XC800 Family

AP08122

16-Button Capacitive Touch Interface with XC836T

Application Note

V1.1 2012-03

Microcontrollers

Edition 2012-03

Published by
Infineon Technologies AG
81726 Munich, Germany

Information

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements, components may contain dangerous substances. For information on the types in question, please contact the nearest Infineon Technologies Office.

Infineon Technologies components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

XC83x

Revision History: V1.1, 2012-03

Previous Version: none

Page	Subjects (major changes since last revision)

DAVE™ is registered trademark of Infineon Technologies AG.

We Listen to Your Comments

Is there any information in this document that you feel is wrong, unclear or missing?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com

Table of Contents

1 Introduction

Touch buttons, touch sliders and wheels are getting ever more popular in the latest Human-Machine Interface (HMI) designs. Using capacitive touch technology to create these input elements gives the user flexibility, an opportunity for high-level customization, a significant reduction in overall system cost and increased mechanical reliability.

The '16 Buttons Kit' from Infineon provides all the necessary components for evaluating capacitive touch techniques in button interface solutions. It contains 16 independent touch buttons and 16 independent LEDs for user feedback.

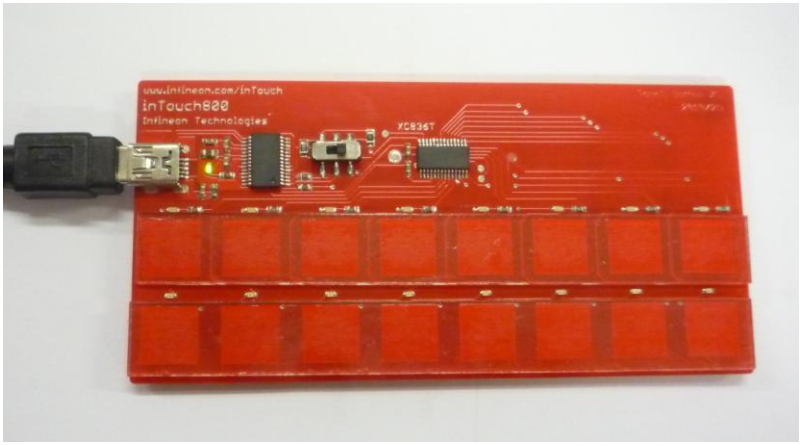

Figure 1 16 Buttons Kit

Embedded software is provided as part of the kit.

The DAVE™ Bench tool chain is based on Eclipse technology and can be downloaded free of charge from: http://www.infineon.com/dave-bench.

The tool chain includes:

- Compiler
- Flash loader
- Debugger
- IDE
- U-Spy – a real-time user interface

2 Hardware Setup and Basic Program Flow

2.1 Hardware Setup

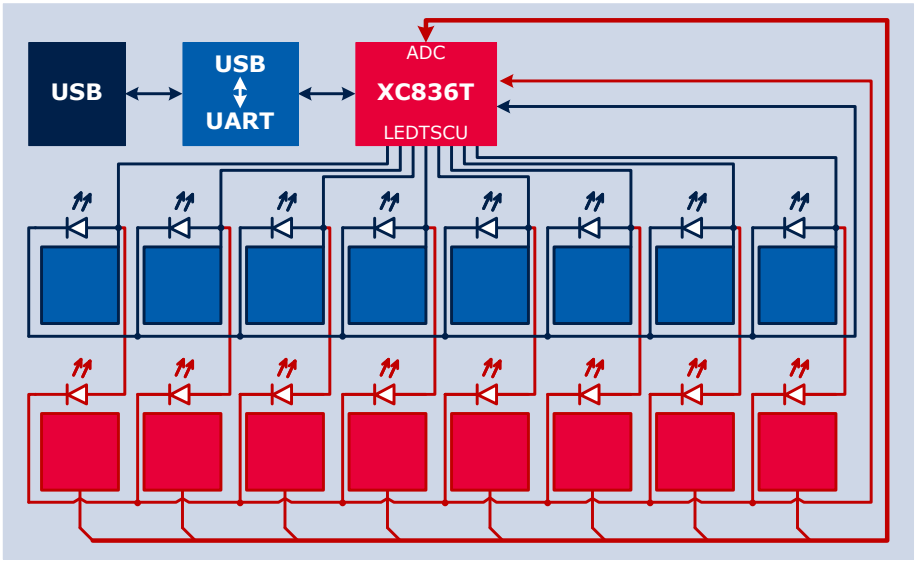

Figure 2 Block diagram

The ‘16 Buttons Kit’ comes on a standard PCB with two pieces of 2mm-thick plexi cover glued directly over the touch pad rows. Users can touch-and-hold (default) or tap the 16 buttons depending on the downloaded software.

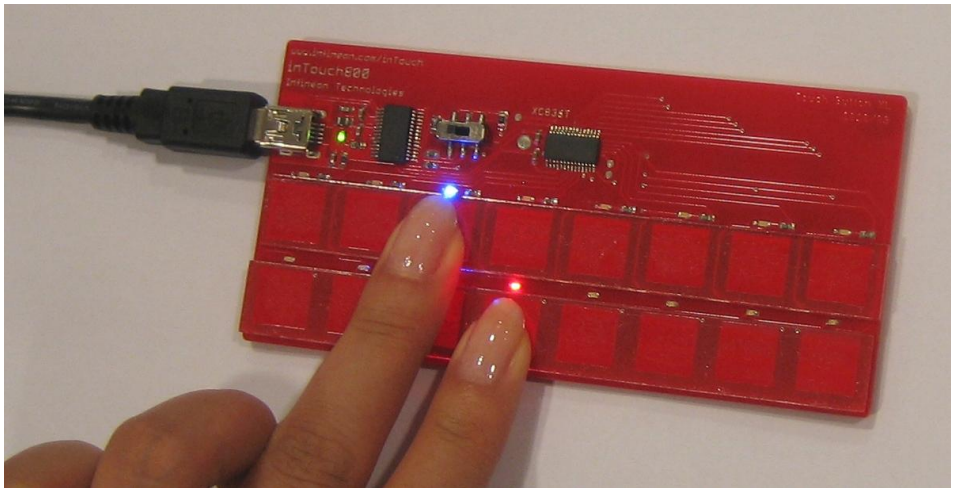

Figure 3 LEDs light up when the pads are touched

2.2 Program Flow

The microcontroller spends most time in idle mode to limit power consumption. The software is primarily interrupt driven. An external interrupt pin emulates a traditional Reset pin and is assigned the highest interrupt priority. The UART has medium priority because of the high data rate. Touch sense related tasks are performed with medium priority each time pad capacitances have been measured. LED updates have the lowest priority. The ADC (Analog-to-Digital Converter) interrupt, if enabled, also has high priority to facilitate accurate charge time measurements.

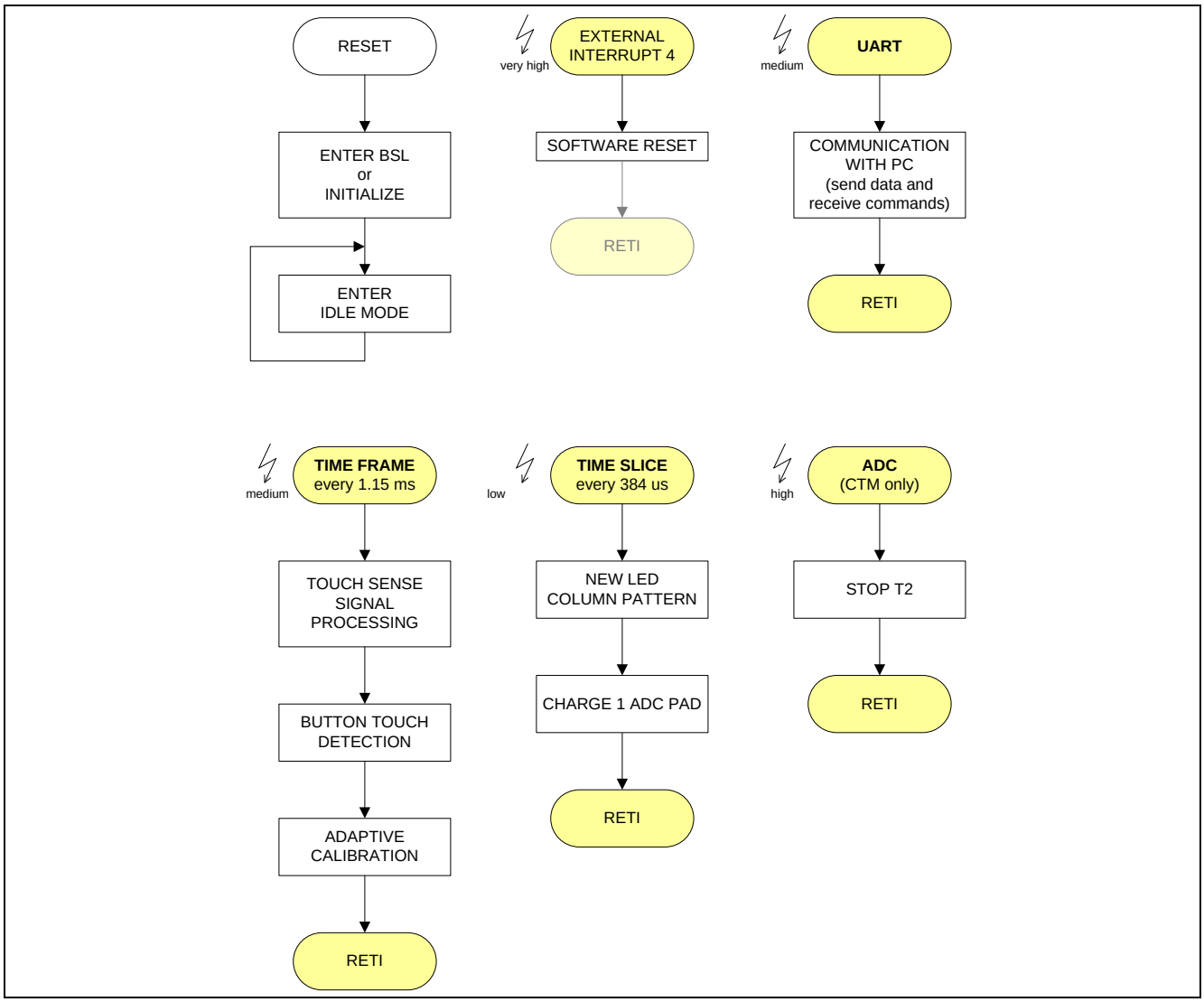

Figure 4 Overview of the main functions and interrupts of the 16 Buttons Kit

Only a few of the XC836T modules are required to perform all the required functions, with minimal CPU-load and shared interrupts.

- The LEDTSCU (LED Touch-Sense Control Unit) handles 8 touch pads and 16 LEDs; it generates a so-called *Time Frame* interrupt after every measurement where signal processing and touch detection take place and a so-called *Time Slice* interrupt after every LED column activation where the pattern for the next LED column is loaded into shadow registers
- The ADC handles the other 8 touch pads; in every *Time Slice* interrupt measurements are taken on a different touch pad and signal processing takes place in the *Time Frame* interrupt
- The UART module, which is part of the XC800 core, is used for full-duplex UART communication with the PC

3 Touch Interface

The *16 Buttons Kit* has 16 touch pads used as touch buttons. The top 8 pads are handled by the XC836T microcontroller's LEDTSCU module which is a dedicated touch-sense controller module. The bottom 8 pads are handled by the ADC module.

The main touch sensing functions, handled by software, are:

- Sample accumulation (ROM library + Flash)
- Signal filtering and moving average generation (ROM library + Flash)
- Touch detection (ROM library + Flash)
- Touch pad calibration (user software in Flash)

Capacitive touch sense control is a form of HMI where a touch pad controller constantly measures the capacitance of touch pads. Capacitance increases when a pad is touched and the touch pad controller detects it.

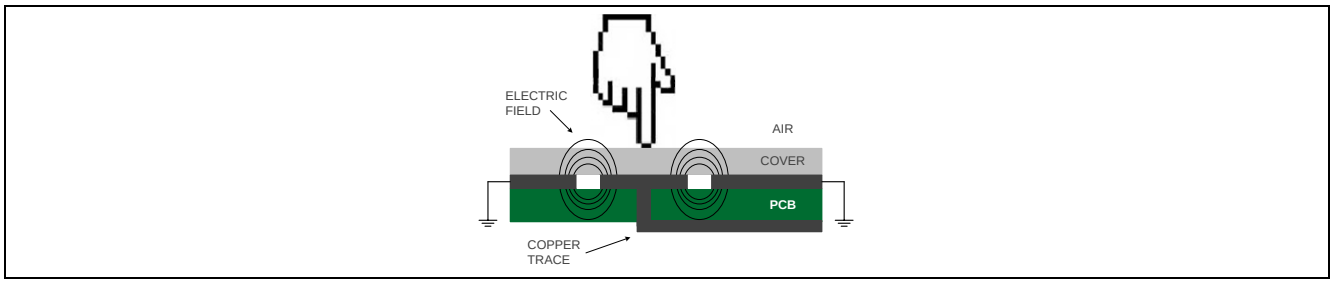

Figure 5 Touch pad with cover

Infineon offers three different solutions for measuring the capacitance of touch pads.

- Relaxation Oscillator (RO)
- Charge Redistribution (CR)
- Charge-Time Measurement (CTM)

3.1 Relaxation Oscillator (RO)

XC836T has a dedicated LED and Touch-Sense Controller module (LEDTSCU) implementing the relaxation oscillator topology on up to 8 channels. The top 8 touch pads of the *16 Buttons Kit* are handled by the LEDTSCU.

In the RO topology, a simple circuit generates oscillations on the touch pads and the module measures the oscillation frequency by counting the number of oscillations in a pre-defined monitoring window. A touched pad will have a lower oscillation frequency because the frequency is inversely proportional to the capacitance. The pre-defined monitoring window is called the oscillation window. It has to be wide enough to count enough oscillations for a high-resolution measurement, but to avoid counter overflow it must not be too wide. The oscillation counter is an 8-bit register so the maximum measurable number of oscillations is 255. A self-calibration routine (*adaptive calibration*) regularly adjusts the size of the oscillation window for optimal resolution.

The LEDTSCU regularly measures the frequency on every pad and the results are further processed by a software library that is provided by Infineon in the microcontroller's ROM ('ROM Library'). An adaptive averaging function in this library generates a moving average from the measurements and detects changes in the capacitance. All parameters can be configured by the user.

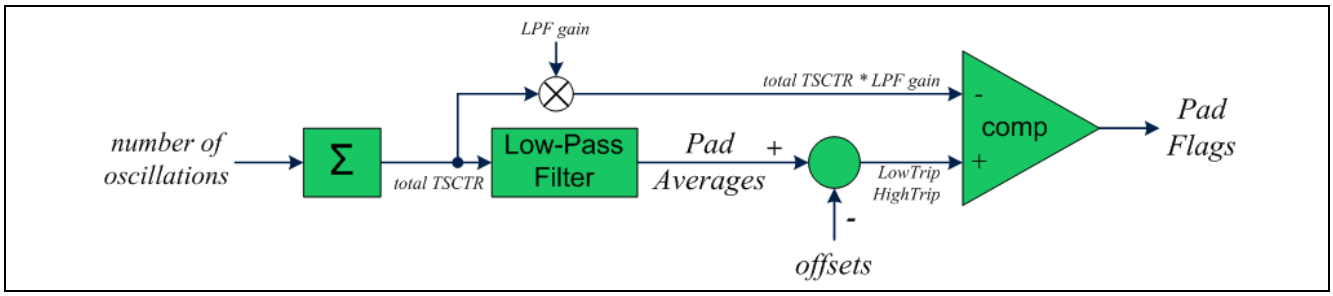

Figure 6 Adaptive average control

Controlled by the Touch Sense State Machine, certain variable flags will be set or cleared when a pad is touched, released, touched for too long or too short a time, or when touched for the correct amount of time.

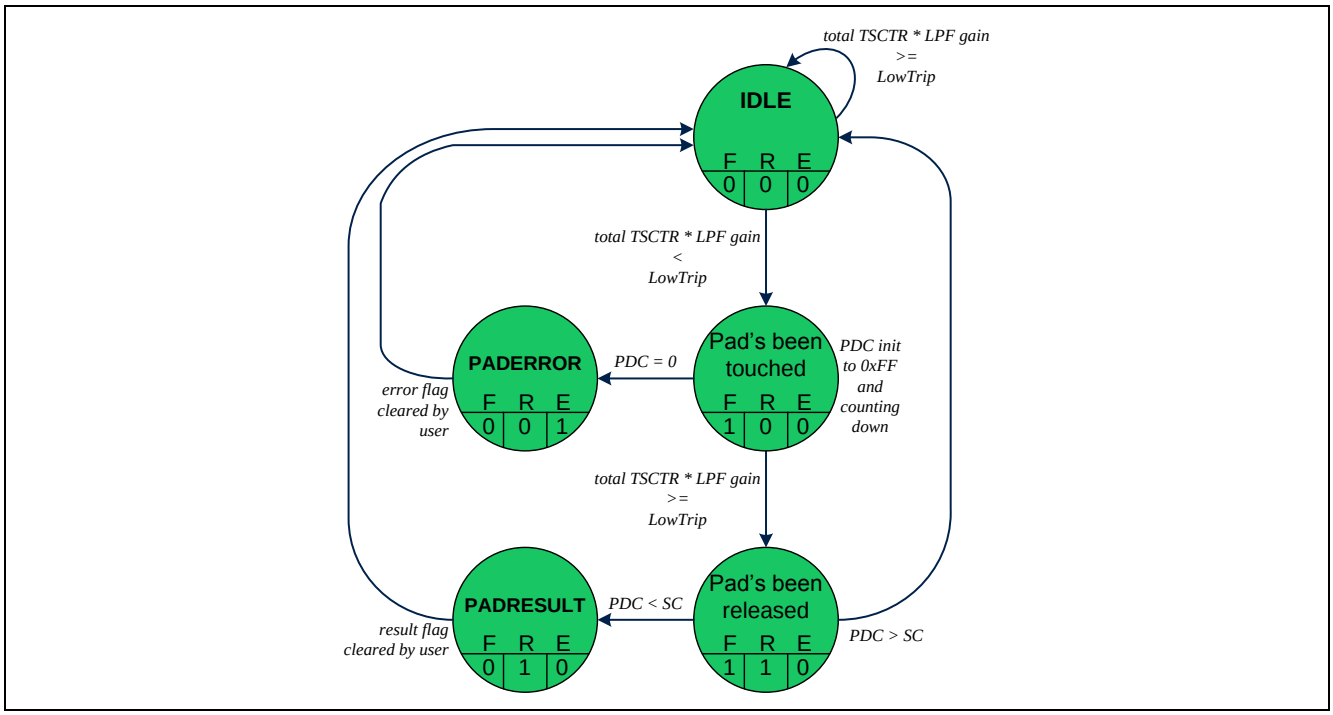

Figure 7 Touch sense state machine

When a pad is touched, the respective *PADFLAG* will be set by the *adaptive average control* routine. When the touch is released after the right amount of time, the respective *PADRESULT* flag will be set by the state machine. If the touch is never released, the state machine will assume that the pad has been stuck and a *PADERROR* flag will be set. *PADRESULT* and *PADERROR* support tapping control and their use is optional.

The LEDTSCU also functions as an LED matrix controller capable of controlling up to 64 LEDs. An LED matrix consists of many LEDs organized in lines and columns in various layouts where a maximum of one column is active at any given time, due to time multiplexing.

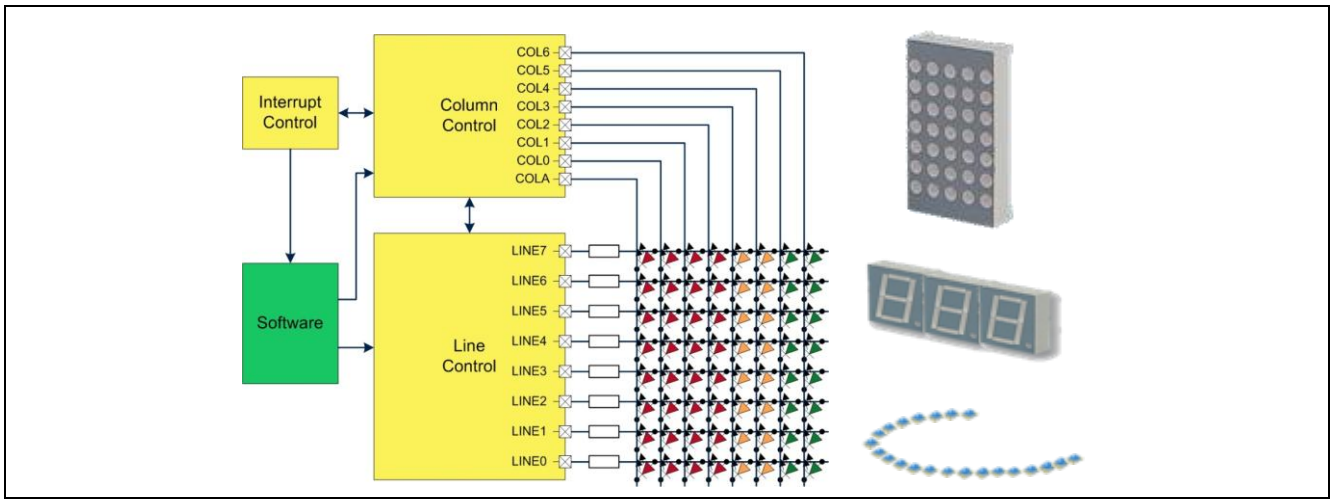

Figure 8 LED matrix controller

The touch sense pins are shared with the LED lines while the touch sense and LED control functions are combined in a time multiplexed manner. Since fewer pins are needed overall, a design of multiple touch buttons and LEDs is possible with a smaller microcontroller, a smaller PCB and reduced overall system cost. All 16 LEDs of the '16 Buttons Kit', organised into 8 lines and 2 columns, are controlled by the LEDTSCU.

3.2 Charge Redistribution (CR) and Charge-Time Measurement (CTM)

The CR and CTM methods use the 8 ADC channels of the microcontroller to measure the capacitance of touch pads. The bottom 8 touch pads are connected to the ADC.

Capacitance is measured by pumping charge onto the touch pads and observing the resulting voltage level. Touch pads are regularly checked one by one. Before the measurement, the pads are discharged and the sampling capacitor in the ADC is pre-charged to VAREF/2 (half the ADC reference voltage). Charge flows from the sampling capacitor to the touch pad during sampling until the voltage level is equalized. For the Charge Redistribution method, this voltage level is the result of the measurement. The larger the pad capacitance, the lower the measured voltage will be.

In the CTM method, the sampling is repeated without discharging the pad until the voltage level reaches a pre-defined threshold. The result of the measurement is the time between start of the first sample and the point at which the threshold is crossed, measured by a timer. The larger the pad capacitance, the longer this time will be.

Sampling takes place in the *Time Slice* interrupt. When the threshold is reached, an ADC interrupt is automatically generated. Timer 2 is used to measure the time it takes to charge the pads.

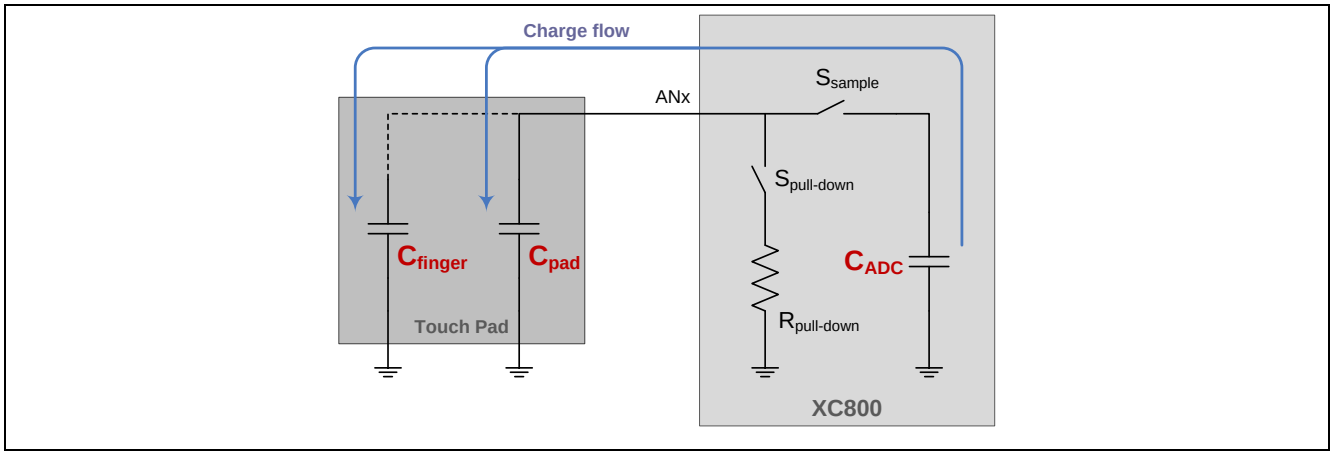

Figure 9 CR and CTM methods

The capacitance of every pad is regularly measured and the results are further processed by an enhanced adaptive average control function in a software library provided by Infineon. A moving average is generated and changes in the capacitance are detected. All parameters are configurable.

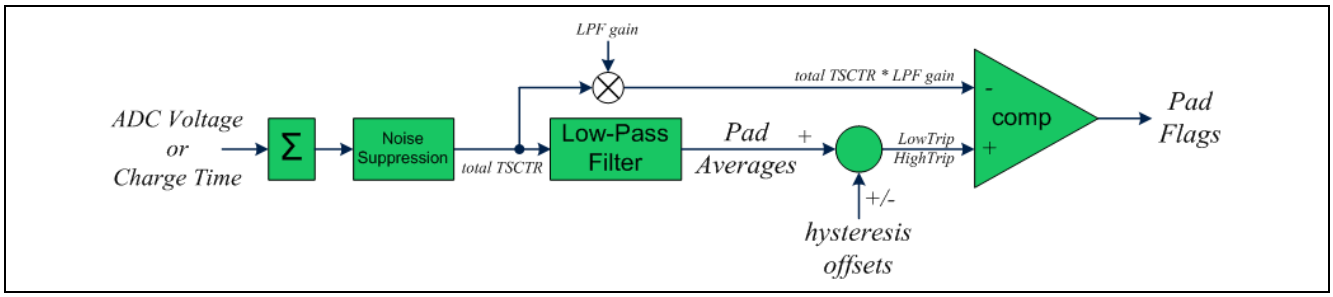

Figure 10 Enhanced adaptive average control

Flags in variables will be set or cleared by an updated touch sense state machine depending on the nature of the finger touch. The updated touch sense state machine has two versions. Version B is similar to the touch sense state machine in ROM library and it supports tapping control. Version A supports touch&hold control; when a pad has been touched long enough, a respective PADTOUCHED flag will be set. The PADTOUCHED flag will be reset when the pad has been left untouched long enough.

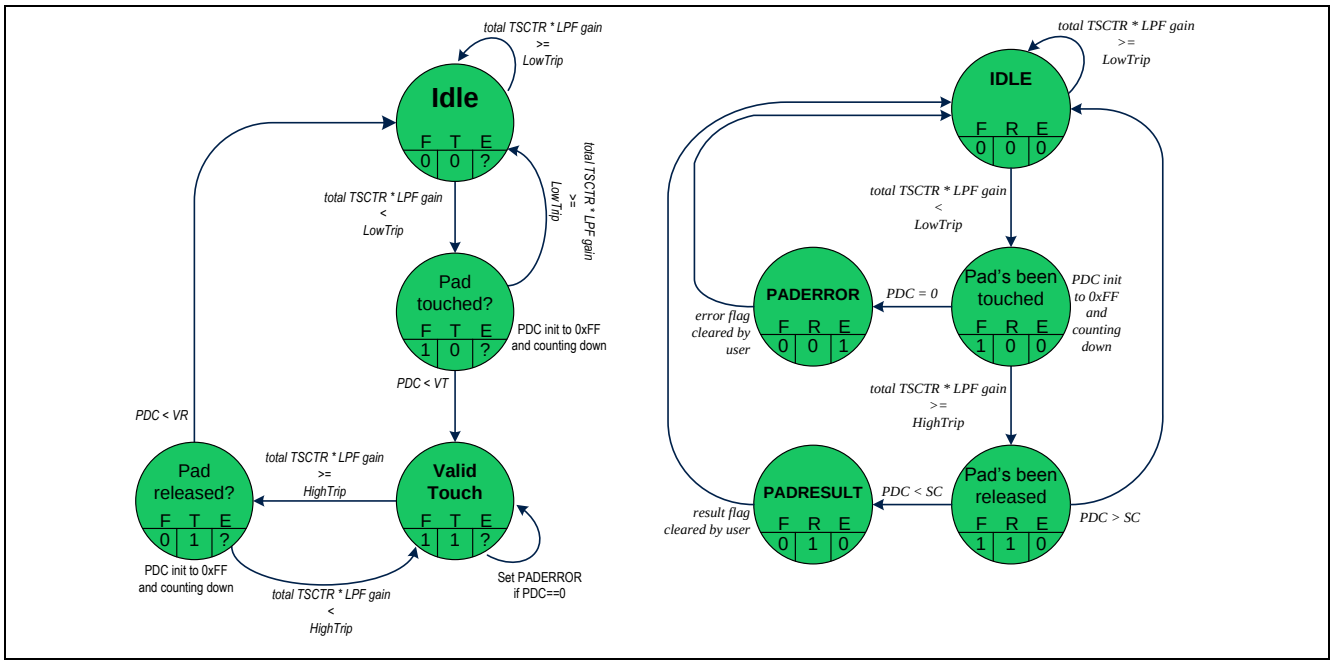

Figure 11 Updated touch sense state machine versions A and B

Enhanced adaptive average control and the state machine are executed in the *time frame* interrupt to easily integrate the CR or CTM methods with the RO method using shared interrupts.

3.3 Software Versions

The *16 Buttons Kit* has 4 different software versions:

1. The LEDs light up when respective pads are touched, using the CR method and library version A for the ADC touch pads (this is the default software)
2. LEDs toggle when respective pads are tapped, using the CR method and library version B for the ADC touch pads
3. The LEDs light up when respective pads are touched, using the CTM method and library version A for the ADC touch pads
4. LEDs toggle when respective pads are tapped, using the CTM method and library version B for the ADC touch pads

4 Programming Access

The board provides programming access to the microcontroller when the onboard switch is set to the “LOAD” position. It contains an FTDI chip, FT232RL, which acts as a USB-to-UART bridge and also controls the microcontoller’s emulated MBC (P3.0) and RESET (P1.2) pins. Programming access is wired for half-duplex UART on pin P3.2. Flash content can be modified with the XC800 FLOAD tool which is integrated into DAVE™ BENCH and is also available in a stand-alone version.

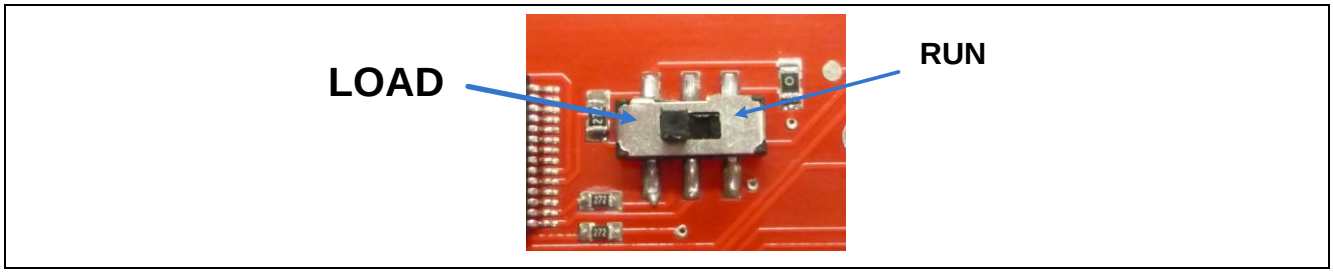

Figure 12 Onboard switch in LOAD position

The XC836 boot configuration does not depend on any pin status during reset. Instead, a Boot Mode Index (BMI) configuration determines the entry to various boot modes such as User Mode, Boot-Loader (BSL) Mode and On-chip Debug (OCDS) Mode. After reset, the BMI value is read and the respective boot mode entry is automatically executed.

The onboard microcontroller is programmed to “User Mode (Productive)”. In this mode, the Boot ROM jumps to the program memory address 0x0000 on startup to execute the user code in the Flash memory. This mode provides Flash memory protection from external access (read/write).

Changing the BMI value to enter another boot mode is achieved by programming a specific code embedded in the user code. It is located in MAIN.C at the *(MAIN_main,2)* section. This section only gets executed once after RESET. This code checks GPIO pin P3.0 (MBC emulation) and if it is low, a Boot ROM routine is called to re-enter UART Boot-Loader Mode. Once UART Boot-Loader Mode is entered, the user can change the contents of the Flash memory. This specific user code must be present in Flash and executed under certain conditions to ensure that programming the microcontroller is possible.

RESET happens during power-on and is also emulated on pin P1.2. Pulling down P1.2 momentarily generates an interrupt (External Interrupt 4) and the interrupt service routine generates a soft RESET. The FTDI chip on board the *USB Docking Station* automatically controls these pins when the user tries to access the microcontrollers with FLOAD.

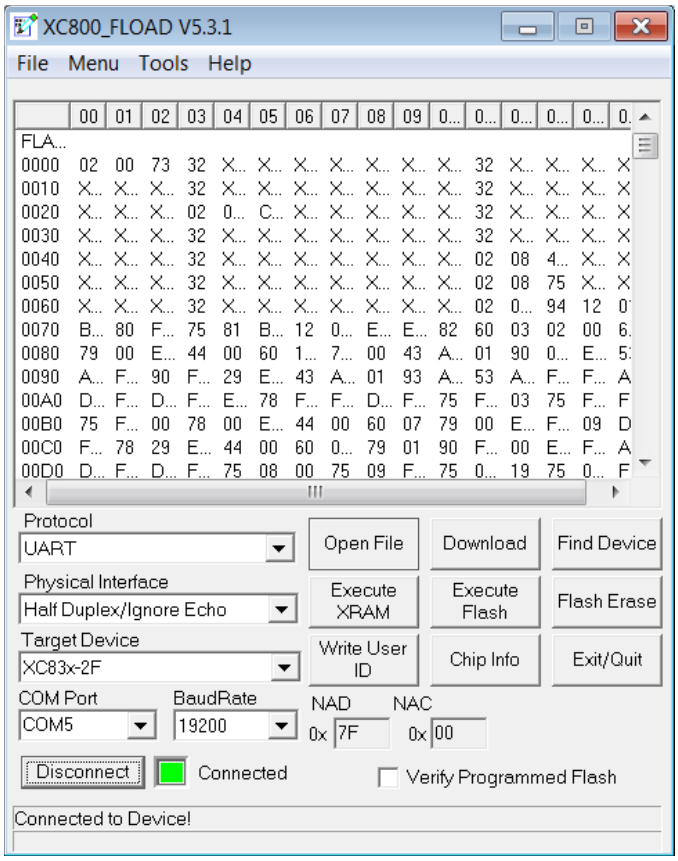

Figure 13 XC800 Fload

5 Monitoring

5.1 U-SPY

U-SPY is a UART terminal program which allows the user to view a serial communication through a PC serial port. Its features include transmission of a byte or group of bytes, configuration of protocol for bytes transmission/reception and creation of dedicated control buttons, display fields, progress bars and an oscilloscope for better visualization.

For more information, please refer to the Help menu in U-SPY.

The U-SPY can be launched directly from DAVE™ Bench by clicking on the  icon.

5.2 Settings

The custom configuration and user interface for a particular task or application can be saved in the format "*xxx.ini*". This allows specific setting files to be shared among users.

For the *16 Buttons Kit*, two *.ini* files have been configured, **tspad1_8.ini** and **adcpad1_8.ini**. Serial communication is via **full-duplex UART** protocol at a baudrate of **500 kbps**.

*Note: Ensure that the 16 Buttons Kit mode is switched to "**RUN**" (**Figure 14**) before running any of the monitoring routines.*

In "RUN" mode, the UART interface will be full-duplex, whereas in "LOAD" mode, the UART interface will be half-duplex.

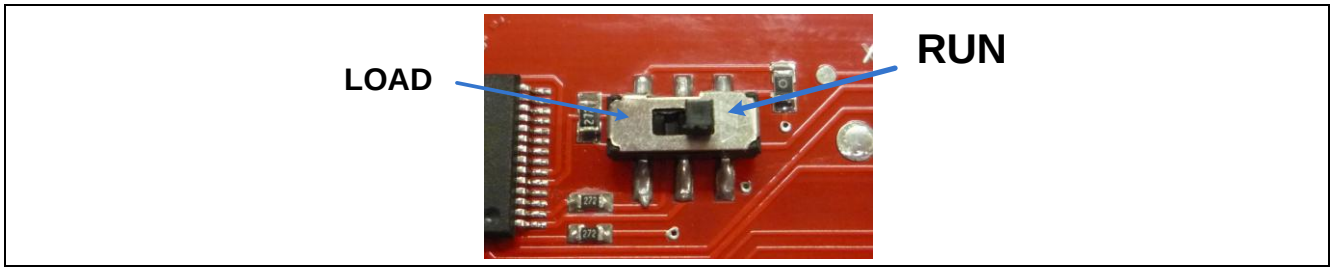

Figure 14 16 Buttons Kit connected to the PC with switch in RUN position

5.3 UART Interrupt

Any data transmission to or from U-SPY will trigger the UART interrupt in the XC836 microcontroller. Checks are performed during the interrupt to determine whether data is to be transmitted or received. The data transmit or receive process is then carried out automatically.

5.4 tspad1_8.ini

This settings file (**Figure 15**) is customized to allow the user to monitor the parameters of the LEDTS ROM Library.

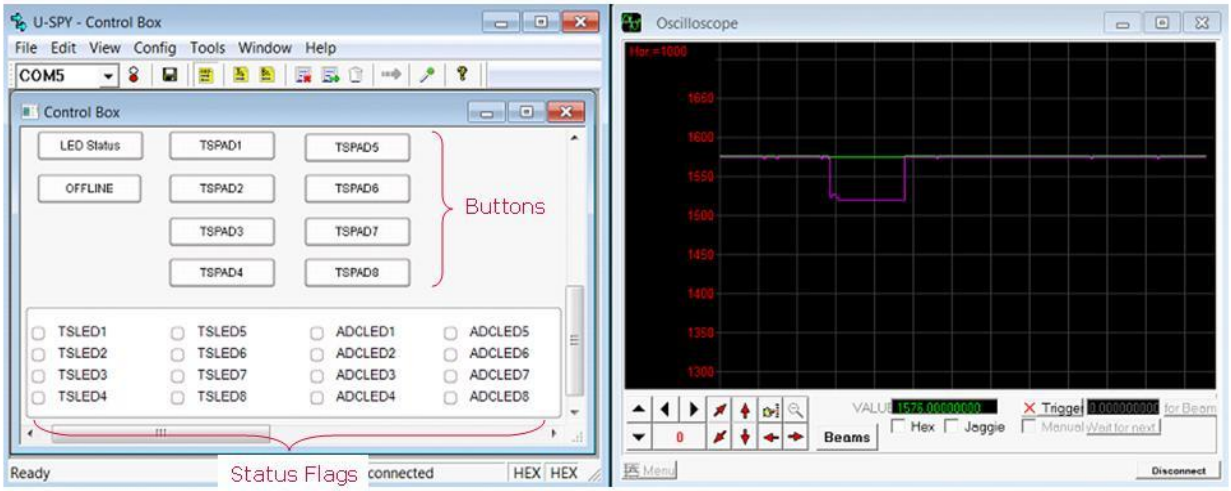

Figure 15 tspad1_8.ini User Interface

5.4.1 Data Format

Buttons

The buttons are used to transmit data from U-SPY to the XC836 microcontroller. Data is transmitted in the following format (**Table 1**):

Table 1 U-SPY to Microcontroller Transmitted Data Format

	D0	D1
Value (hex)	08	XX
Description	I.D.	Button Number

Based on this data received by the XC836 microcontroller, the display mode can be switched from monitoring of of the LED status via software or hardware to monitoring of the LEDTS pad average values.

Oscilloscope

The oscilloscope function allows the user to monitor the LEDTS pad average values and accumulated values at the same time.

Based on the button selected by the user, the XC836 microcontroller will capture the average and accumulated values of the selected pad and transmit them out to U-SPY which will then display the values out on the oscilloscope. The signals displayed in each mode are as follows (**Table 2**):

Table 2 Signals Displayed for Left/Right Button

	Signal1	Signal2
Description	Pad Total_TSCTR * $2^{DIVISORN}$	Pad Average
Colour	Green	Pink

Status Flags

The format of the transmitted data for the status flags is as follows (**Table 3):**

Table 3 Transmit Data Format for Status Flags

	D0	D1	D2
Value (hex)	84	30/31	XX
Description	I.D.	TSLED – 30 ADCLED - 31	Mask

The statuses of the LEDs received by U-SPY are masked before they are displayed as status flags. It is important that the bits of a mask do not overlap with the bits of another mask. This is to ensure that status flags are not falsely turned on. The masks used are as follows (**Table 4**):

Table 4 LED masks for Status Flags

LED Number	1	2	3	4	5	6	7	8
Mask (hex)	01	02	04	08	10	20	40	80

5.5 adcpad1_8.ini

This settings file (**Figure 16**) is customized to allow the user to monitor parameters of the ADC Library.

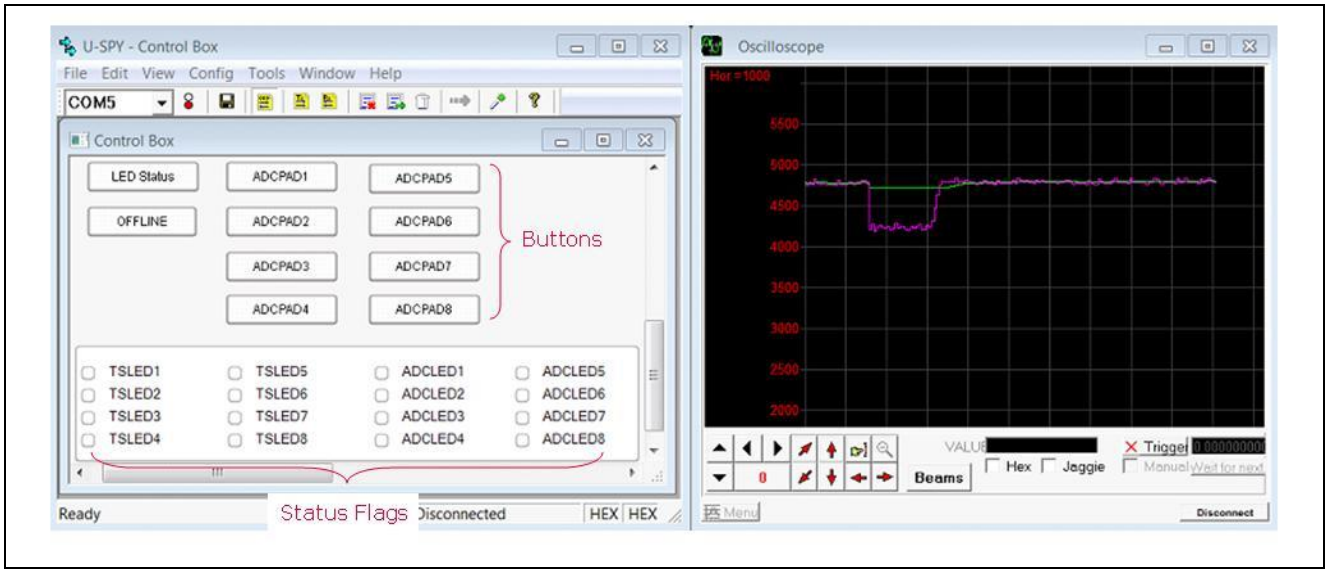

Figure 16 adcpad1_8.ini User Interface

5.5.1 Data Format

Buttons

The buttons are used to transmit data from U-SPY to the XC836 microcontroller. Data is transmitted in the following format (**Table 5**):

Table 5 U-SPY to Microcontroller Transmitted Data Format

	D0	D1
Value (hex)	08	XX
Description	I.D.	Button Number

Based on this data received by the XC836 microcontroller, the display mode can be switched from monitoring of of the LED status via software or hardware to monitoring of the ADC pad average values.

Oscilloscope

The oscilloscope function allows the user to monitor the ADC pad average values and accumulated values at the same time.

Based on the button selected by the user, the XC836 microcontroller will capture the average and accumulated values (*Total TSCTR* and *Pad Average*) of the selected pad and transmit them to U-SPY which will then display the values on the oscilloscope. The signals displayed in each mode are as follows (**Table 6**):

Table 6 Signals Displayed for Left/Right Button

	Signal1	Signal2
Description	Pad Total_TSCTR * $2^{DIVISORN}$	Pad Average
Colour	Green	Pink

Status Flags

The format of the transmitted data for the status flags is as follows (**Table 7**):

Table 7 Transmit Data Format for Status Flags

	D0	D1	D2
Value (hex)	84	30/31	XX
Description	I.D.	TSLED – 30 ADCLED - 31	Mask

The statuses of the LEDs received by U-SPY are masked before they are displayed as status flags. It is important that the bits of a mask do not overlap with the bits of another mask. This is to ensure that status flags are not falsely turned on. The masks used are as follows (**Table 8**):

Table 8 LED masks for Status Flags

LED Number	1	2	3	4	5	6	7	8
Mask (hex)	01	02	04	08	10	20	40	80

6 Schematics and Layout

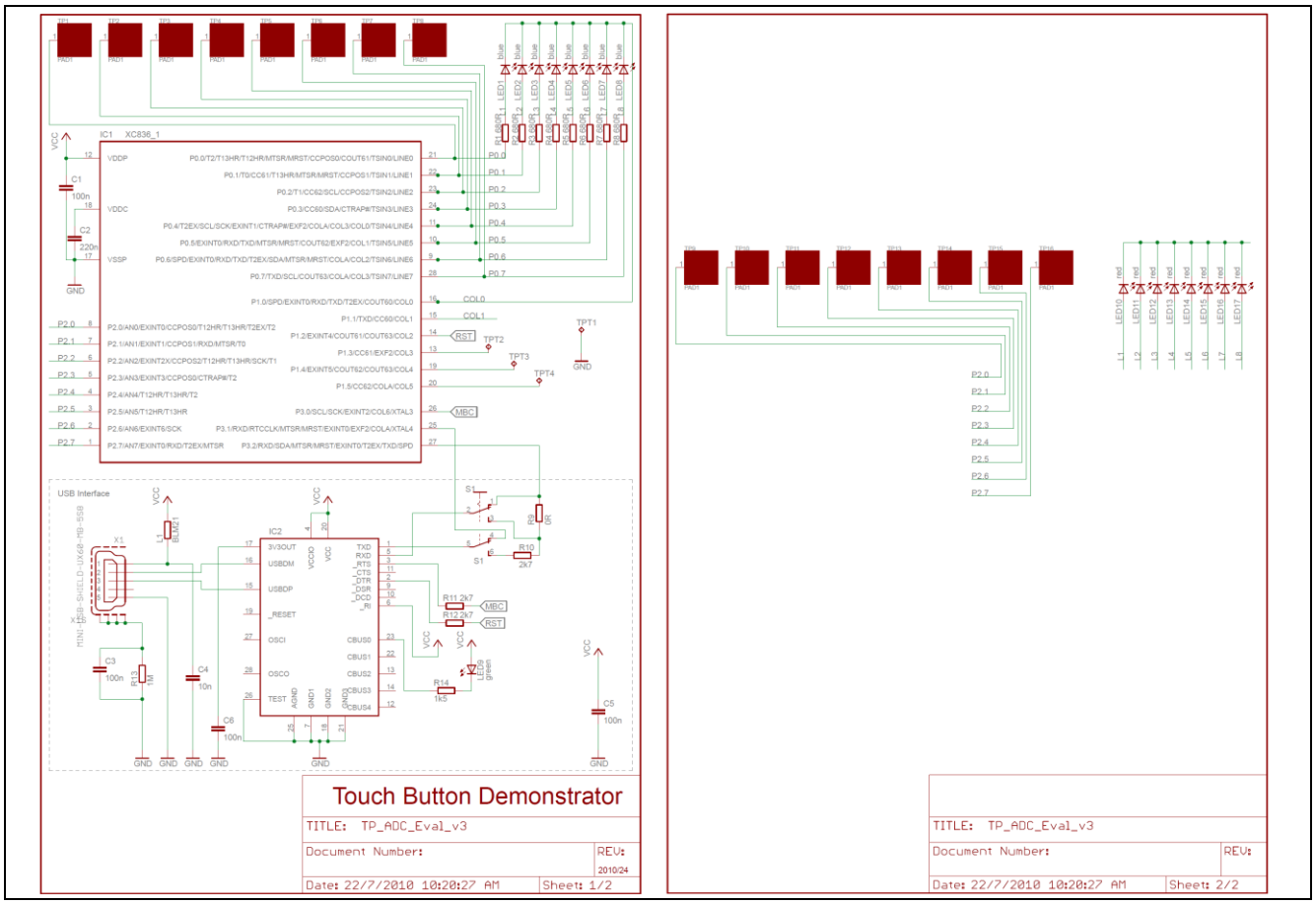

Figure 17 Schematics

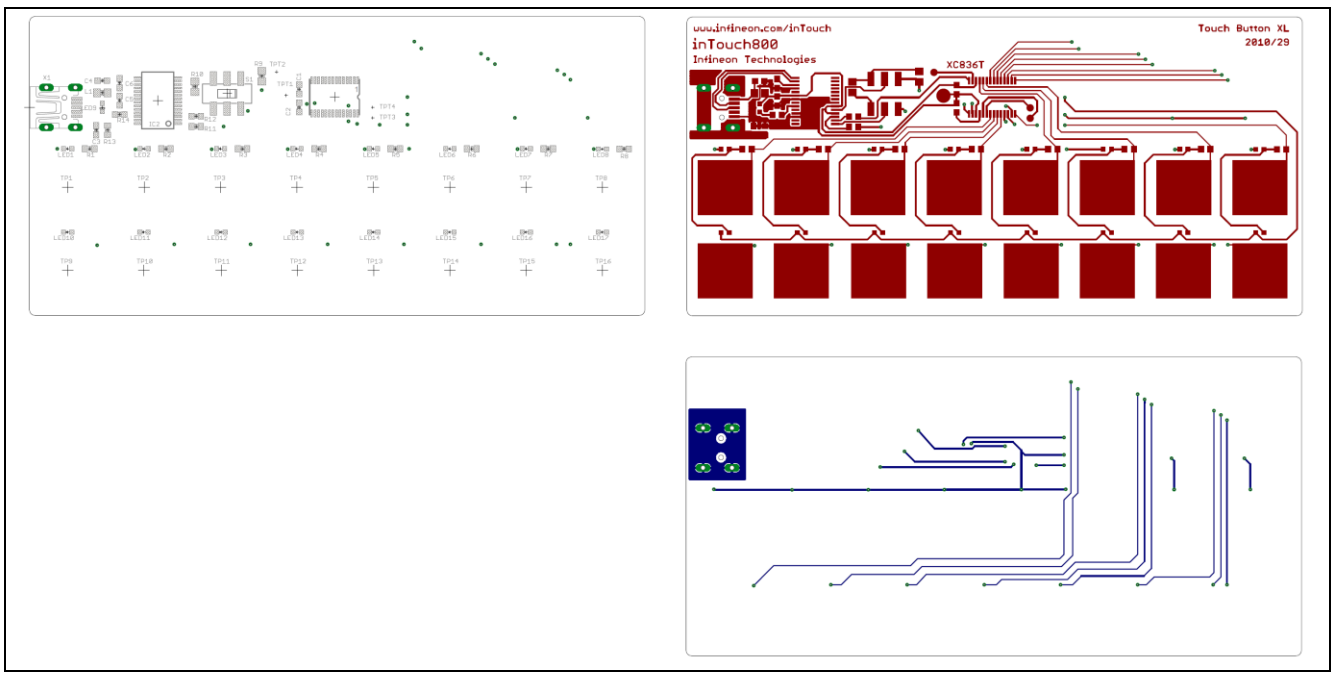

Figure 18 Components and Layouts

7 References

User's Manual – XC83x; 8-Bit Single-Chip Microcontroller

Data Sheet – XC835/836; 8-Bit Single-Chip Microcontroller

Application Note – *AP08100* – Configuration for Capacitive Touch-Sense Application

Application Note – *AP08108* – Programming the BMI value in the XC82x and XC83x products

Application Note – *AP08115* – Design Guideline for Capacitive Touch-Sensing Application

Application Note – *AP08121* – Infrared Remote Controller with Capacitive Touch Interface

Link to *XC83x-Series* – www.infineon.com/xc83x

Link to *Solutions for advanced touch control* – www.infineon.com/intouch

w w w . i n f i n e o n . c o m

Published by Infineon Technologies AG