XC83x

AP08130

inTouch Application Kit - LED Matrix Display

Application Note

V1.0, 2012-02

Microcontrollers

Edition 2012-02

Published by
Infineon Technologies AG
81726 Munich, Germany

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

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: V1.0 2012-02

Previous Version(s):

Page	Subjects (major changes since last revision)
–	

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

In today's Human-Machine Interface (HMI) designs, capacitive touch technology is now often more widely used than traditional mechanical buttons. Capacitive touch technology is the more popular choice because it brings flexibility, a high-level of customization, and a significant reduction in overall system cost.

HMI designs often make user of indicator LEDs as a form of feedback for touch. Infineon's XC83x microcontrollers are capable of driving up to 8 LED columns, with up to 8 LEDs per column. By using shared touch input pins and LED output pins, further cost reductions are possible.

The *inTouch Application Kit* aims to showcase the LED driving capability through a 5 by 7 LED Matrix. Application example code is provided, which can serve as a platform for developing new projects.

The *inTouch Application Kit* comprises of a mother board, supplied as a USB stick, and a number of daughter boards. **Figure 1** shows the USB stick with the LED Matrix daughter board.

This application note describing the LED Matrix daughter board, aims to highlight the ease of implementing a design with Infineon's XC82x and XC83x microcontrollers. Topics covered include column and individual LED dimming behavior.

Figure 1 inTouch Application Kit (USB Stick and LED Matrix board)

2 Hardware and Program Flow

This section describes the hardware used and the connections involved.

2.1 Hardware

Infineon's XC836MT 2FRI (**Figure 2**) is used in this application. The XC836MT is embedded in the *inTouch Application Kit*'s USB stick. For more details regarding the USB stick, please refer to *AP08126: Infineon Touch Solutions - inTouch Application Kit*.

Figure 2 Infineon's XC836MT 2FRI

The *inTouch LED Matrix* board (**Figure 3**) is available as a plug-in daughter board which is part of the *inTouch Application Kit*.

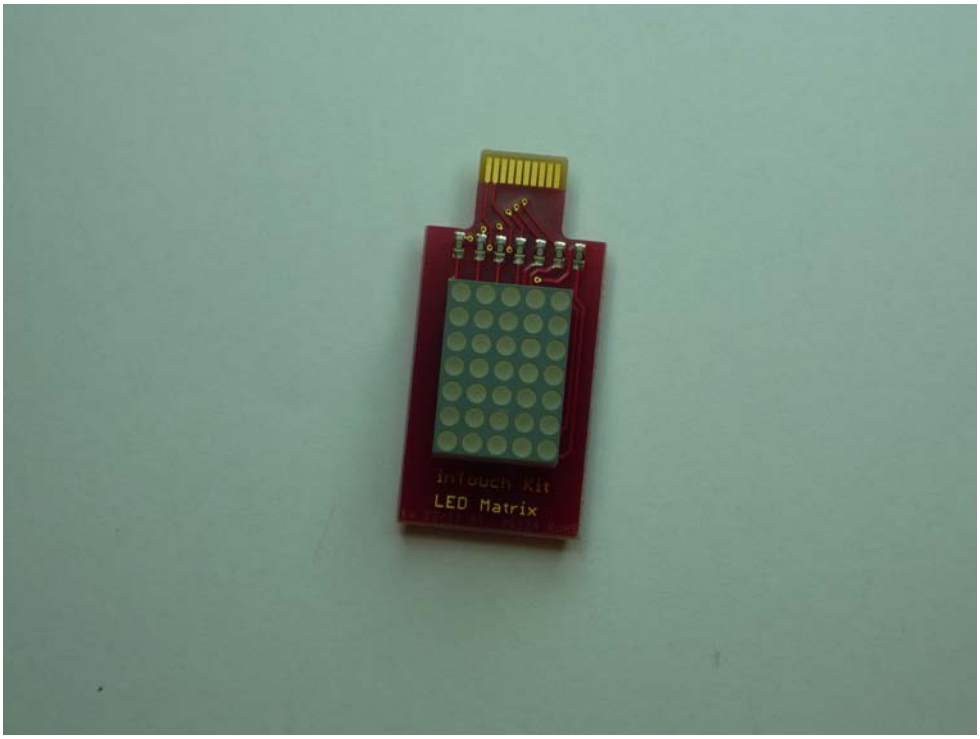

Figure 3 LED Matrix daughter board

The *inTouch LED Matrix* board is a standard PCB incorporating Kingbright's 18mm single color dot matrix display. The column and row pins of the dot matrix display are connected to 5 LEDTS column pins and 7 LEDTS line pins of the XC836 respectively. The schematics are available in the **Appendix - Schematics and Layout**.

2.2 Program Flow

The *inTouch LED Matrix* board comes with 4 program versions for demonstration and tutorial purposes:

- Demonstration
  - **inTouch_LEDMatrix_PCDemo**: provides an intensive demonstration of the capabilities of the LED driving functionality in the XC836 microcontroller with an option to control the display via UART interface
  - **inTouch_LEDMatrix_StandAloneDemo**: provides a demonstration which runs at power-on without the need for a PC interface
- Tutorial
  - **inTouch_LEDMatrix_ColumnControl**: familiarizes the user with LED column brightness control
  - **inTouch_LEDMatrix_IndividualControl**: familiarizes the user with LED individual brightness control

For inTouch_LEDMatrix_ColumnControl and inTouch_LEDMatrix_IndividualControl, the demonstration mode is the default mode, and can also be activated by clicking on the "Demo Mode" button in the attached U-SPY settings files.

inTouch_LEDMatrix_PCDemo

This demonstration code aims to show some display patterns or animations that can be accomplished by controlling the LED column brightness. Using a simple RS232 monitoring software such as Hyperterminal, the user is also able to key in keyboard characters which will then be reflected on the LED Matrix display.

Figure 4 gives an overview of the interrupts that make up the program.

Three important components of the demonstration code are:

1. Timer 2 overflow interrupt (**Figure 5**)
   a) Timer 2 will run each time a new user input is received and an overflow interrupt will be invoked every 0.1s. The interrupt is mainly used to perform the display patterns or animations for the entrance and exit of characters displayed on the LEDs.
2. LEDTS Time slice interrupt (**Figure 6**)
   a) The setting up of the LED display values and the actual writing of LDLINE_VALUE[x] registers occur here.
3. UART interrupt (**Figure 7**)
   a) The UART interrupt is invoked each time the user keys in a new input. The input is then stored and Timer 2 is switched on.

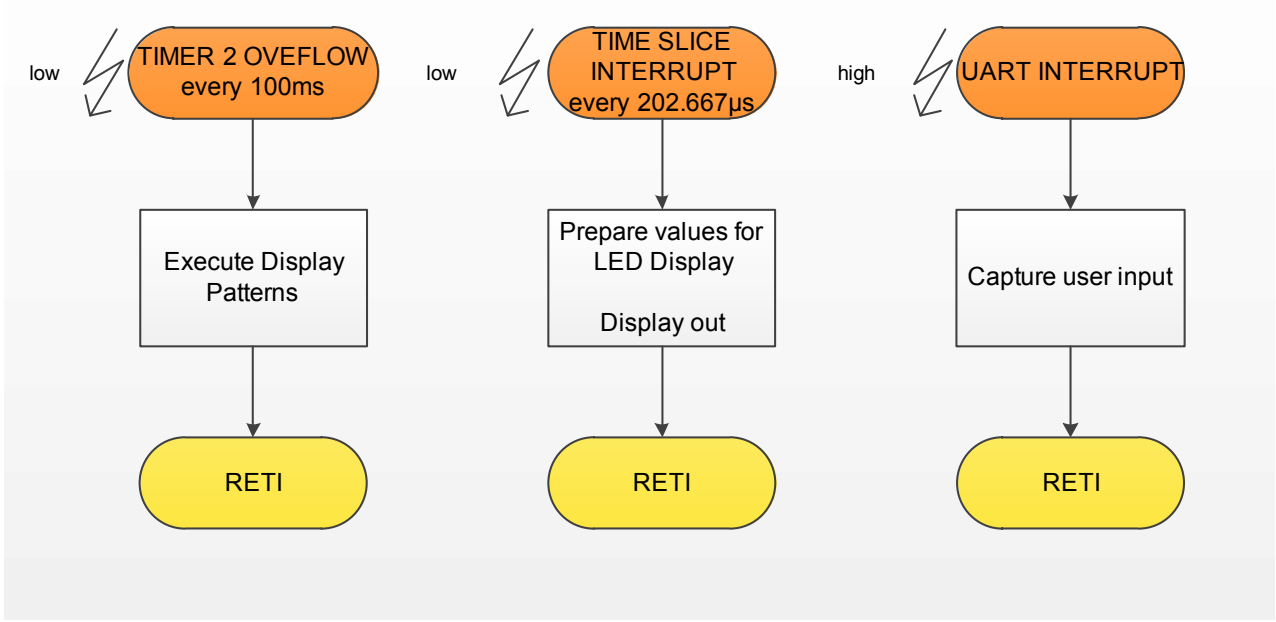

Figure 4 Overview of interrupts in inTouch_LEDMatrix_PCDemo

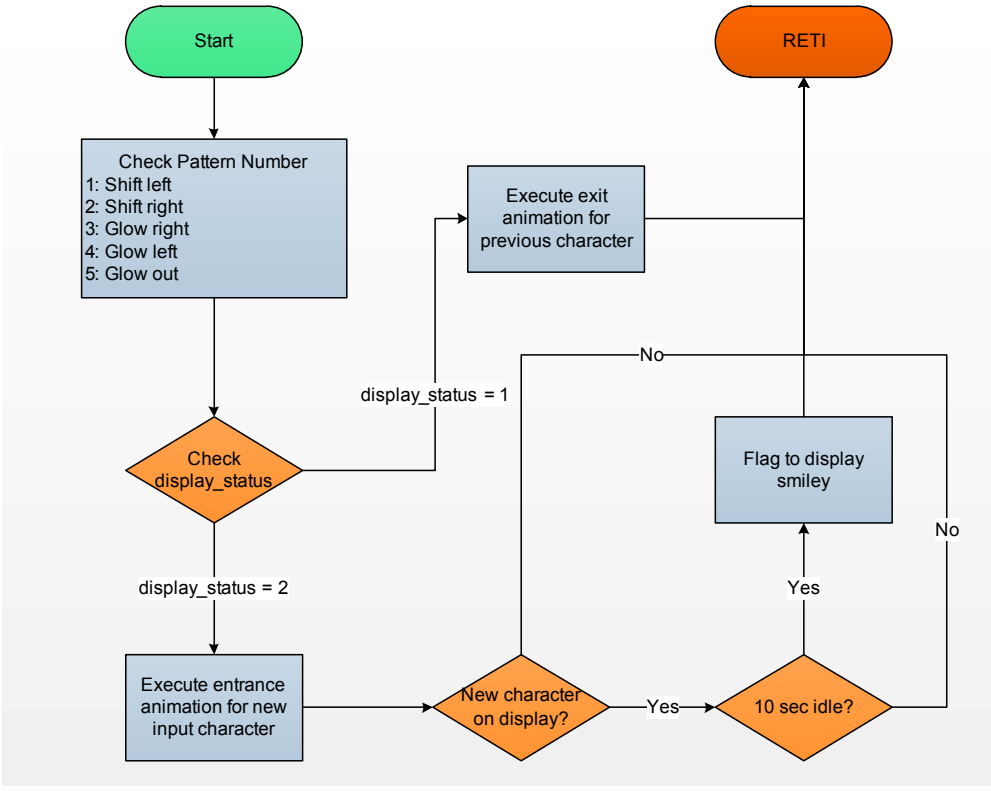

Figure 5 Timer 2 Overflow Interrupt Service Routine in inTouch_LEDMatrix_PCDemo

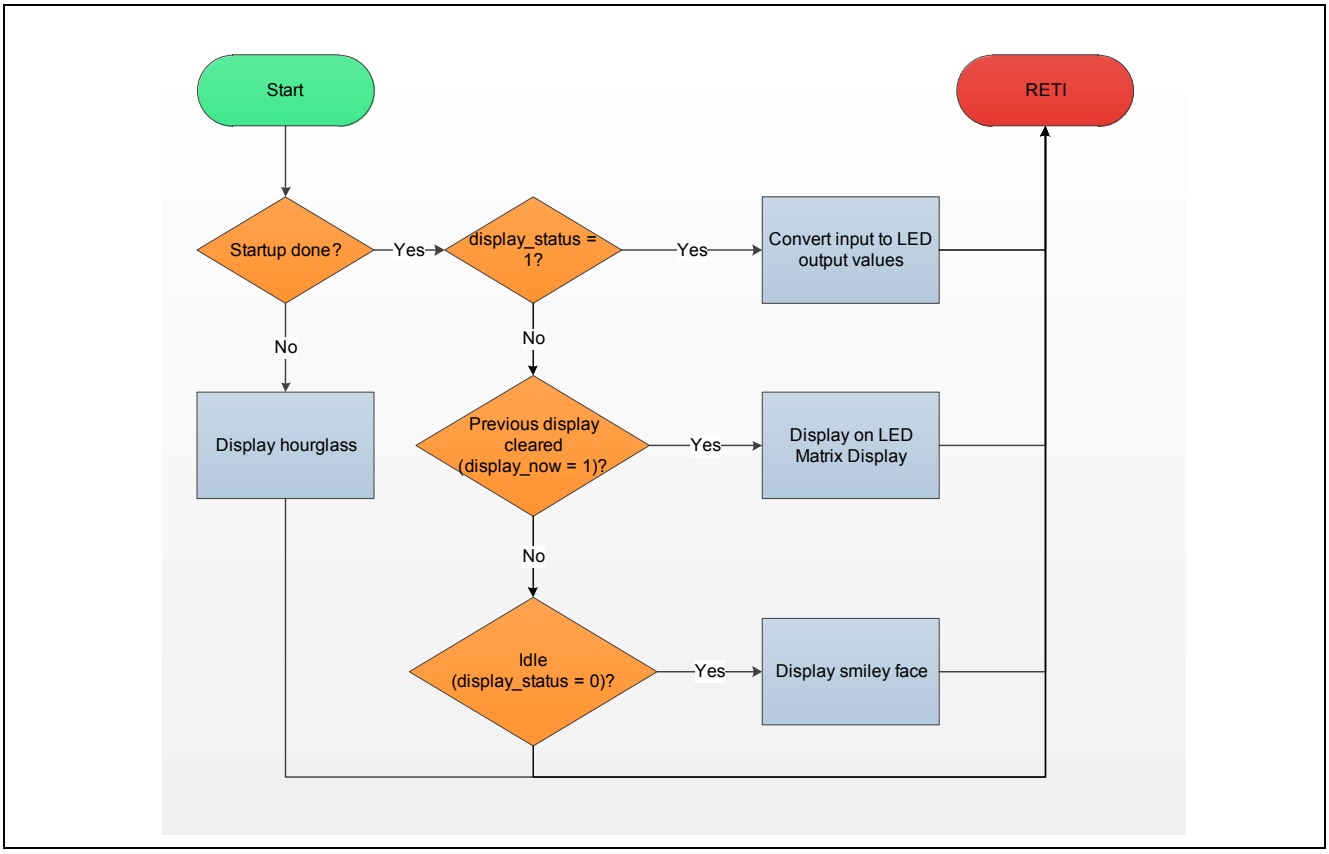

Figure 6 LEDTS Time Slice Interrupt Service Routine in inTouch_LEDMatrix_PCDemo

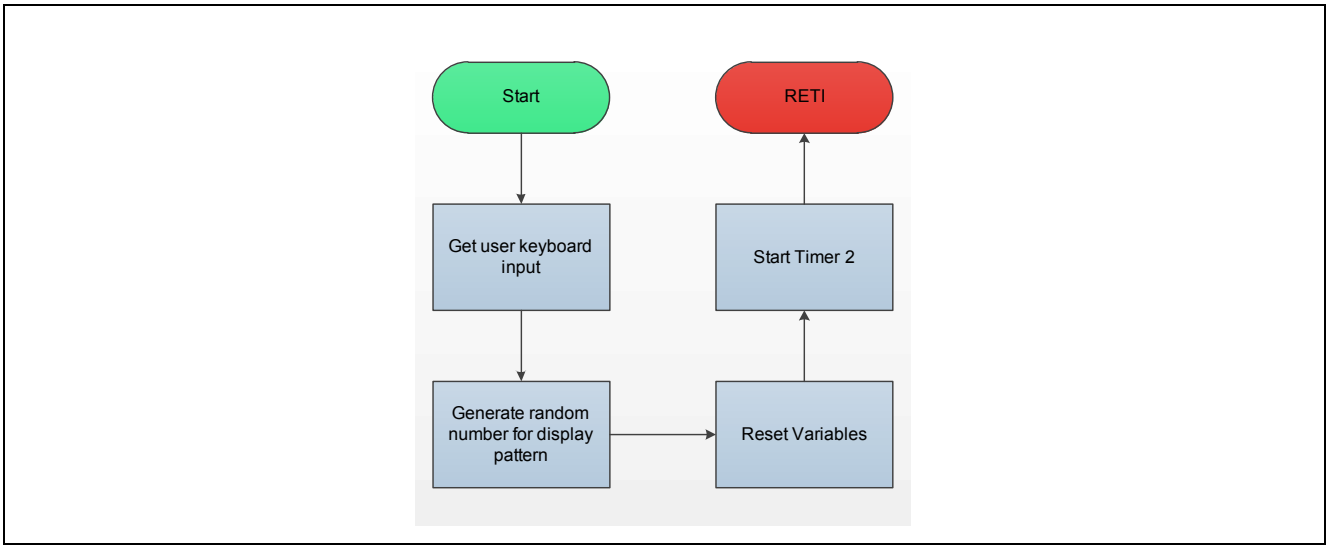

Figure 7 UART Interrupt Service Routine in inTouch_LEDMatrix_PCDemo

inTouch_LEDMatrix_StandAloneDemo

This demonstration code aims to show some display patterns or animations that can be accomplished by controlling the LED column brightness.

Figure 8 gives an overview of the interrupts that make up the program.

Two important components of the demonstration code are:

1. Timer 2 overflow interrupt (**Figure 9**)

a) Timer 2 will run each time a new user input is received and an overflow interrupt will be invoked every 0.1s. This interrupt is mainly used to perform the display patterns or animations for the entrance and exit of characters displayed on the LEDs.

2. LEDTS Time slice interrupt (**Figure 10**)
   a) Used for setting up the LED display values and for the actual writing on the LDLINE_VALUE[x] registers.

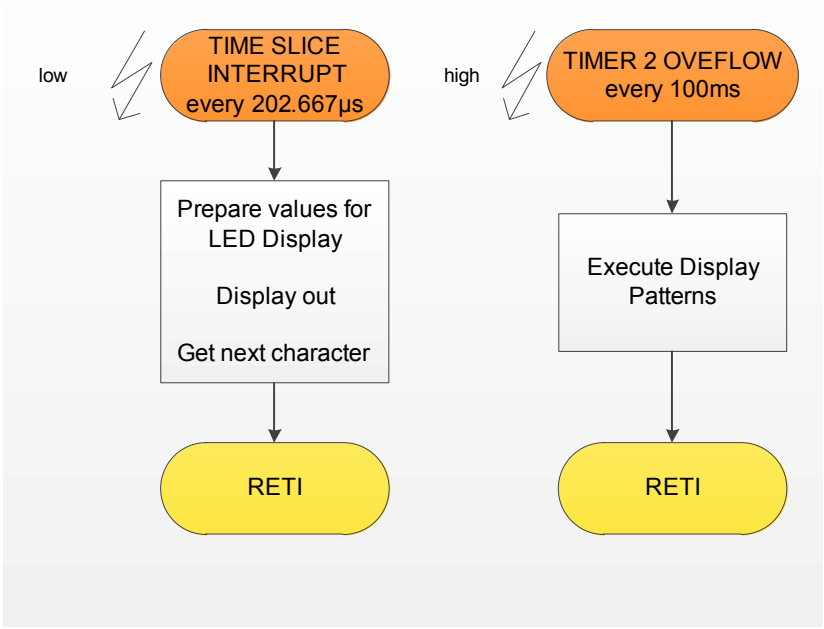

Figure 8 Overview of interrupts in inTouch_LEDMatrix_StandAloneDemo

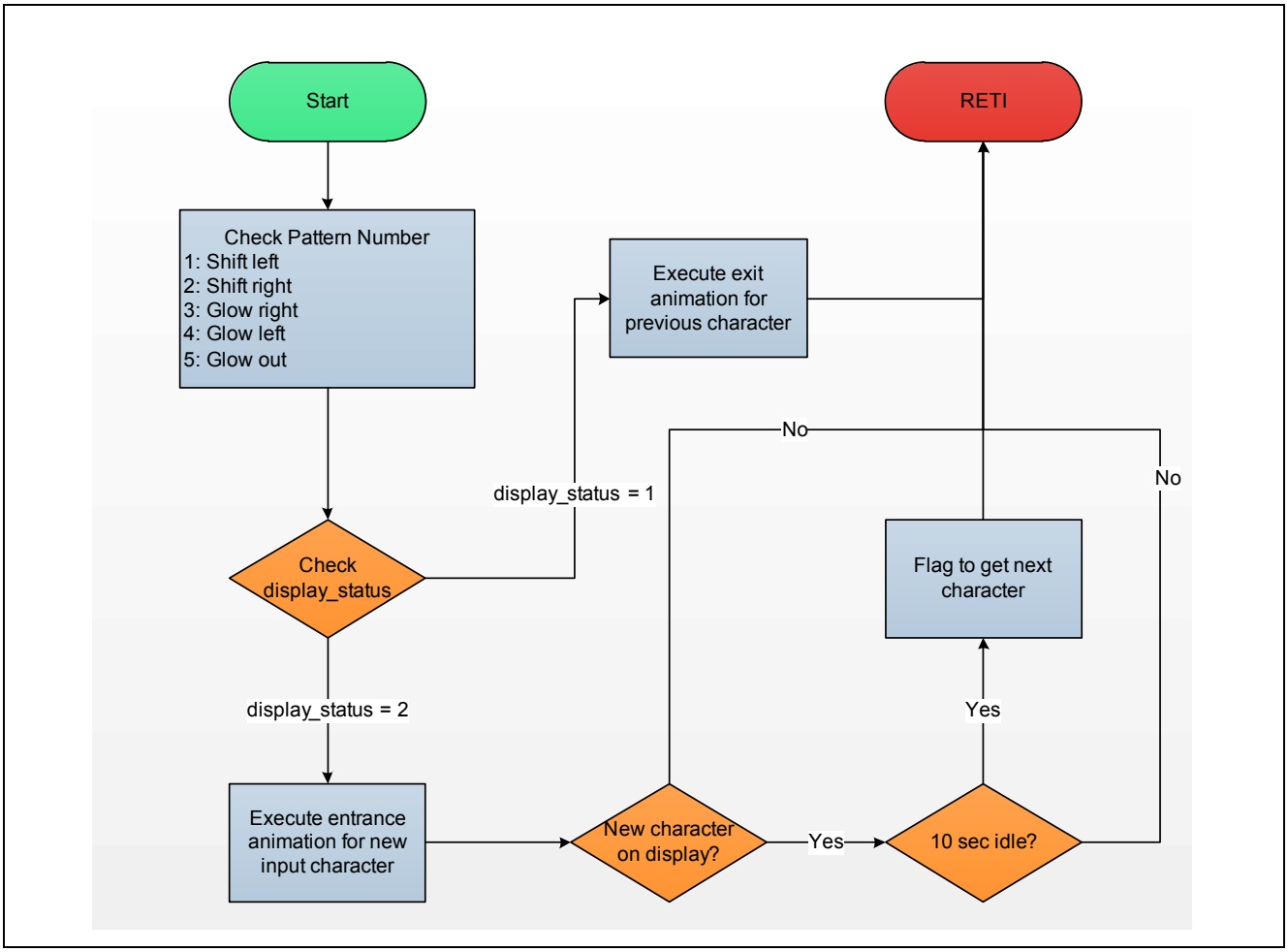

Figure 9 Timer 2 Overflow Interrupt Service Routine in inTouch_LEDMatrix_StandAloneDemo

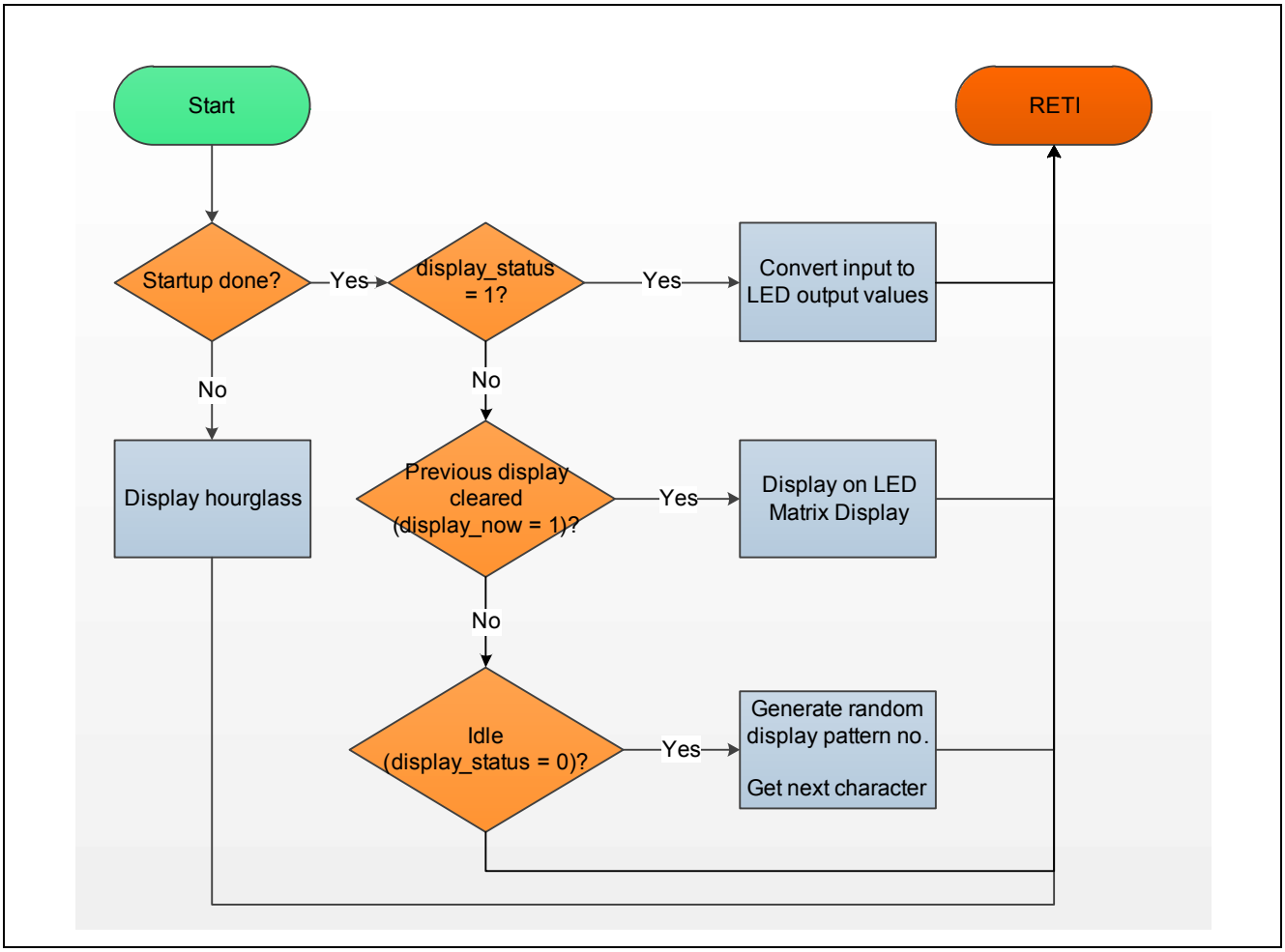

Figure 10 LEDTS Time Slice Interrupt Service Routine in inTouch_LEDMatrix_StandAloneDemo

inTouch_LEDMatrix_ColumnControl

With an objective to demonstrate the ability to control the brightness of the LEDs via column, the demonstration code showcases the toggling of brightness level for 2 colums of LEDs. At any one point of time, all 7 LEDs in each column will have different brightness levels.

For simplicity, the description of the program flow will focus on one column of LEDs.

There are two parts to the demonstration code, namely:

1. Setting CMP_BRIGHTNESS[x] with the individual LED brightness, which will be described in **Figure 19**.
2. Adjusting the brightness of each LEDs.

Both these routines are carried out in the Time Slice Interrupt service routine. **Figure 11** gives an overview of how the two routines are scheduled in the interrupt.

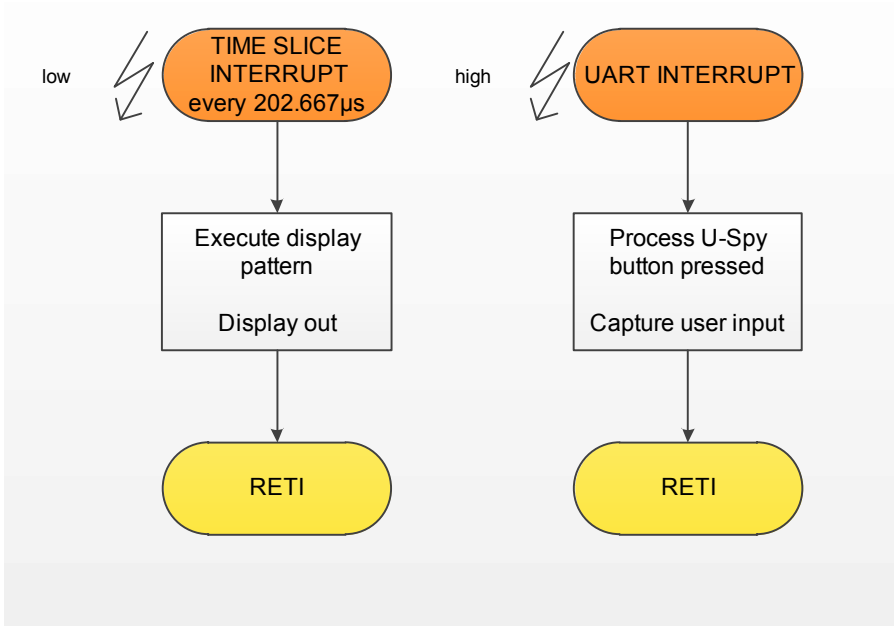

Figure 11 Overview of interrupts in inTouch_LEDMatrix_ColumnControl

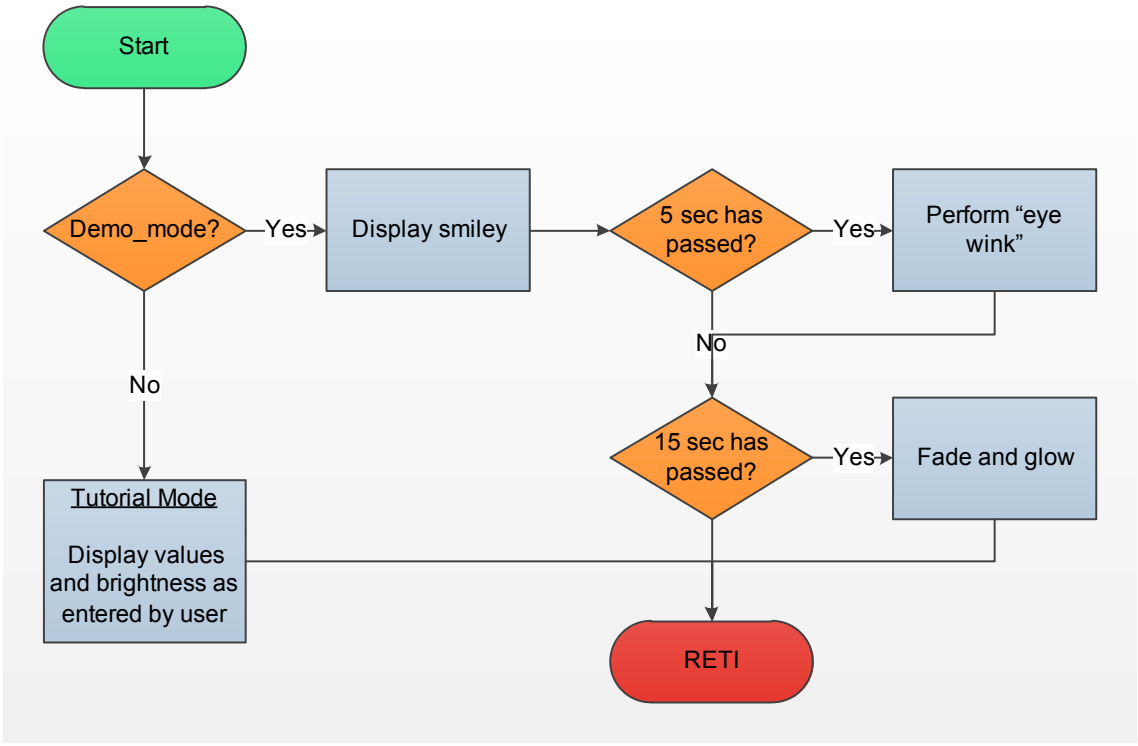

Figure 12 LEDTS Time Slice Interrupt Service Routine in inTouch_LEDMatrix_ColumnControl

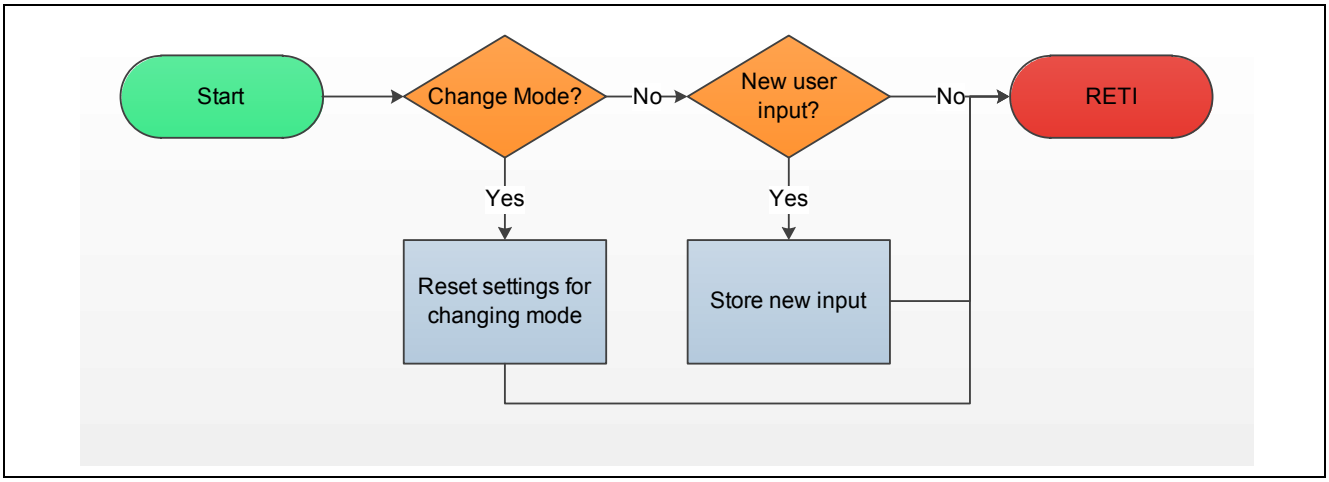

Figure 13 UART Interrupt Service Routine in inTouch_LEDMatrix_ColumnControl

inTouch_LEDMatrix_IndividualControl

With an objective to demonstrate the ability to control the brightness of the LEDs individually, the demonstration code showcases the toggling of brightness level for 2 colums of LEDs. At any one point of time, all 7 LEDs in each column will have different brightness levels.

For simplicity, the description of the program flow will focus on one column of LEDs.

There are two parts to the demonstration code, namely:

1. Setting CMP_BRIGHTNESS[x] with the individual LED brightness, which will be described in **Figure 20**.
2. Adjusting the brightness of each LEDs.

Both these routines are carried out in the Time Slice Interrupt service routine. **Figure 14** gives an overview of how the two routines are scheduled in the interrupt.

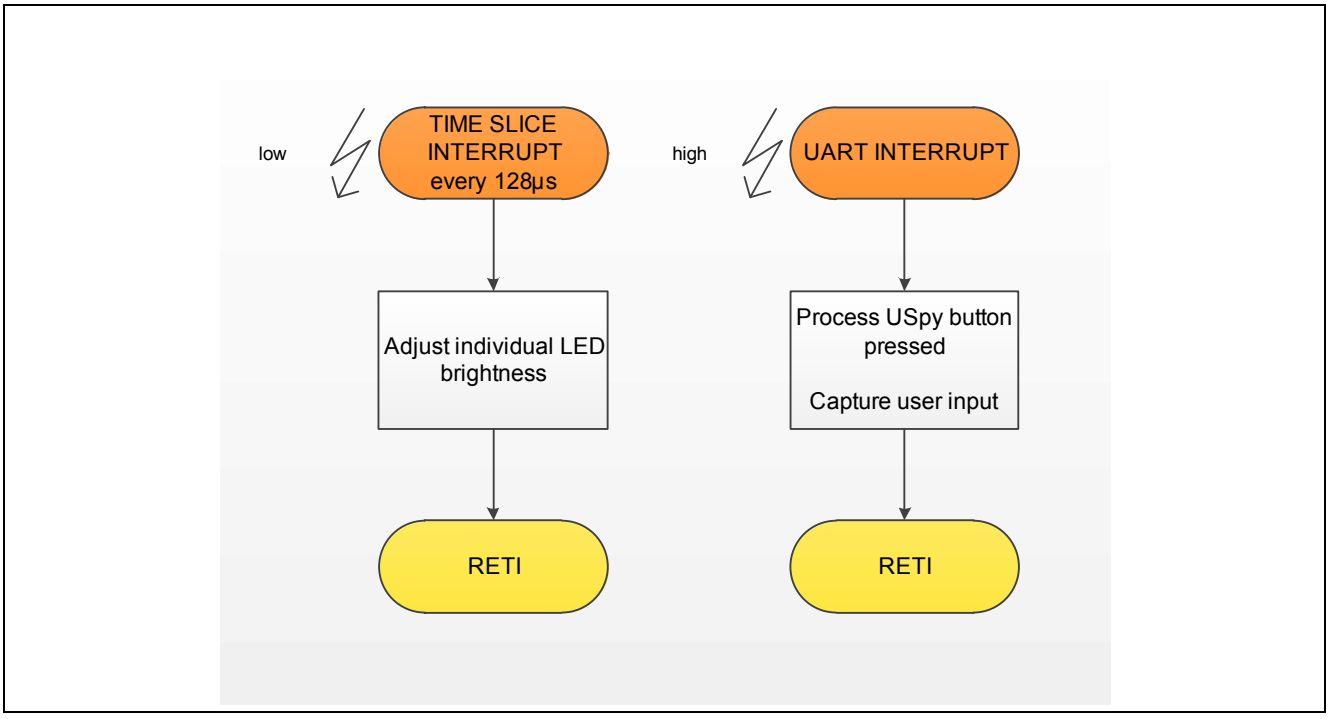

Figure 14 Overview of interrupts in inTouch_LEDMatrix_IndividualControl

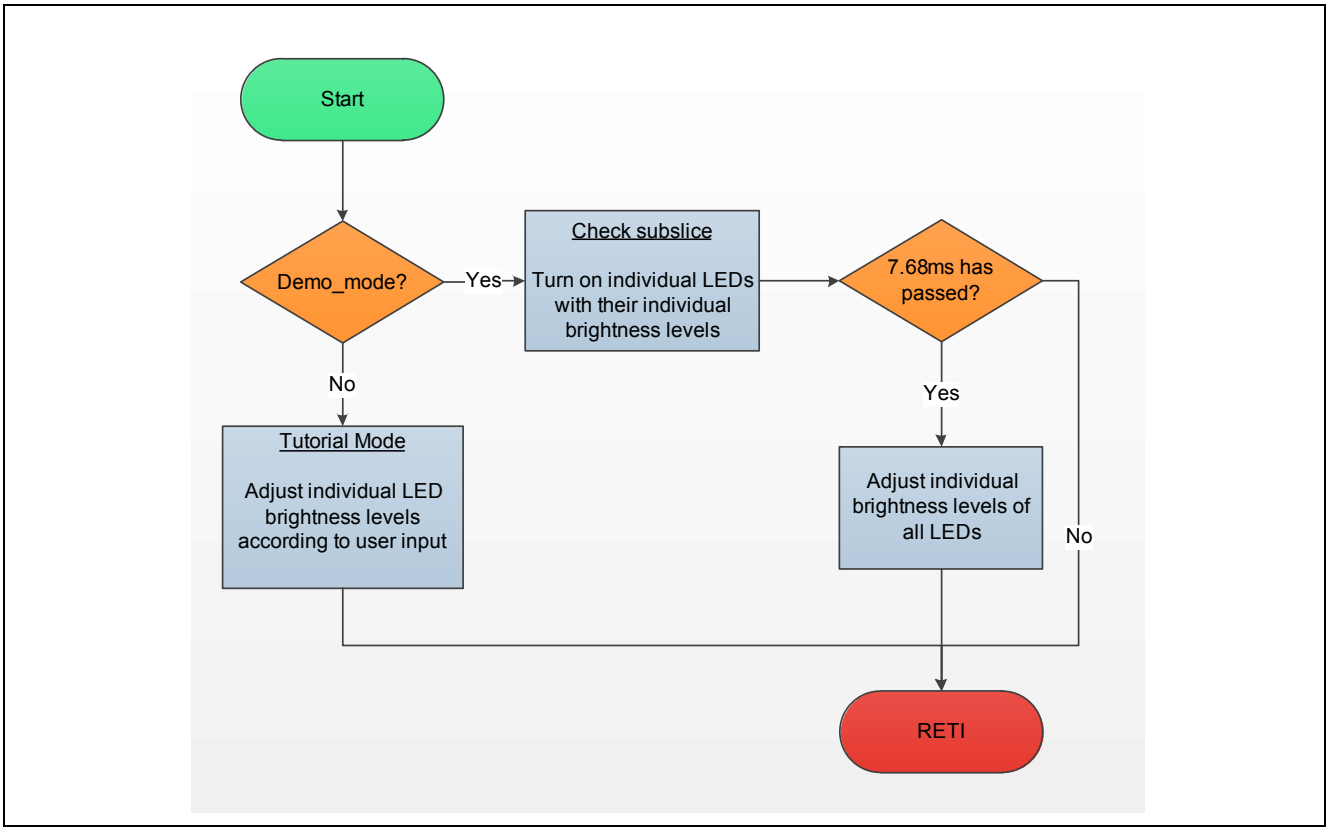

Figure 15 LEDTS Time Slice Interrupt Service Routine in inTouch_LEDMatrix_IndividualControl

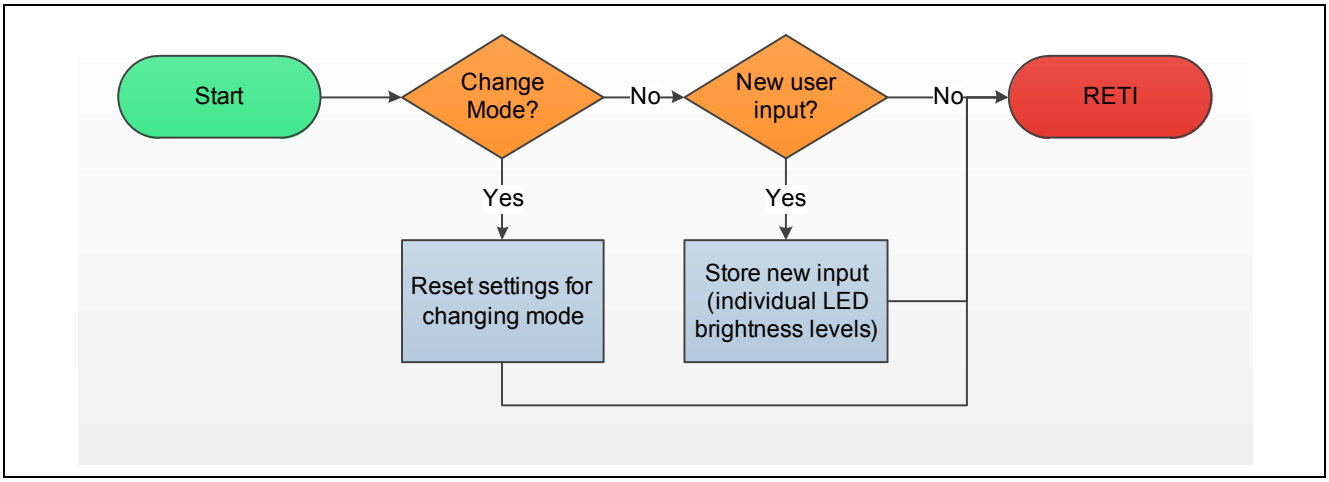

Figure 16 UART Interrupt Service Routine in inTouch_LEDMatrix_IndividualControl

3 LED Matrix Controller and ROM Library

The XC836 microcontroller drives LEDs via an LED Matrix controller (**Figure 17**). The LED Matrix consists of many LEDs arranged in lines and columns. The line pins are shared with touch pads. If the touch-sensing function is also enabled, one of the column pins (COLA) will be used for touch control, so reducing the number of LEDs that can be driven to 56. If the touch-sensing function is not enabled then up to a maximum of 64 LEDs can be driven, as shown in **Figure 17**. A resistor lies in each line path to limit the current.

The LED Matrix controller activates the columns one after another in a round-robin fashion. Each time period, in which one column is enabled, is called a time slice. A compare register for LEDTS-counter is provided so that the duty cycle for column enabling per time slice can be adjusted. This duty cycle essentially corresponds to the LED column's brightness level. The LED column is enabled from the beginning of the time slice until a compare match event is achieved. For 100% duty cycle, the compare value should be set to FF_H. Setting this value to 00_H will cause the LED column to stay at passive level throughout the time slice. The compare register is updated at the beginning of each time slice by automatic shadow transfer.

During each time slice, LEDs connected to the active column can be enabled via the line signals which are automatically synchronized to the column activation. A line register is provided to hold the LED line pattern value. This register is also updated at the beginning of each time slice by automatic shadow transfer.

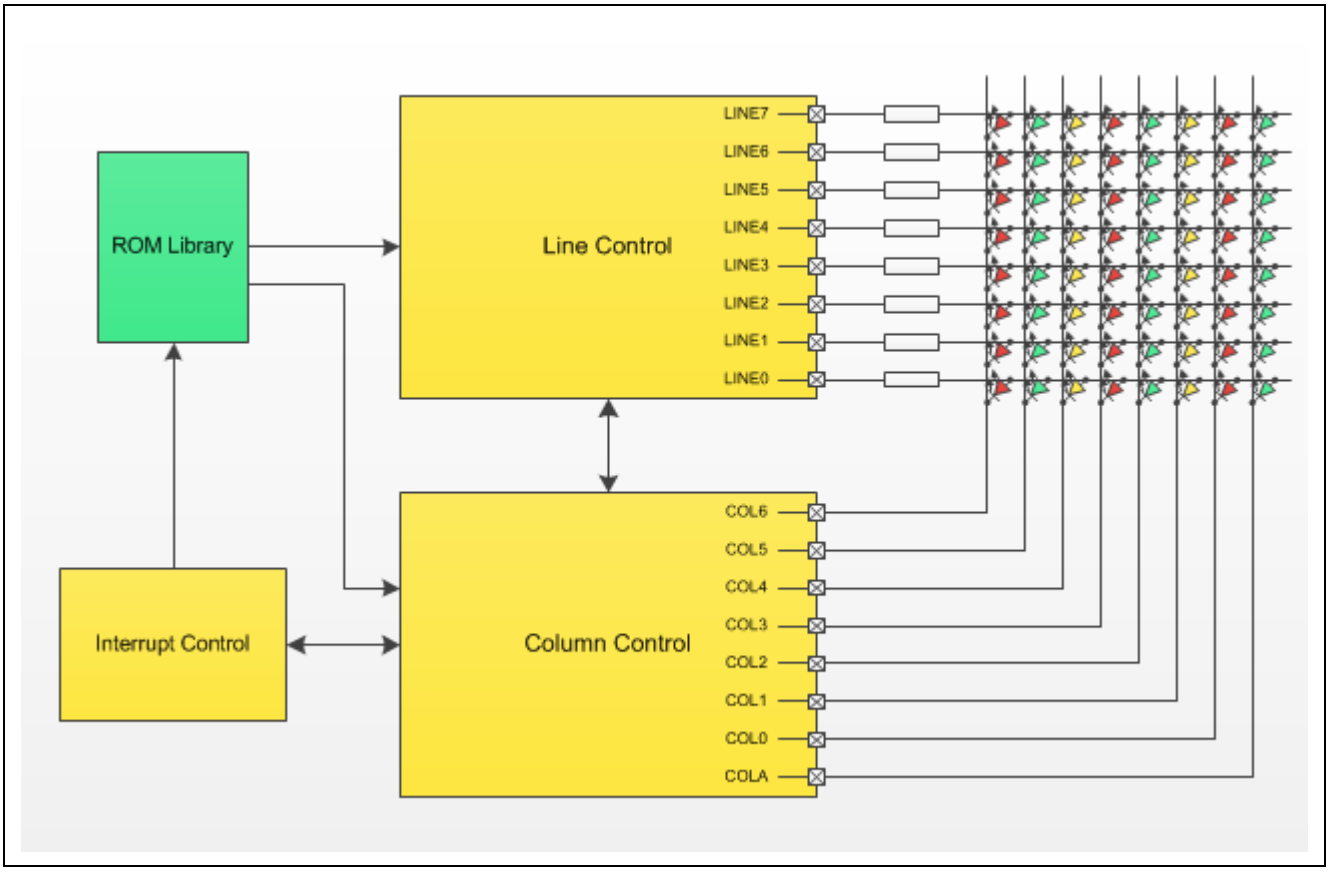

Figure 17 LED Matrix Controller

The ROM Library has a function which takes care of updating the compare and line registers with the correct values. The function, SET_LDLINE_CMP, synchronizes the user's input data for each column and line with the time slice and updates the registers accordingly. This function can be called at every Time Slice or Time Frame Interrupt.

4 Behavior

This section will provide some tips on controlling the LEDs such as switching the LEDs on and off, and adjusting the brightness level. The user can select to control the brightness level of the LEDs either at a column level or at each individual LED level. Both methods of control are described.

4.1 Switching ON/OFF

The LEDs can be switched on or off by programming the LEDTS ROM Library variable array, LDLINE_VALUE[x]. This is easy once the column and line pins for that particular LED are known. The following diagram (**Figure 18**), shows the column and line pin connections to each LED in this application.

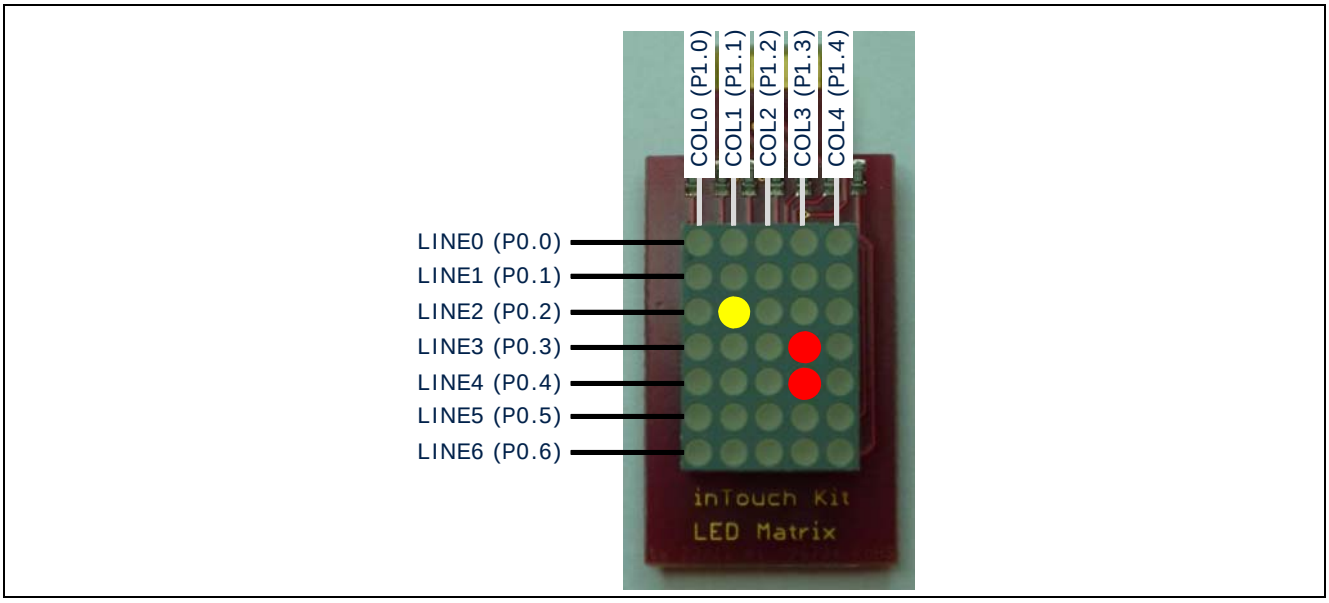

Figure 18 Column and line pin connections to LEDs

The column number will determine the line value array and the line number will determine the bit position for the LED. Here are some examples:

- To switch on the LED marked with colour yellow in **Figure 18**, the user will have to set LDLINE_VALUE[2] = 04_H.
- To switch on the LEDs marked with colour red in **Figure 18**, the user will have to set LDLINE_VALUE[4] = 18_H.

4.2 Controlling Brightness (Column Level)

The brightness level of the LEDs can be controlled on a column-level, meaning all LEDs connected to the same column pin will have the same brightness level. This brightness level depends on the time period within the column time slice, during which the LEDs are turned ON. This “LED ON” time period can be adjusted via LEDTS ROM Library variables CMP_BRIGHTNESS[x][1]. Again, the column number will determine the CMP_BRIGHTNESS number. In **Figure 19**, CMP_BRIGHTNESS[5] is set to FF Hex. Hence, the LEDs on COL4 have the maximum brightness level.

1) CMP_BRIGHTNESS will be called TSCOMPARE if the touch-sensing function is enabled.

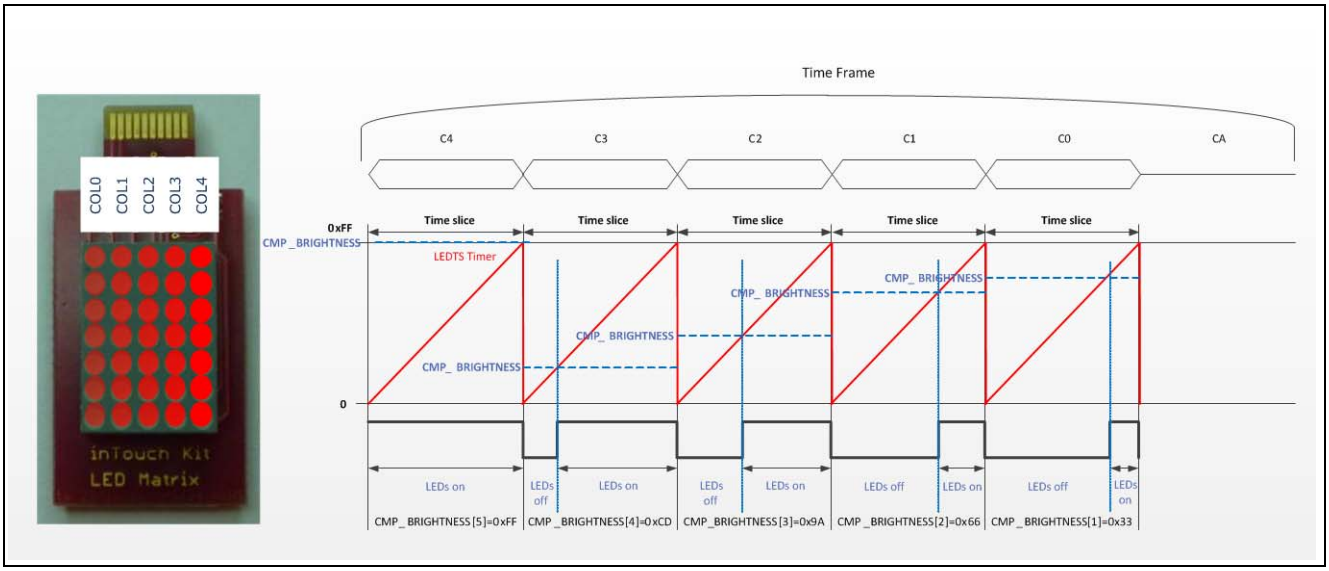

Figure 19 Brightness Control (Column) Example

4.3 Controlling Brightness (Individual LEDs)

Besides column-level control for LED brightness, it is also possible for LEDs connected to the same column pin to have different brightness levels. This can be done by switching on only 1 LED with its desired brightness level in every time slice. The example (**Figure 20**) will make this clearer:

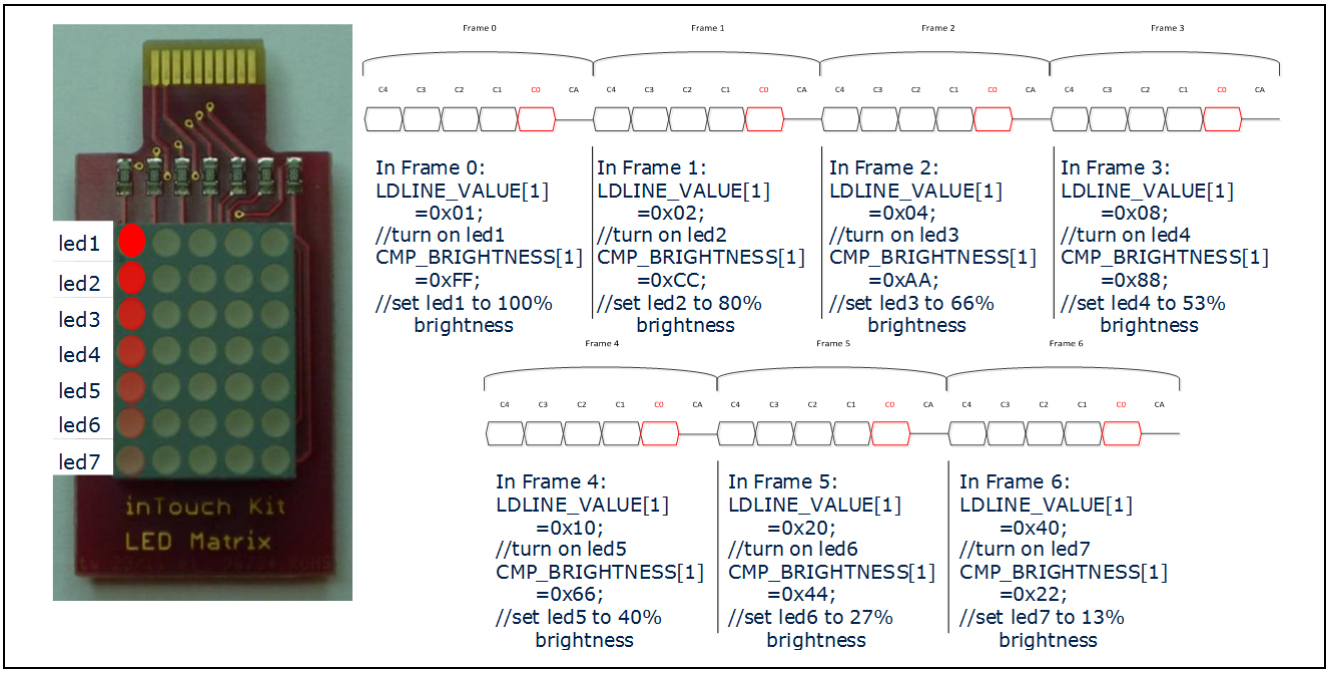

Figure 20 Brightness Control (Individual) Example

Now, lets say that we want to have different brightness levels for all the 7 LEDs on COL0. So, in the COL0 time slice in Time Frame 0, we set LDLINE_VALUE[1] to 1 hex and the CMP_BRIGHTNESS[1] to FF hex. Therefore, we are only switching on the topmost LED with maximum brightness level. In the next COL0 time slice in Time Frame1, we set LDLINE_VALUE[1] to 2 hex and CMP_BRIGHTNESS[1] to CC hex. Therefore, in this time frame, we are only switching on the 2nd LED with 80% brightness level. In the subsequent time frames, the next LED is switched on with its individual brightness level and this cycle is repeated.

This method works due to the high operation frequency. So, it is not evident to our human eyes that we are doing the cycling. However, one point to note here is that the maximum brightness level is lowered due to relative brightness. This can be compensated by choosing line resistors with a lower value.

5 U-SPY

For the *inTouch LED Matrix Board*, two settings files have been configured:

- **inTouch_LEDMatrix_Column_Control.ini**
- **inTouch_LEDMatrix_Individual_Control.ini**

5.1 inTouch_LEDMatrix_Column_Control.ini

This settings file (**Figure 21**) is customized to allow the user to control the brightness of the LEDs at the column level.

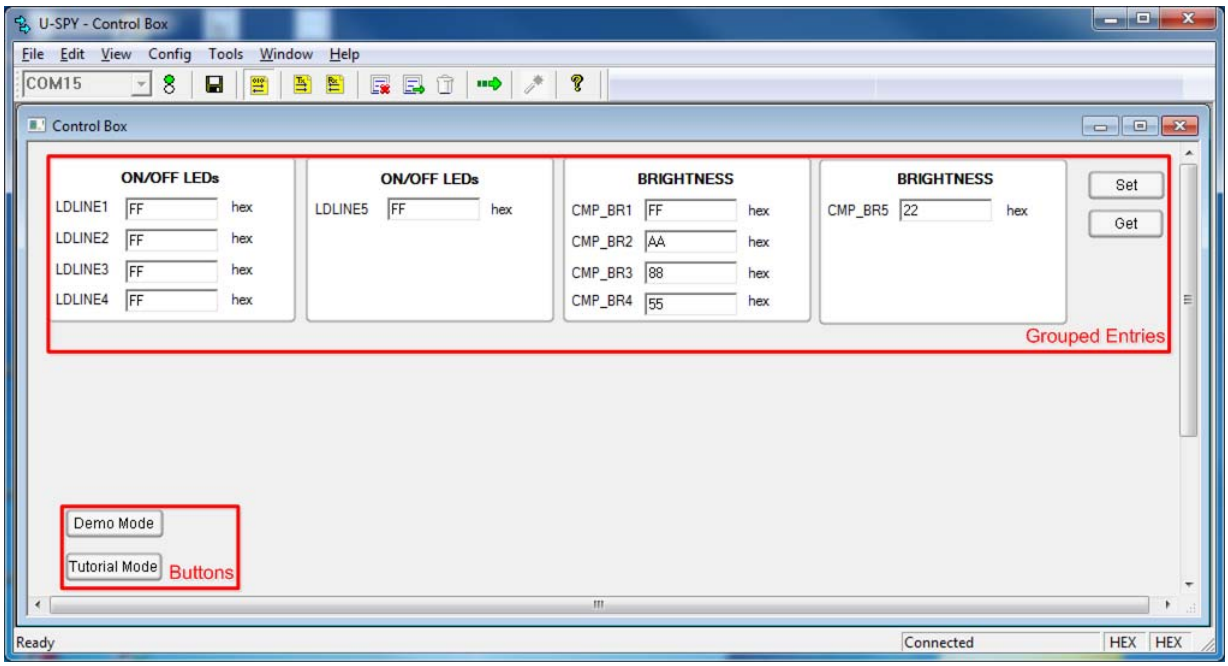

Figure 21 inTouch_LEDMatrix_Column_Control.ini User Interface

Grouped Entries

The grouped entries are used as the interface for the user to change the value of LEDTS ROM Library parameters, LDLINE_VALUE[x] and CMP_BRIGHTNESS[x], without directly accessing the code. The first two boxes of grouped entries are for turning on and off each individual LEDs via bit position values. The other two boxes of grouped entries are for controlling the brightness of the LEDs at the column level as described in **Chapter 4.2**.

The “Set” button is used to change the stored values of LDLINE_VALUE[x] and CMP_BRIGHTNESS[x] and the “Get” button is used to retrieve their stored values.

A data group consisting of 3 bytes is transmitted between U-SPY and the XC836 microcontroller, in the following format (**Table 1**):

Table 1 Grouped Entries Data Format

	DataByte0(D0)	D1	D2
Value (hex)	84	Assigned no.	XX
Description	Data Group I.D.	Box Position I.D.	Data

The data group transmitted/received is then matched before it is processed.

Buttons

The buttons are used to switch the modes. Data is transmitted from U-SPY to the XC836 microcontroller in the following format (**Table 2**):

Table 2 Buttons Data Format

	D0	D1
Value (hex)	08	XX
Description	Data group I.D.	Mode no.

5.2 inTouch_LEDMatrix_Individual_Control.ini

This settings file (**Figure 22**) is customized to allow the user to control the brightness of each one of the 35 LEDs individually.

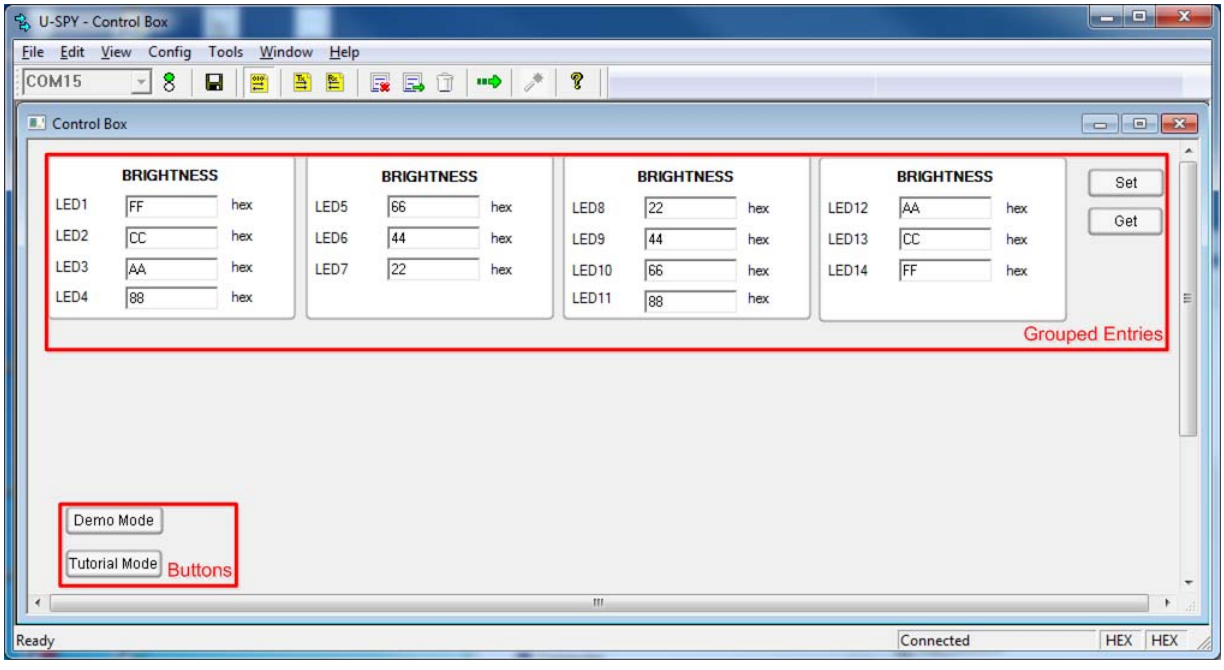

Figure 22 inTouch_LEDMatrix_Individual_Control.ini User Interface

Grouped Entries

The grouped entries are used as the interface for the user to set the brightness level of each individual LEDs without directly accessing the code. These values are then used to control the brightness of each LED in the manner as described in **Chapter 4.3**.

The “Set” button is used to change the stored values for the brightness level of each LED and the “Get” button is used to retrieve their stored values.

A data group consisting of 3 bytes is transmitted between U-SPY and the XC836 microcontroller, in the following format (**Table 3**):

Table 3 inTouch_LEDMatrix_Individual_Control.ini Transmitted Data Format

	DataByte0(D0)	**D1**	**D2**
Value (hex)	84	Assigned no.	XX
Description	Data Group I.D.	Box Position I.D.	Data

The data group transmitted/received is then matched before it is processed.

Buttons

The buttons are used to switch between Demo and Tutorial Mode. Data is transmitted from U-SPY to the XC836 microcontroller in the following format (**Table 4**):

Table 4 Buttons Data Format

	D0	**D1**
Value (hex)	08	XX
Description	Data group I.D.	Mode no.

Appendix - Schematics and Layout

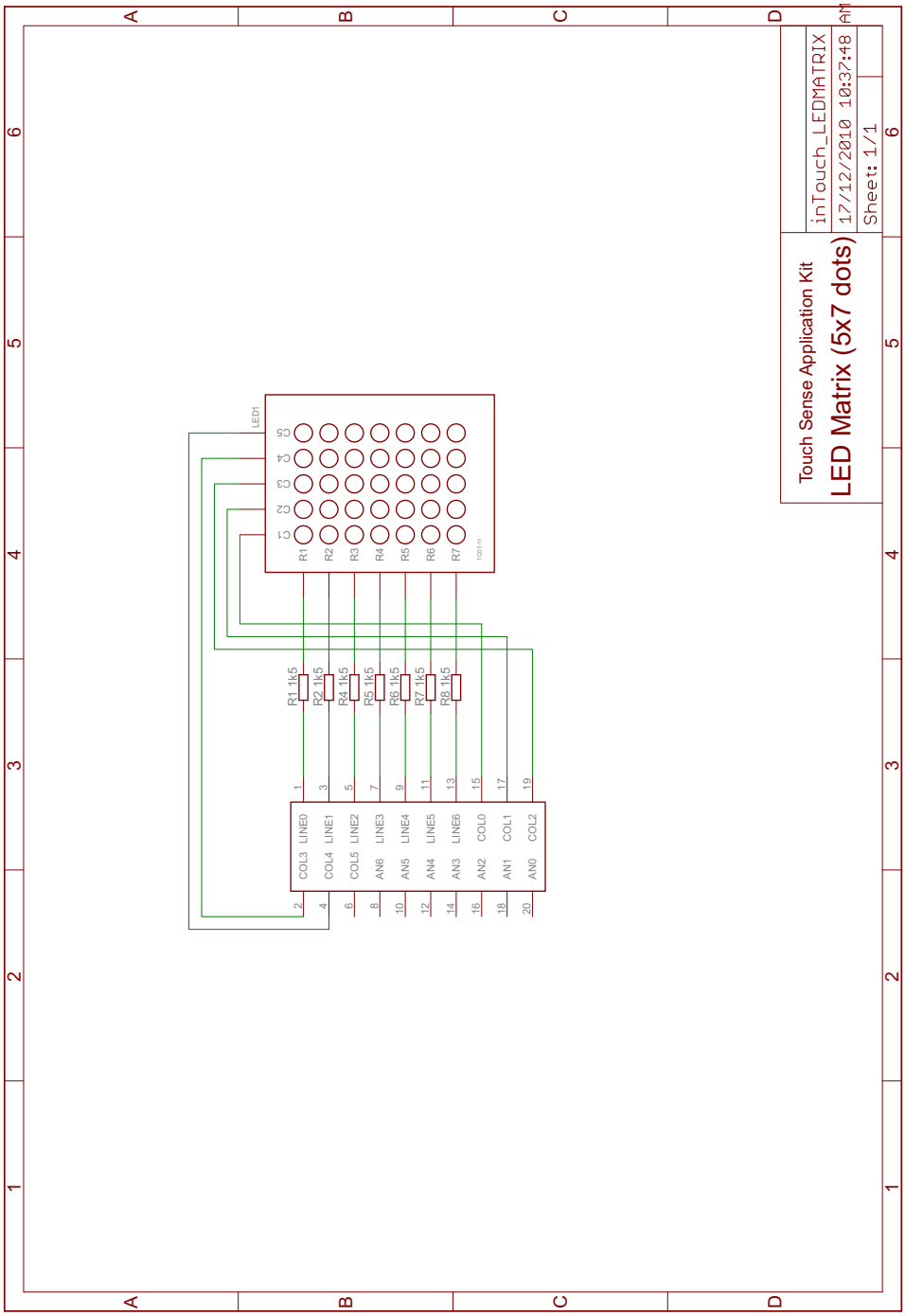

Figure 23 *inTouch LED Matrix* Board Schematics

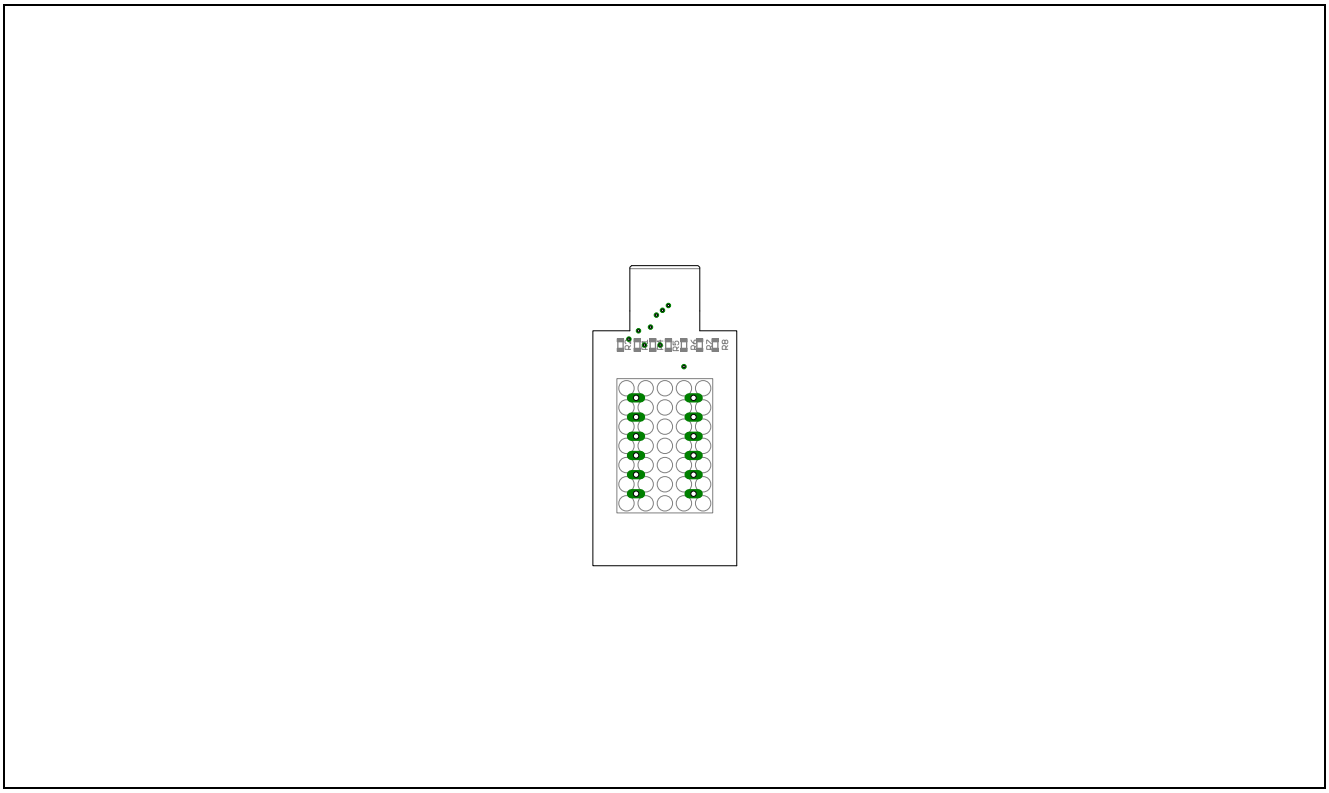

Figure 24 *inTouch LED Matrix* Board Component Top Layout

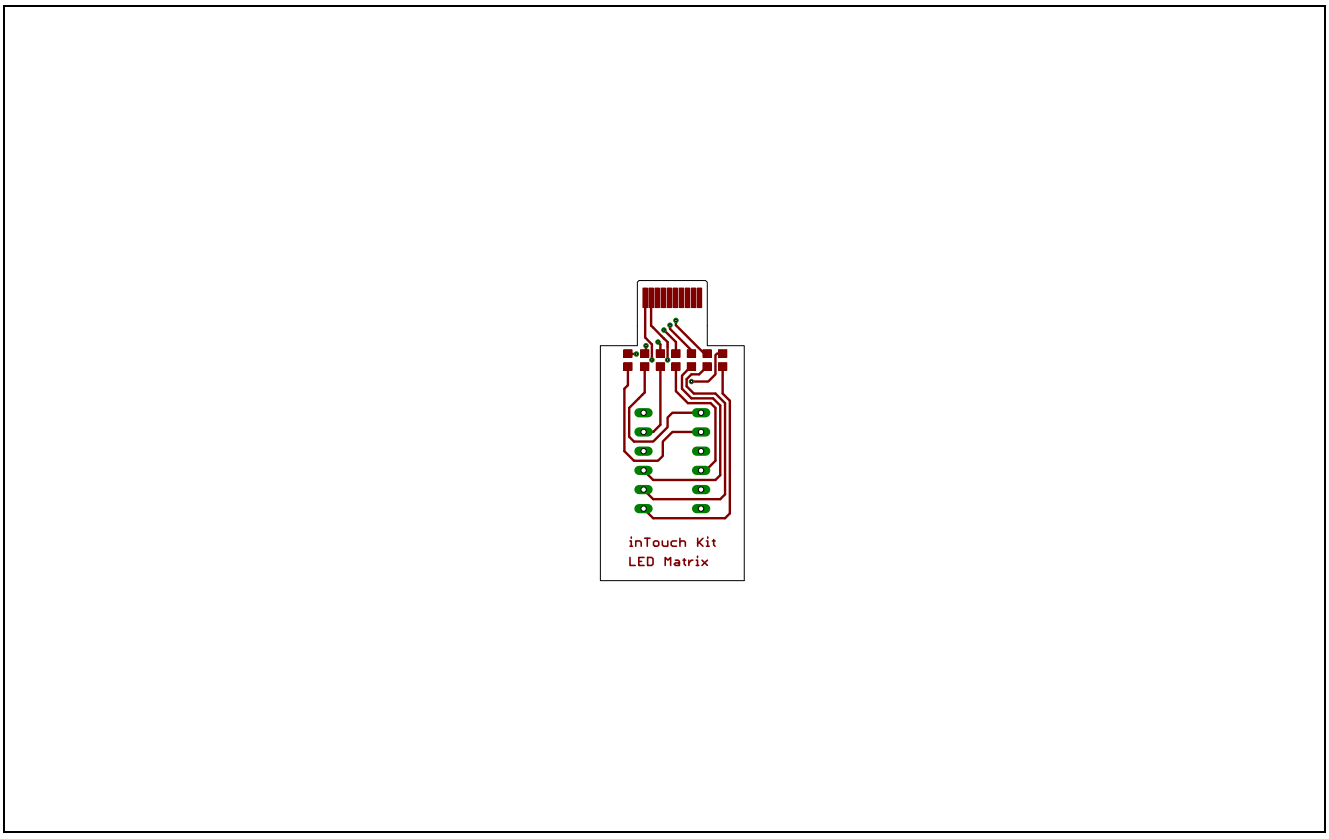

Figure 25 *inTouch LED Matrix* Board Top Layout

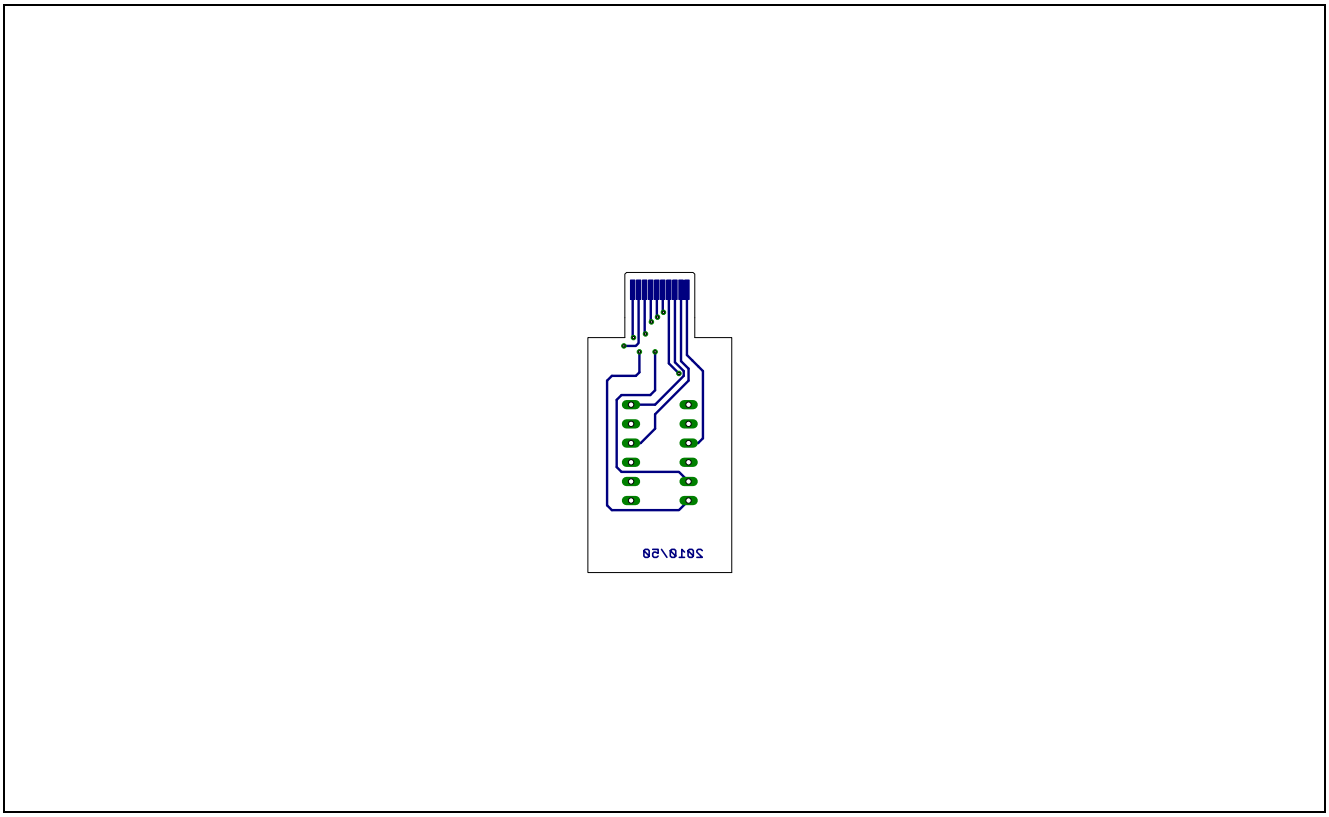

Figure 26 *inTouch LED Matrix* Board Bottom Layout

References

The list below provides resources that may be useful to the user.

1. User's Manual - *XC83x; 8-Bit Single-Chip Microcontroller*
2. Application Note - *AP08100 - Configuration for Capacitive Touch-Sense Application*
3. Application Note - *AP08110 - Design Guidelines for XC82x and XC83x Board Layout*
4. Application Note - *AP08113 - Capacitive-Touch Color Wheel Implementation*
5. Application Note - *AP08115 - Design Guidelines for Capacitive Touch-Sensing Application*
6. Application Note - *AP08121 - Infrared Remote Controller with Capacitive Touch Interface*
7. Application Note - *AP08122 - 16-Button Capacitive Touch Interface with XC836T*
8. Application Note - *AP08124 - XC82/83x Design Guidelines for Electrical Fast Transient (EFT) Protection in Touch-Sense Applications*
9. Application Note - *AP08126 - Infineon Touch Solutions - inTouch Application Kit*
10. Application Note - *AP08127 - inTouch Application Kit - Buttons*
11. Application Note - *AP08128 - inTouch Application Kit - Touch Wheel*
12. Application Note - *AP08129 - inTouch Application Kit - Touch Sliders*
13. Link to *XC83x-Series* - **www.infineon.com/xc83x**
14. 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