AN11023

Capacitive touch sensing using the LPC11xx

Rev. 1 — 3 February 2011 Application note

Document information

Info	Content
Keywords	LPC1112, LPC1100, Cortex M0, capacitive touch sensing
Abstract	This application note describes the design of a simple capacitive touch sensing method based on the LPC1100 microcontroller from NXP Semiconductors.

Revision history

Rev	Date	Description
1	20110203	Initial version.

Contact information

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

This application note describes a simple capacitive touch sensing method using an ADC input of the LPC11xx microcontroller.

The capacitive touch sensors used in this application note are areas of copper on the PCB of a PCF8883 evaluation board (see Fig 1). One of four available sensors is tied to a RC network and connected to an ADC input channel of the micro (see Fig 2).

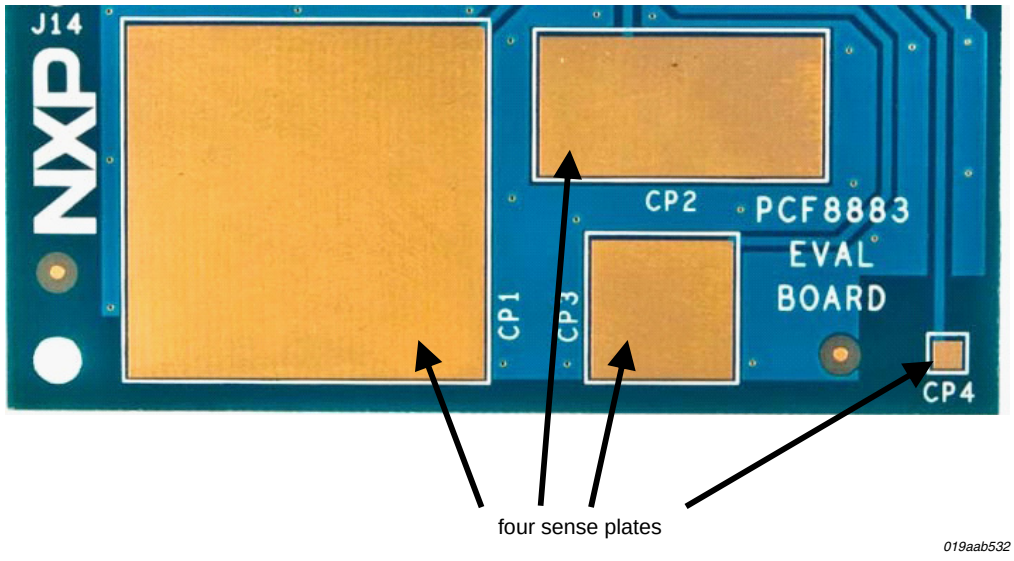

Fig 1. Four sensors at the PCF8883 evaluation board

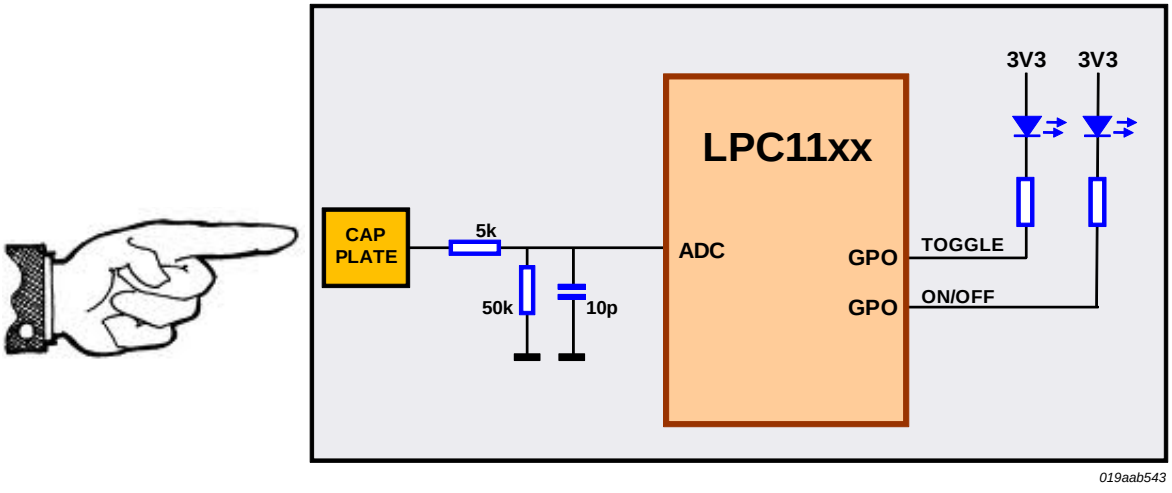

Fig 2. Capacitive touch sensing simplified block diagram

2. Way of working

Sensing requires only one pin, configured either as an ADC input or as a general purpose output. The reading process (see Fig 3) is performed in just a few simple steps.

Firstly configuring the I/O pin as a 'high' output. This will charge both the external capacitor (10pF) and the external capacitive plate.

Next the I/O pin is re-configured as an ADC input. This will cause the external capacitor and the capacitive plate to de-charge via both resistors (5k – 50k used in example). With the addition of a finger touching the sensor, the total capacitance will increase, and the discharging curve will be slower.

After that, an ADC conversion is started. A touch will cause the ADC reading to increase. In the code example, an average stable value is created for an un-touched condition, and then a deviation more than that will be what is detected as a touch.

Finally, the I/O pin is configured back again as a 'high' output (back to first step).

Sensing steps (see Fig 3):

1. Drive sensor line to VDD as digital output (charge)
2. Turn sensor line as ADC input (discharge) and start ADC conversion
3. ADC sample point. Sample and hold takes one ADC clock. After that 10 more clocks are needed to perform the full 10-bit ADC conversion (DONE bit set, reading result is in register LPC_ADC->DR[x])
4. Back to step 1 (and now check the reading for a sense plate touch, decode and de-bounce it and take action)

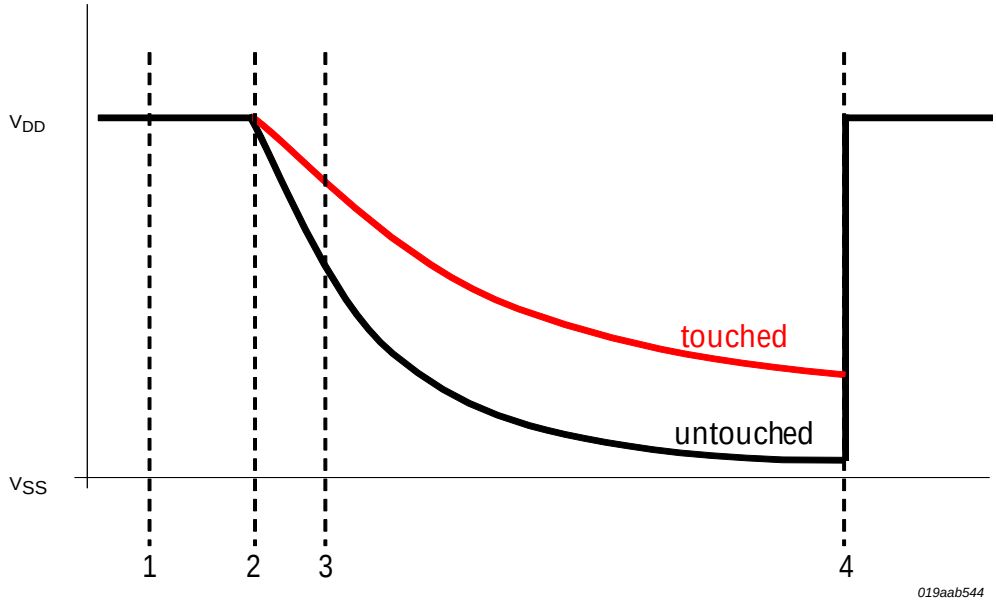

Fig 3. Sensor line input waveform

3. Demo

The LPC1100 example code listed below uses ADC input channel 1 (PIO1_0) as the sensing input line. Furthermore, to show the sensor switching behavior, two outputs connected to LEDs, are used.

One output (PIO3_2) operates in toggle mode (touch on, touch off).

The other output (PIO3_3) operates in momentary switch mode (like a push-button). The output is active (LED on) as long as the capacitive touch event lasts.

The software example is written in C language and compiled using Keil's uVision (MDK-ARM, V4.14) evaluation version compiler. For LPC11xx microcontroller configuration the standard CMSIS startup code (**startup_LPC11xx.s** and **system_LPC11xx.c**) from Keil were used and set as CCLK = IRC = 12 MHz

3.1 Source code listing

```
1   /*****************************************************************************
2   *  Title    : LPC11xx Capacitive Touch Sensing demo program
3   *  Hardware : MicroCore48 board + PCF8883 evaluation kit
4   *
5   *  1. Use Systick timer to generate a 10 msec timer tick (interrupt driven).
6   *  2. Capacitive sense plate is connected to P1.0 = AD1 input
7   *  3. Every 10 msec: use ADC to read the capacitive sense input
8   *  4. P3.3 = ON/OFF LED used to indicate press and release condition
9   *  5. P3.2 = toggle LED is used to indicate a new press condition
10  *****************************************************************************/
11  #include <LPC11xx.h>                          // LPC11xx definitions
12
13  static char k_press = 0;
14  static int  average = 0;
15
16  static short ADC_ReadCH1(void)                // read ADC channel AD1
17  {
18      LPC_IOCON->R_PIO1_0 = 2;                  // set sensor line as AD1 input
19      LPC_ADC->CR = 2 |                         // SEL = 2, select channel 1 on ADC
20                    (3 << 8) |                  // ADC_CLK = Fpclk / CLKDIV = 4 MHz
21                    (1 << 24);                  // start conversion
22
23      while (!(LPC_ADC->DR[1] & 0x80000000)) ;  // wait until end of AD1 conversion
24
25      LPC_IOCON->R_PIO1_0 = 0x81;               // sensor line is output high
26      LPC_ADC->CR &= 0xF8FFFFFF;                // stop ADC
27      return (LPC_ADC->DR[1] >> 6) & 0x3FF;     // return A/D conversion value
28  }
29
30  void SysTick_Handler(void)                    // SysTick Timer ISR every 10 msec
31  {
32    static char  debounce = 0;
33    static char  avgindex = 0;
34           char  result   = 0;
35           short reading;
```

```
36        reading = ADC_ReadCH1();                      // read AD1 = Cap sense input
37
38        if (reading > average + (average >> 4))       // above (average + 6% of average)?
39        {
40            if (debounce == 4)                        // debounce, 4 triggers for press
41            {
42                k_press = 1;                          // reached max, indicate pressed
43                result  = 1;                          // set result for return value
44            }
45            else
46                debounce ++;                          // still going toward max
47        }
48        else if (reading < average + (average >> 5))  // below (average + 3% of average)?
49        {
50            if (debounce == 0)
51            {
52                k_press = 0;                          // reached min, indicate release
53                result  = 0;                          // clear result for return value
54            }
55            else
56                debounce --;                          // going toward min
57        }
58
59        if (result == 0 && debounce == 0)             // recalculate average
60        {
61            if (++avgindex == 8)                      // average index delay
62            {
63                average = (reading + (15 * average)) / 16;
64                      avgindex = 0;
65            }
66        }
67    }
68
69    int main (void)
70    {
71      static char  toggle = 0;
72      static char  ledon  = 0;
73             short i;
74
75        SystemInit();
76        LPC_GPIO1->DIR |= (1<<0);                     // P1.0 connected to cap sense plate
77        LPC_GPIO3->DIR |= (1<<2);                     // P3.2 = toggle LED
78        LPC_GPIO3->DIR |= (1<<3);                     // P3.3 = ON/OFF LED
79
80        LPC_SYSCON->PDRUNCFG      &= ~(1<<4);         // disable pd bit to the ADC block
81        LPC_SYSCON->SYSAHBCLKCTRL |=  (1<<13);        // enable AHB clock to the ADC
82
83        for (i=0; i<200; i++)                         // warm up, establish average
84        {
85            average = (32 + ADC_ReadCH1() + (15 * average)) / 16;
86        }
```

```
87        SysTick_Config(SystemCoreClock/100);          // generate interrupt each 10 ms
88
89        while (1)
90        {
91            __wfi();                                  // go to sleep
92
93            if (k_press)                              // key pressed ?
94            {
95                LPC_GPIO3->DATA &= ~(1<<3);           // P3.3 low = LED ON
96                if (!toggle)
97                {
98                    toggle = 1;
99                    if (!ledon)
100                   {
101                       ledon = 1;
102                       LPC_GPIO3->DATA &= ~(1<<2);    // P3.2 low = LED ON
103                   }
104                   else
105                   {
106                       ledon = 0;
107                       LPC_GPIO3->DATA |= (1<<2);     // P3.2 high = LED OFF
108                   }
109               }
110           }
111           else                                       // key released
112           {
113               LPC_GPIO3->DATA |= (1<<3);             // P3.3 high = LED OFF
114               if (toggle)
115               {
116                   toggle = 0;
117               }
118           }
119       }
120   }
```

4. References

For further details please refer to the following publications:

- Datasheets / User Manuals / Application Notes / Example code:

http://ics.nxp.com/microcontrollers/

- AN10832: “PCF8883 - capacitive proximity switch with auto-calibration”:

http://www.nxp.com/documents/application_note/AN10832.pdf

- UM10370: “User Manual for the PCF8883 Evaluation Kit OM11055”

http://www.nxp.com/documents/user_manual/UM10370.pdf

5. Legal information

5.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

5.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

5.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

6. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 3 February 2011
Document identifier: AN11023