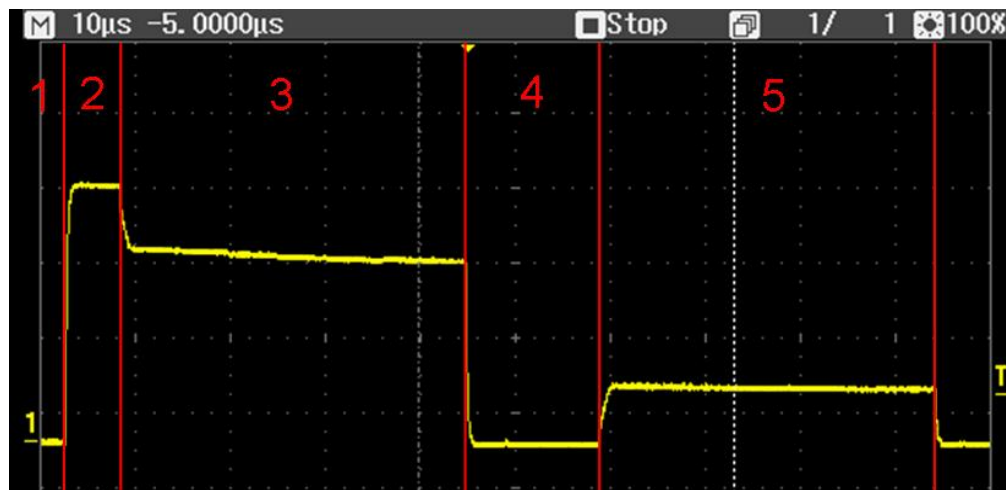
This method uses an oscilloscope to observe the pulses. The scope probe is coupled to the sensor electrode capacitively using a small coin on top of the overlying panel as shown in [Figure 2-2](#). Alternatively a small piece of copper tape can be used. Probing the sensor directly will add the capacitance of the probe and will give an unrealistic wave shape.

**Figure 2-2. Coin-scope method used to measure charge pulses.**



[Figure 2-3](#) represents the signal observed on the oscilloscope using the coin-scope method.

Figure 2-3. Complete charging on the sensor capacitance.



Five distinct regions can be identified in the signal:

- Region 1: Initially all the capacitors are discharged to eliminate any influence from previous measurements
- Region 2: The  $C_{xt}$  capacitor is pre-charged
- Region 3: Charge is shared from  $C_{xt}$  to  $C_s$  and the first measurement is performed
- Region 4: The  $C_s$  capacitor is pre-charged. This is not visible as sensor input is grounded during this phase
- Region 5: Charge is shared from  $C_s$  to  $C_{xt}$  and the second measurement is performed

The important aspect to check on this waveform is that the signal levels off least  $1\mu s$  before the end of region 2. A key with insufficient charge time will show a waveform with a sharp peak similar to that shown in Figure 2-4.

Figure 2-4. Incomplete charging on the sensor capacitance.



Where such a waveform is observed, the Charge Share Delay (CSD) time should be increased until a waveform similar to the one in [Figure 2-3](#) is observed. In [Atmel QTouch Libraries v4.4](#) and above the charge time can be increased by changing the value of the 'DEF\_QT\_CHARGE\_SHARE\_DELAY' parameter in the 'touch\_config\_dp.h' file associated with library file of the device. The value of this parameter can be varied from 1 to 255. Each unit of the parameter represents  $\sim 1\mu\text{s}$  of delay. CSD is a global parameter that affects all the channels.

### 2.1.2 CSD adjustment method

Proper charge transfer can also be verified without the use of an oscilloscope. To do this the Pulse and Scale factors should be set to 0. This is done by setting 'qt\_pulse\_scale[Ch] = 0X00, where Ch is the channel number.

Initially the CSD is set to the minimum value. Now the signal should be noted while a full touch<sup>1</sup> is applied. Next, with the CSD increased by  $1\mu\text{s}$ , the signal level for a full touch is noted again. If the signal level increases by more than 1 with the extended charge time, then it can be determined that the sensor was not fully charged with the previous setting. CSD should be increased incrementally until the signal remains stable from one setting to the next. This is the optimal signal level and indicates that the charge transfer is complete.

Note: 1. A full touch is where the sensor has the maximum capacitive loading by applied touch which would be present in the target environment and usage of the application. For example, a grounded metal plate may be placed on top of the cover dielectric over the keys.

## 2.2 Adjust pulse-scale factor

The gain and averaging factor need to be adjusted to provide an appropriate touch delta and appropriate level of noise rejection to the measured signal. These two factors can be adjusted by changing the value of the 'qt\_pulse\_scale[Ch]' parameter, where 'Ch' is the channel number. The qt\_pulse\_scale[Ch] parameter can be assigned an 8-bit value where the high nibble indicates the 'Pulse' value and the low nibble sets the 'Scale' value.

### 2.2.1 Pulse factor

The Pulse setting configures the number of measurements to be carried out in each cycle, with the resulting values being accumulated to obtain measurement gain. In order to provide for an extremely wide dynamic range of  $C_x$  measurement, Pulse is implemented as a power of 2, that is,  $2^{\text{Pulse}}$  measurements are accumulated. Each incremental increase of Pulse doubles the measured signal and also the touch delta.

**Table 2-1. Pulse vs. number of measurements.**

Pulse	Number of measurements
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
A	1024
B	2048
C	4096

Pulse	Number of measurements
D	8192
E	16384
F	32768

### 2.2.2 Scale factor

The Scale setting provides a division factor for the accumulated signal (accumulated for  $2^{\text{Pulse}}$  measurements). Its purpose is to provide an averaging factor which will reduce or eliminate the effects of noise on the measured signal. To match the dynamic range achieved by the pulse, the scale follows the same  $2^n$  implementation.

**Table 2-2. Pulse vs. averaging factor.**

Pulse	Averaging factor
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
A	1024
B	2048
C	4096
D	8192
E	16384
F	32768

### 2.2.3 Setting the pulse-scale factor

With Scale = 0 (that is, divide by 1), Pulse should be used to configure the gain of each sensor key such that the delta seen on touch is in the range of 20 to 100.

Scale should be increased incrementally in order to achieve the desired level of signal averaging. In order to maintain the gain for which Pulse was configured, Pulse should be incremented once for each increase in Scale.

For example, it may be found that the required touch delta can be achieved with Pulse = 4 (16 measurements), and Scale = 3 (divide by 8). This gives `qt_pulse_scale[Ch] = 0X43`.

The same touch delta will be achieved with Pulse = 3 (8 measurements) and Scale = 2 (divide by 4) or any other settings with the same ratio of accumulated pulses to scale factor. It should be noted that increasing the Pulse factor will also increase the measurement time and response time. The default value of the Pulse-Scale factor is `qt_pulse_scale[Ch] = 0X21`.



### 3. Salient features

Following are some of the salient features of the QTouchADC acquisition method.

#### 3.1 Single pin per channel

QTouchADC can be implemented on devices with an ADC peripheral. The device can support as many keys as the number of ADC inputs. When only a single key is configured the consecutive key needs to be reserved as a partner key. The partner key needs to be left unconnected. The partner key assignment is handled internally when two or more keys are configured.

#### 3.2 Simple design

The sensor design for QTouchADC is very simple. A copper fill connected to the pin through a series resistor will function as a touch sensor. Shape, size and tracking considerations are same as those specified for standard QTouch sensors. Further details can be found in the application note '[QTAN0079 Buttons, Sliders and Wheels Sensor Design Guide](#)' from Atmel.

#### 3.3 No external $C_s$

Since the internal sample and hold capacitor of the ADC is used there is no need to add an external  $C_s$  capacitor. Only a single resistor ( $R_s$ ) needs to be added on the ADC input pin. Thus reducing the number of external components used for each key.

#### 3.4 Noise immunity

As described in Chapter 1, two pulses are used to carry out each acquisition. The process of averaging the signal over the two differential pulses inherently reduces low frequency noise present in the signal. Oversampling and averaging can also be used to reduce the effects of other types of noise.

#### 3.5 Faster acquisition

The number of pulses for a reliable QTouchADC measurement is generally about 8 or 16 for a standard key. The number of pulses required may increase to some extent depending on application and environment. Conventional QTouch acquisition method can use hundreds of pulses for each acquisition. Hence the measurement time for each QTouchADC channel is much lower. With the device running at 4MHz, each acquisition takes 100-300 $\mu$ s.

#### 3.6 Proximity sensing

QTouchADC can be used to make proximity sensors whose performance is better than that of standard QTouch proximity sensors.

#### 3.7 Resilient to $AV_{CC}$ changes

The ADC is referenced to the same supply voltage used to charge the capacitors. Therefore the measured signal is independent of  $AV_{CC}$ . Over long term variations of  $AV_{CC}$ , only 5% variation is observed on the signal.

### 4. References

- [1] [Atmel QTouch Library User Guide](#).
- [2] [QTAN0079 Buttons, Sliders and Wheels – Sensor Design Guide](#).

**Atmel Corporation**

1600 Technology Drive  
San Jose, CA 95110  
USA

**Tel:** (+1)(408) 441-0311

**Fax:** (+1)(408) 487-2600

[www.atmel.com](http://www.atmel.com)

**Atmel Asia Limited**

Unit 01-5 & 16, 19F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
HONG KONG

**Tel:** (+852) 2245-6100

**Fax:** (+852) 2722-1369

**Atmel Munich GmbH**

Business Campus  
Parking 4  
D-85748 Garching b. Munich  
GERMANY

**Tel:** (+49) 89-31970-0

**Fax:** (+49) 89-3194621

**Atmel Japan G.K.**

16F Shin-Osaki Kangyo Bldg.  
1-6-4 Osaki, Shinagawa-ku  
Tokyo 141-0032  
JAPAN

**Tel:** (+81)(3) 6417-0300

**Fax:** (+81)(3) 6417-0370

© 2012 Atmel Corporation. All rights reserved. / Rev.: 8497B-AVR-08/2012

Atmel®, Atmel logo and combinations thereof, AVR®, Enabling Unlimited Possibilities®, QTouch®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.