XC82x/XC83x

DMX512 Transmitting Device with XC836

AP08132

Application Note

V1.0, 2012-10

Microcontrollers

Edition 2012-10

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.

XC82x/XC83x
Revision History: V1.0 2012-10

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

DMX512 is a communication protocol commonly used in stage lighting applications. It describes the digital data transmission between the controller and the stage equipment, such as a washlight, moving head, or fog machine for example. The E1.11-2008 USITT DMX512-A protocol is maintained by ESTA (Entertainment Service and Technology Association).

This application note describes the implementation of a DMX512 transmitting device using the XC836 Easy Kit from Infineon. We start with an overview of the DMX512 communication protocol, where the system architecture and protocol structure for transmitting device are explained, then discuss the implementation using the XC836 Easy Kit, and finally describe a use-case demonstration with the integrated DALI-DMX512 board to control the on-board RGB LED via the DMX512 protocol.

2 Overview of DMX512

DMX512 is a packet-based, asynchronous, serial and unidirectional communication protocol. Because there is no error checking or correction mechanism specified in the standard, this makes it relatively simple, but also means that it is unsuitable for safety-critical applications.

DMX512 uses differential signals for communication specified by the RS-485 standard. Therefore, it has good immunity against noise and is able to communicate at relatively long distance (up to 1200 meters). However, it also inherits the limitation of RS-485 which allows only up to 32 devices to be connected to the same communication line.

2.1 System Overview

A DMX512 system consists of one master device (transmitting device) connected to multiple slave devices (receiving devices) in "daisy chain" manner as illustrated in **Figure 1** below.

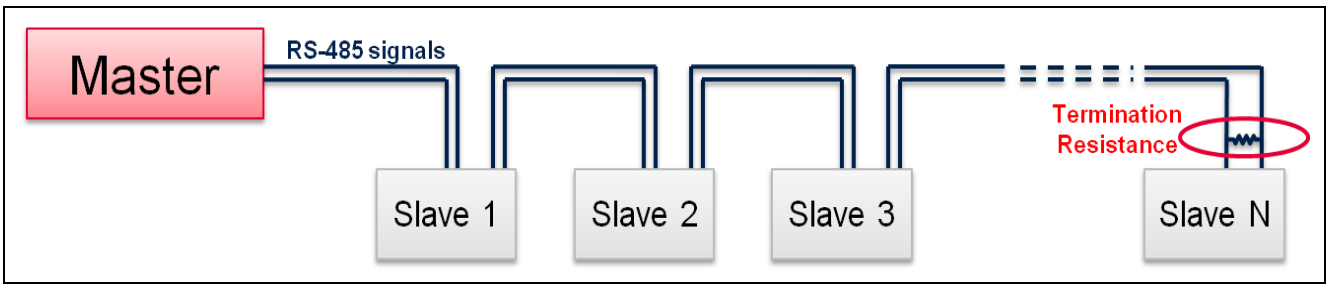

Figure 1 DMX512 System connected in Daisy Chain

A unique address must be assigned to each slave device by configuring the DIP switch embedded on each device. The address may range from 1 to 512, depending on how many devices are connected and how many DMX512 slots/channels are consumed by each device. For example, an RGB-LED wallwasher may consume one DMX512 slot for each color. If its address is set at 10, it will consume slot 10, 11 and 12. The address for the next device must be set to 13.

A termination resistance typically of 120Ω is connected at the furthest slave device to prevent signal reflection.

In the case where more than 32 devices are required in a system, an in-line device, such as optosplitter, can be used.

2.2 Physical Layer

The standard specifies the physical layer of DMX512 to consist of the connector configuration and the circuit topology. DMX512 system uses XLR-5 connectors, as shown in **Figure 2**. Transmitting device uses Ground Referenced Topology while receving device uses Isolated Topology. In this Application Note, only the physical layer of the master device/transmitting device is discussed.

2.2.1 XLR-5 Connector

The standard specifies a female XLR-5 connector shall be used in transmitting device. This is to create the daisy-chain topology that ensures the continuity of communication from the transmitting device to the last receiving device.

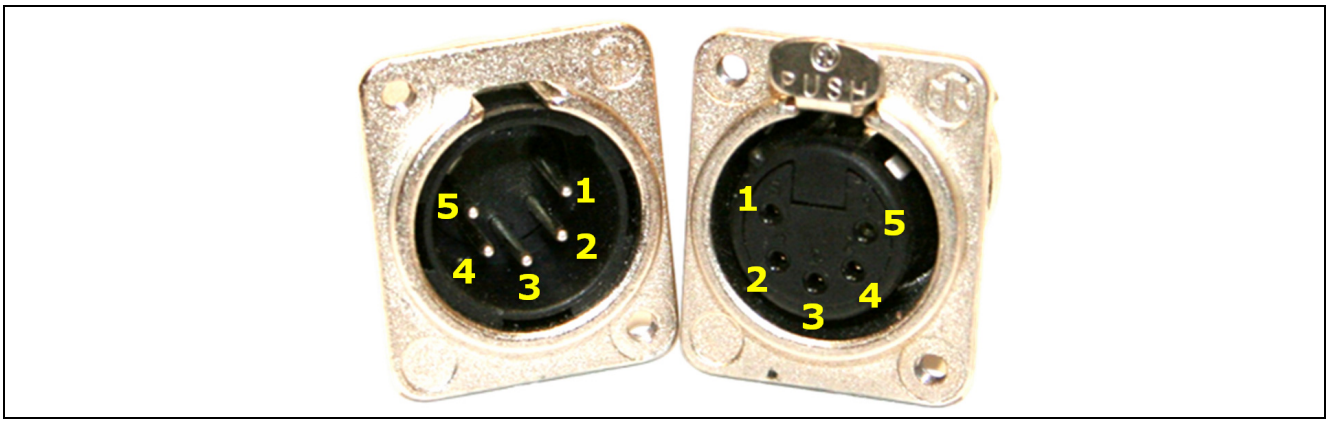

Figure 2 XLR-5 Connector Pinouts. Male-type is on the left and Female-type is on the right.

The following **Table 1** specifies the connection for the transmitting device:

Table 1 DMX512 Pinout on XLR-5 Connector

Pin Number	Signal Name	Description
1	Common Reference	Data Link Common
2	Data 1+	Primary Data Link
3	Data 1-	
4	Data 2+	Secondary Data Link (Optional)
5	Data 2-	

In a typical DMX512 application, only Primary Data Link (DATA1+ and DATA1-) and Common Reference are used.

The Secondary Data Link is not used and reserved for future use. Therefore, it is common to find some lighting fixtures that use XLR-3 connectors.

2.2.2 Ground Referenced Topology

The standard specifies the use of ground referenced topology in the transmitting device circuit, as illustrated in the **Figure 3** below.

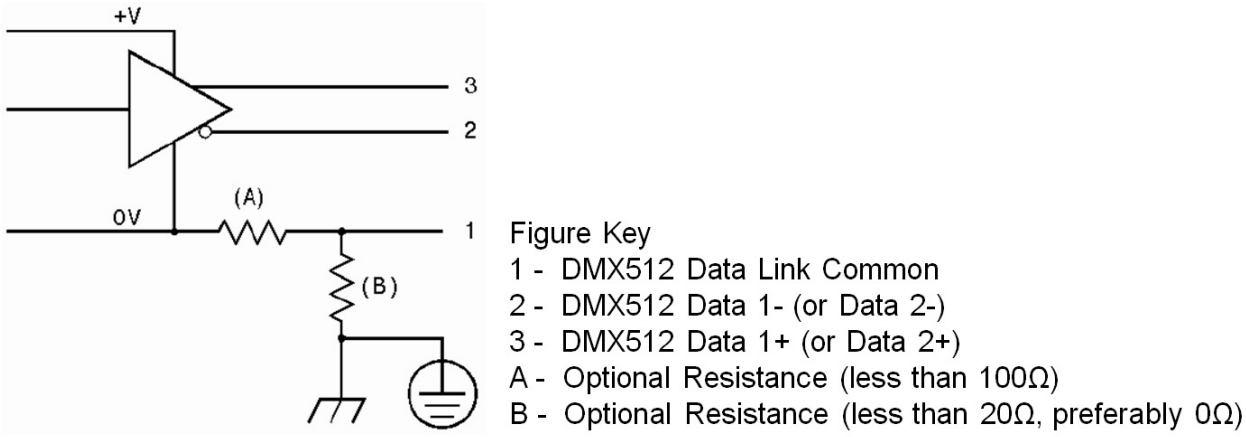

Figure 3 Ground Referenced Topology as specified in the standard

2.3 DMX512 Protocol

As the name suggests, there are 512 "pieces of information" carried in a DMX512 packet. Each "piece of information" is also known as a slot or channel, which consists of 1 start bit, 8 data bits and 2 stop bits. A Reset Sequence consisting BREAK, MAB and NULL Start Code must be transmitted before the slots.

Figure 4 illustrates a DMX512 packet.

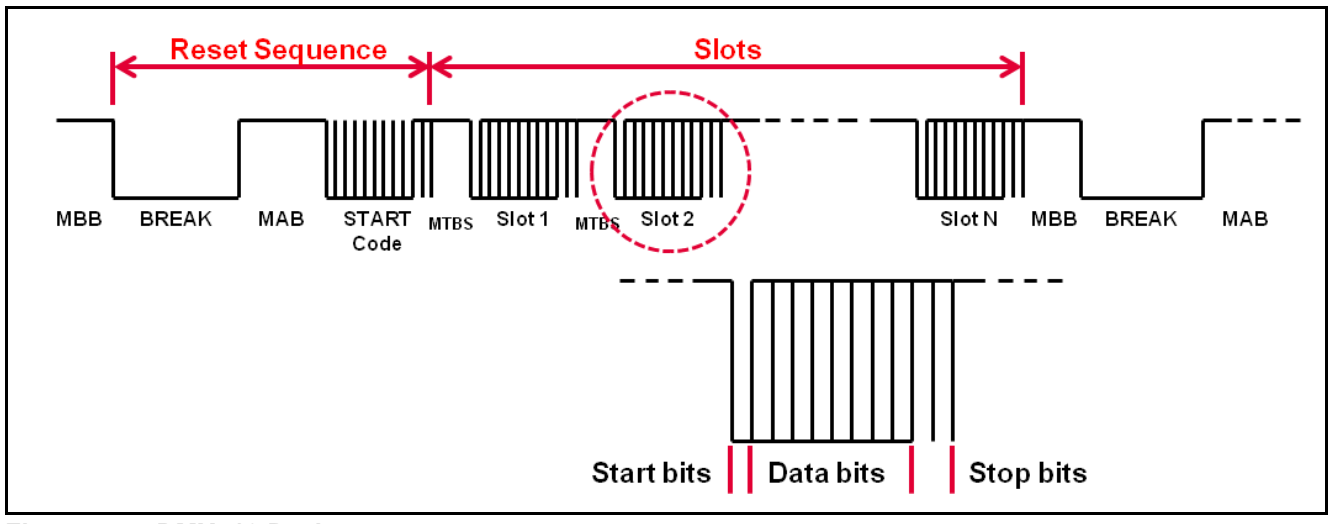

Figure 4 DMX512 Packet

The timing requirements for transmitting device are shown in Table 2.

Table 2 Timing Requirements for Transmitting Device

Signal	Min. Value	Typ. Value	Max. Value	Description
Bit Rate	245 kbps	250 kbps	255 kbps	Transmission rate for DMX512 protocol.
Bit Time	3.92 µs	4 µs	4.08 µs	
BREAK	92 µs	176 µs	-	A falling edge transition followed by a low of at least 88 µs followed by a rising edge.
MAB	12 µs	-	< 1 s	Mark After Break - The period of time measured from the rising edge at the end of BREAK to the falling edge of the start bit of the START Code.
MTBS	0	-	< 1 s	Mark Time Between Slot - The period measured from the end of the second stop bit (bit 9) of the previous slot to the falling edge of the start bit of the current slot.
MBB	0	-	< 1 s	Mark Before Break - The period measured from the end of the second stop-bit of the last slot to the falling edge of the next BREAK.
BREAK-TO-BREAK	1204 µs	-	1.00 s	The period between two BREAKs

3 Implementing A Simple DMX512 Transmitting Device with XC836 Easy Kit

A simple DMX512 transmitting device can be implemented using XC836 Easy Kit because it has on-board capacitive-touch pads and four 7-segment LEDs. This can be used as a simple user interface, where the user can input command and know the value of slots being transmitted. Both are controlled using LEDTS, an integrated peripheral specifically designed for capacitive-touch and LED control application.

The code for transmitting DMX512 signal is implemented using Timer 2 and UART. Timer 2 generates the BREAK, MAB, MBB and MTBS signals, while UART transmits the DMX512 slots.

3.1 Hardware

DMX512 uses differential signals for communication. Therefore, an RS-485 interface board is built to convert MCU signals into differential signals. The board is designed with the same form factor to the Easy Kit connector so that it can fit nicely on it.

Figure 5 shows the XC836 Easy Kit with RS-485 interface board. The schematic can be found in **Figure 19**.

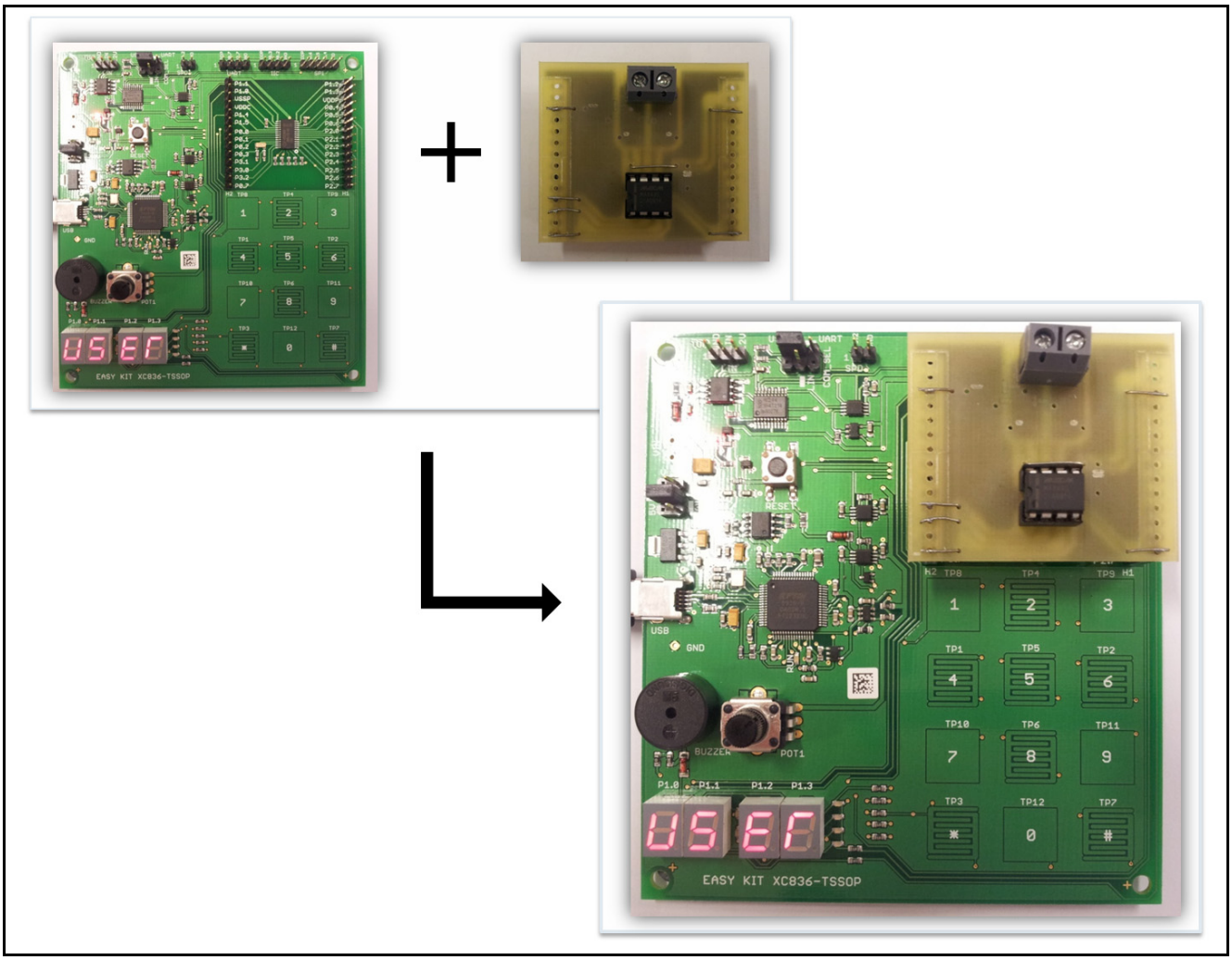

Figure 5 XC836 Easy Kit with RS-485 Interface Board as DMX512 Transmitting Device

3.2 Software

The code uses the following XC836 peripherals:

- Timer 2 and UART are used to transmit the DMX512 signal
- LEDTS is used to take in and display user input in a form of addresses and slot values.

3.2.1 Software Abstraction Layers

Figure 6 shows the software abstraction layers which illustrates the processing of DMX512 signal.

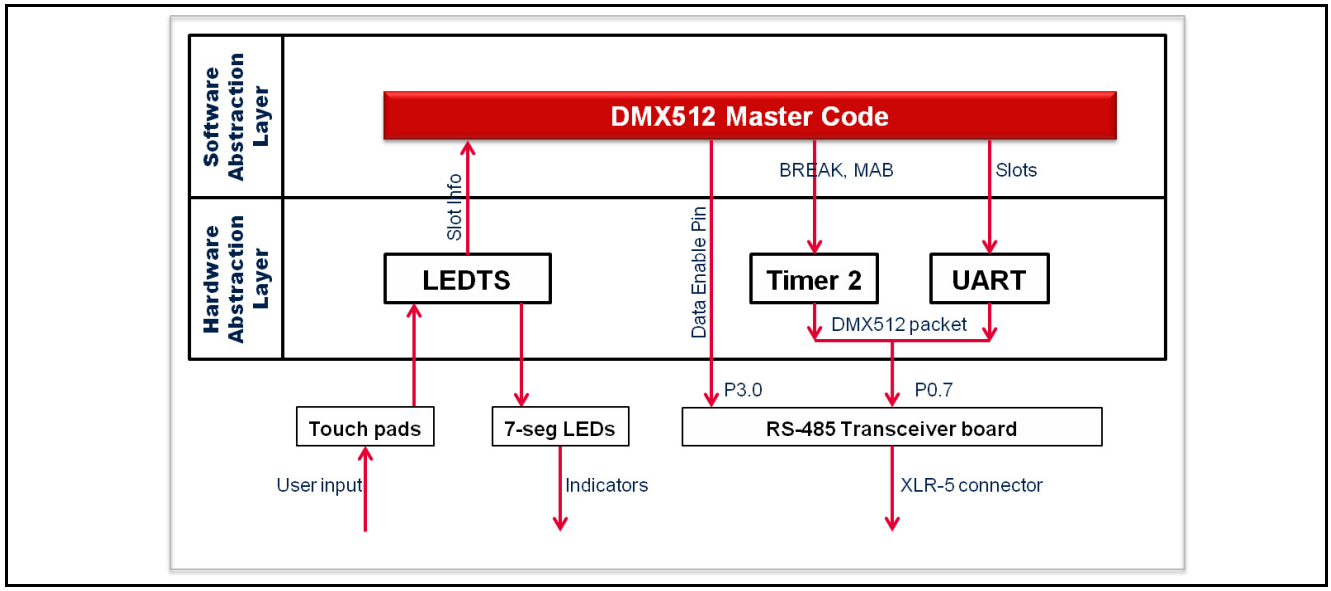

Figure 6 Software Abstraction Layer

DMX512 transmission is controlled by the user via the touch pads. It runs continuously until the user disables it. It is posible to change the slot values and slot address ‘on-the-fly’. Timer 2 provides BREAK, MAB and “window time” for slot transmission. This window time is always greater thanthe slot time (44µs). The difference between these two timings is the MTBS (Mean Time Between Slots).

UART is enabled during MAB transmission to prepare for slot transmission. At every UART Transmit Interrupt, the slot index is increased. When it reaches the specified packet length, UART is disabled and the DMX512 pin is reinitialized back to transmit the next DMX512 packet.

3.2.2 Interrupt Timing Diagram

This process is summarized in **Figure 7**, the interrupt timing diagram.

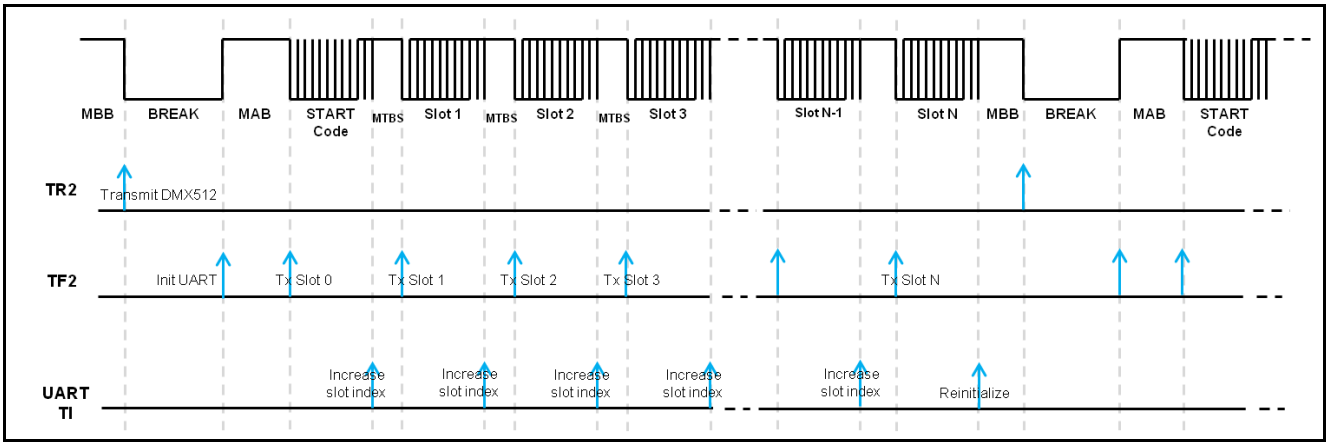

Figure 7 Interrupt Timing Diagram

3.2.3 Source Code Files

The source code files used in the transmitting device are summarized in the following **Table 3**.

Table 3 List of Source Code Files for DMX512 Transmitting Device Code

Code Name	Description
DMX512_MASTER_CONFIG.H	Contains Defines, subroutine declarations and configuration; e.g. DMX512 transmit pin, RS-485 Data Enable pin, number of transmitted slots, scene demo setting, etc.
MAIN.C	Contains pin initializations and slot values initialization.
T2.C	Timer 2 initialization and Timer 2 Interrupt subroutine (TF2 flag) to transmit DMX512 packet (BREAK, MAB and slots).
UART.C	UART Initialization, the interrupt subroutine (TI flag) containing reset sequences for re-transmitting DMX512 packet.
SHARED_INT.C	Contains GUI processing routines to decode user input from touch pads and display to 7-segment LEDs
IO.C	GPIO initialization subroutines. It follows the assigned DMX512 pin in the DMX512_CONFIG.H
LTS.C	LEDTS peripheral initialization.

3.2.4 Transmitted DMX512 Signal Characteristics

The following **Table 4** shows the characteristics of the transmitted DMX512 signal. Note that the measurement values presented are the average values.

Table 4 Transmitted DMX512 Signal Characteristics

Signal	Recommended Value
Bit Rate	250 kbps
Bit Time	4 µs
Number of slots	24
BREAK	184 µs
MAB	44 µs
MTBS	77 µs
MBB	80 µs
BREAK-to-BREAK	3132 µs

4 DMX512 Transmitting Device Code Configuration

This chapter describes the configuration of the implemented software stack. All the settings can be found in DMX512_Master_Config.h

4.1 Configuring DMX512 Slots to be Transmitted

The user can configure the number of DMX512 slots to be transmitted by changing the *#define DMX_PACKET_LENGTH* to the desired number, as shown in **Figure 8**. The defined packet length includes START code (slot 0). In this example 24 slots are available for operation, although the packet length is set to 25.

```
//*****************************************************************************
// DMX512 Transmitting Device Configuration
//*****************************************************************************
#define DMX_PACKET_LENGTH 25
#define MAX_SCENE_NUMBER 4

#define DMX512_PIN P0_7
#define DATA_ENABLE_PIN P3_0
```

Figure 8 Configuring DMX512 Slots to be Transmitted

4.2 Configuring XC836 pins for RS-485 Interface Board

There are two pins required to control RS-485 communication, namely DATA_ENABLE (DE) and DATA_INPUT (DI), as shown in **Figure 9**. This implementation uses P3_0 and P0_7 for DE pin and DI pin, respectively.

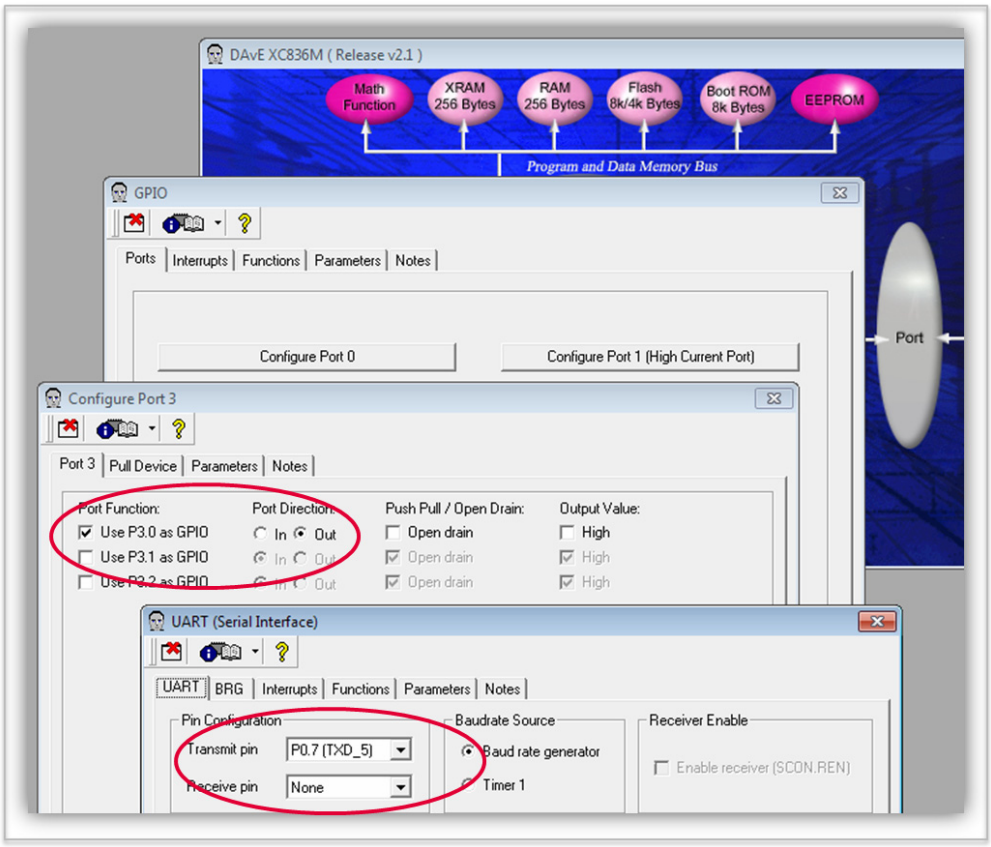

Figure 9 Configuring XC836 pins for Controlling RS485 Interface Board

4.3 Setting Scenes for Demonstration

It is common in stage lighting applications to find the lighting fixtures set and frozen to specific values in a set period of time. This process is known as creating Scenes. Our implementation is equipped with some simple demonstration scenes that work with the DALI-DMX512 boards from Infineon. Please refer to the document ref. AP08131 for more details on these boards.

The demonstration scenes are summarized in the following **Table 5** below. Further explanation on how to evaluate the scenes can be found in **Section 5**.

Table 5 Demo Scenes

Scene #	Description
1	Set slave devices LED to all red
2	Set slave devices LED to all green
3	Set slave devices LED to all blue
4	Set slave devices LED to display red, yellow, green

The demo scenes can be disabled by commenting out the *#define ENABLE_SCENE*, as shown in **Figure 10** below.

```
//*****************************************************************************
// Scene Demonstration Setting
//*****************************************************************************
#define ENABLE_SCENE

#ifdef ENABLE_SCENE
void vUpdateScene(void);
#define MAX_SCENE_NUMBER 4
#endif
```

Figure 10 Enable/Disable Scene

5 Evaluating DALI - DMX512 Board for LED Color Control Application

The DALI-DMX512 Board from Infineon is a receiving or slave device that contains the two common lighting protocols, DALI and DMX512. The board is designed to demonstrate an LED Color Control application on both protocols. This chapter is intended as a step-by-step guide for users to evaluate this board.

5.1 Connecting the Boards in a Daisy-Chain

A simple twisted pair of shielded wires can be used to connect the boards. The diagram inserted in the top-left of the following **Figure 11** shows the wiring connection.

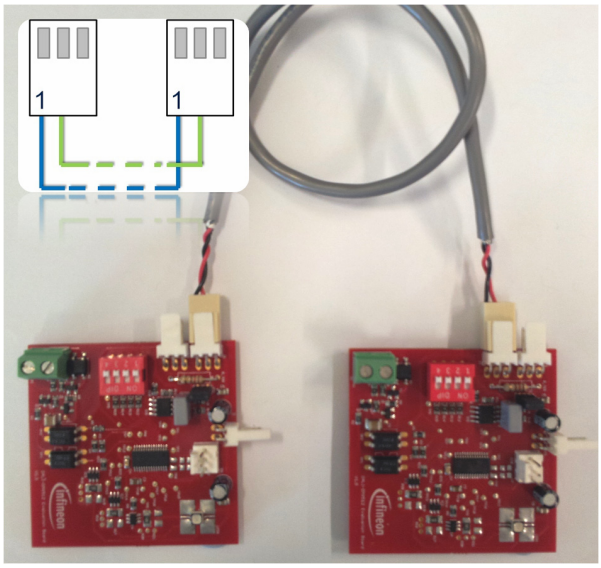

Figure 11 Two Boards connected in Daisy Chain

5.2 Setting the DMX512 Address with DIP switches

In a typical DMX512 device, the user can set the DMX512 address using DIP switch, where DIP switch #4 refers to the Least Significant Bit (LSB) of the address. The following **Figure 12** shows an example where the board is set to Address 11.

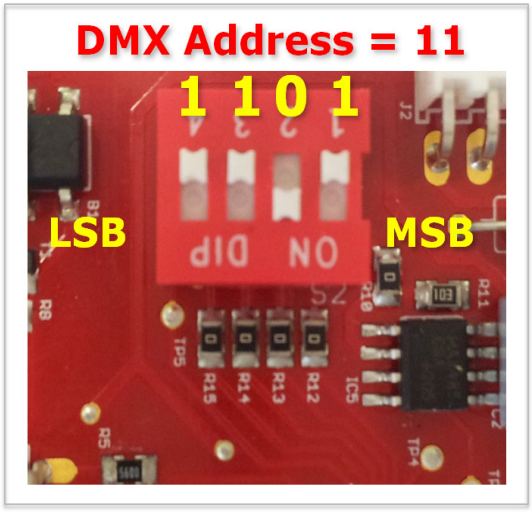

Figure 12 Setting DMX512 Address to 4

5.3 Connecting the Transmitting Device to the Daisy Chain

Once the address of each board is set, the transmitting device can be connected as shown in **Figure 13**.

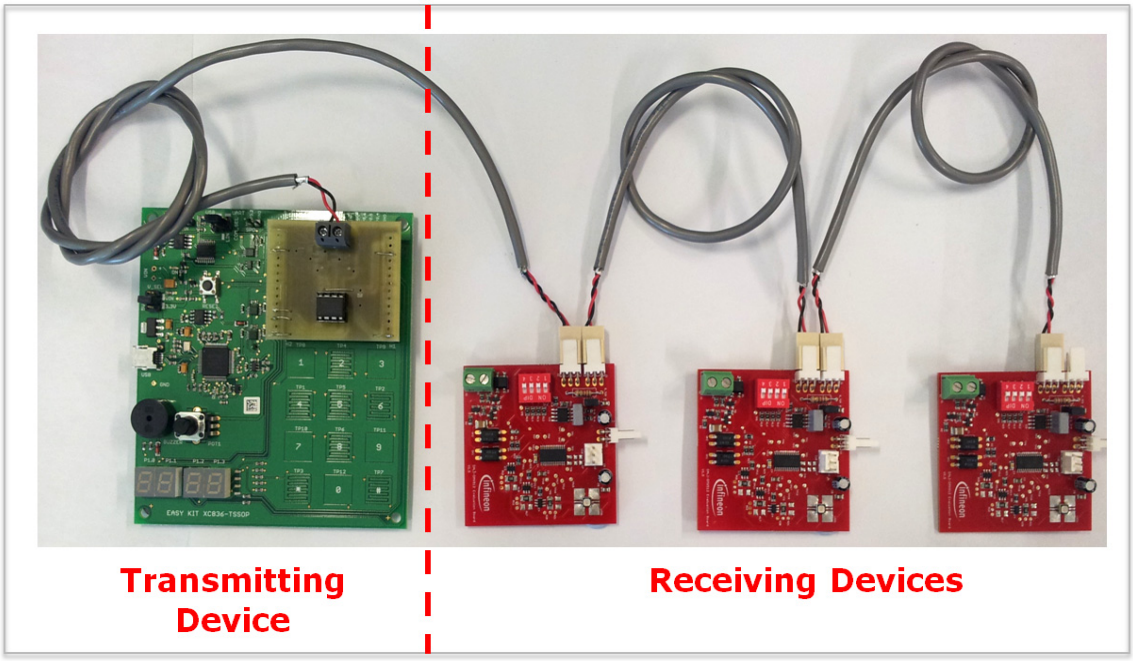

Figure 13 Connecting Transmitting Device with Receiving Devices

5.4 Powering Up the Receiving Devices

Figure 14 shows the receiving devices being powered up from 5V power supply with on-board LED turned on.

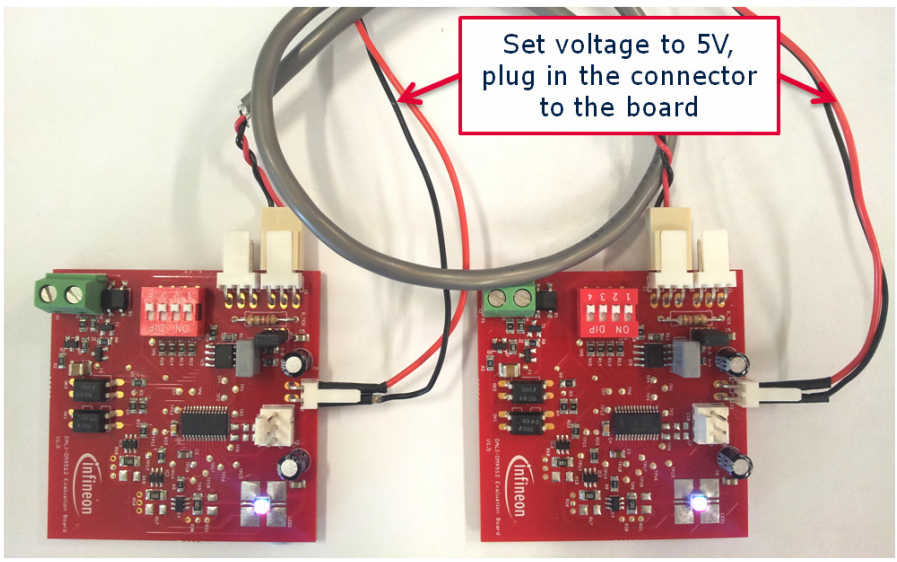

Figure 14 Powering the Receiving Devices

5.5 Powering Up the Transmitting Device from USB

The transmitting device requires 5V DC and is powered from USB, as shown in **Figure 15**.

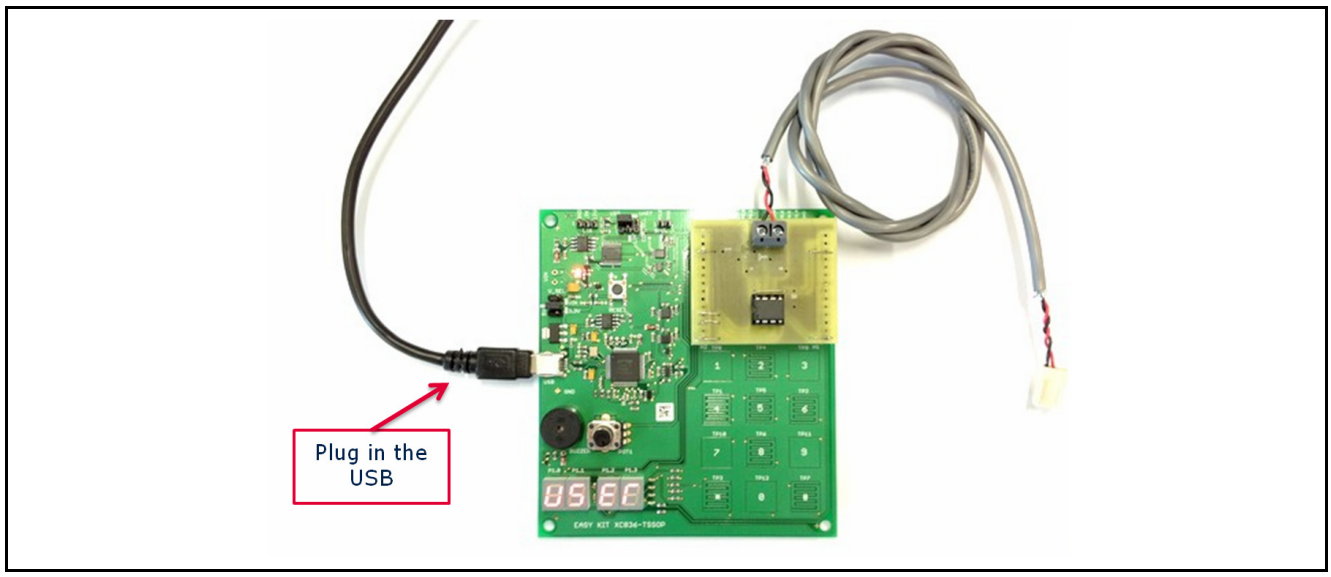

Figure 15 Powering the Transmitting Device via USB

5.6 Controlling LED Color with the Transmitting Device

Figure 16 and **Figure 17** show the touch pads and the 7-segment LED display functions on the transmitting device.

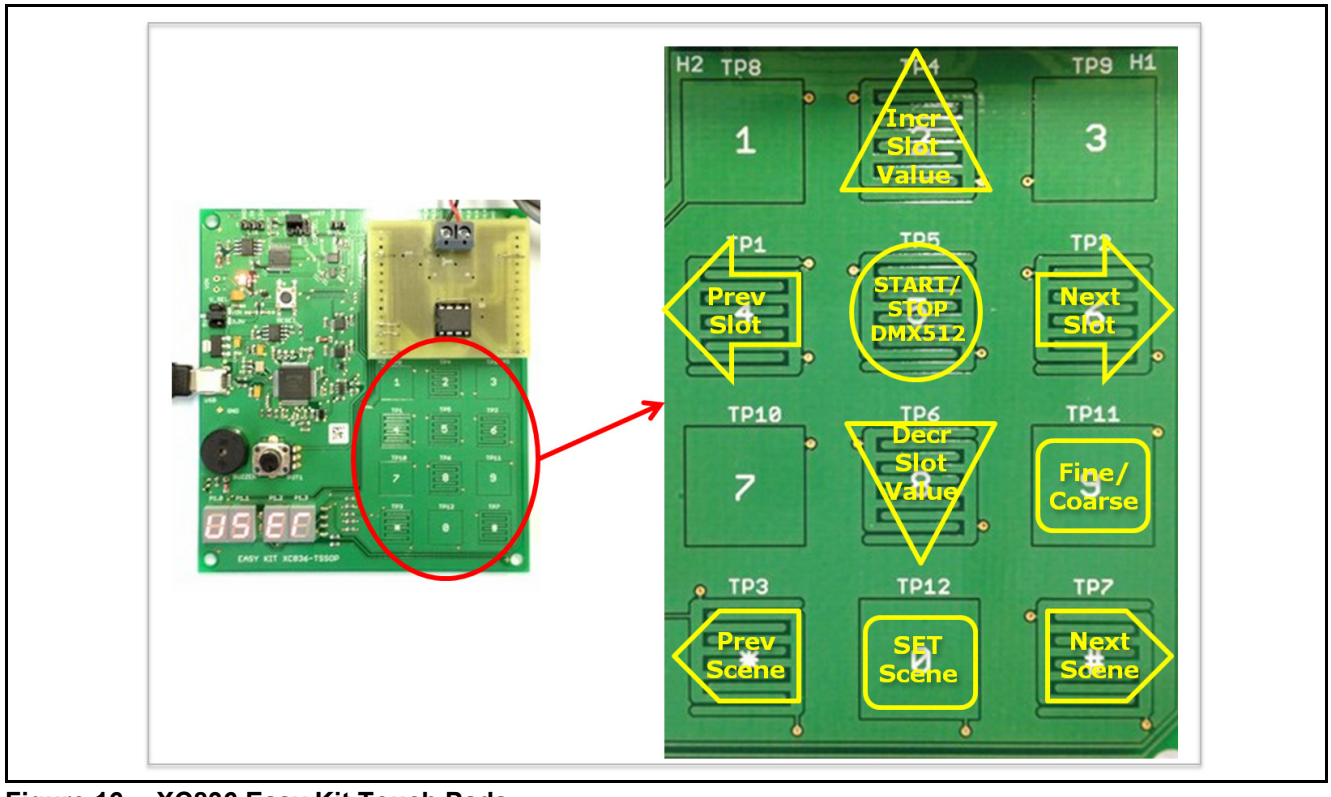

Figure 16 XC836 Easy Kit Touch Pads

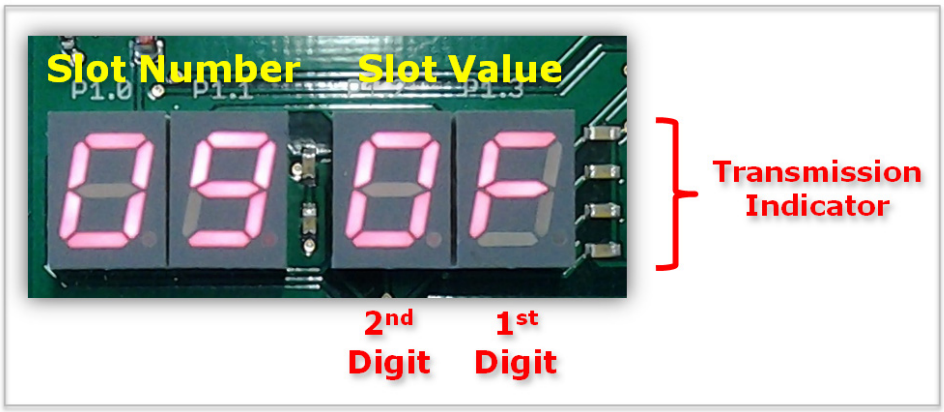

Figure 17 7-Segment LED display functions

The LED to indicate transmission will only light up when there is a DMX512 signal transmission.

The following **Table 6** describes the functionality of each touch pad.

Table 6 Touchpad Descriptions

Touchpad Name	Value Range	Description
Next Address	0 - 24[DEC]	Go to next/previous slot address. For demonstration purposes, only 24 slots are supported by the transmitting device.
Prev Address		
Incr Slot Value	0 - FF [HEX]	Increase/Decrease current slot values. The value is represented as hexadecimal.
Decr Slot Value		
Fine/Coarse	-	Adjust the resolution of Incr/Decr Slot Value. Selecting Coarse Mode will modify the 2nd digit of Slot Value while Fine Mode will modify the 1st digit of the Slot Value. Default mode is Coarse Mode.
START/STOP DMX512	-	Start/Stop DMX512 Transmission
Next Scene	0 - 99 [DEC]	Go to the next/previous predefined scene.
Prev Scene		
Set Scene	-	Set the selected scene to the receiving devices. Pressing this button will automatically enable the DMX512 transmission.

The following **Figure 18** shows the complete setup of DMX512 system for RGB-LED color control application.

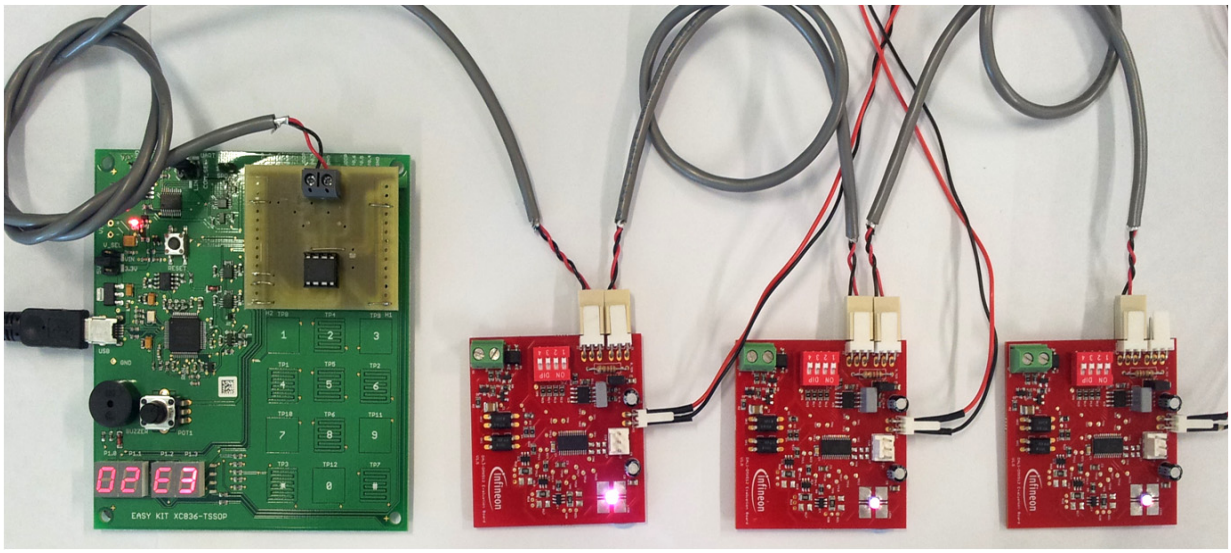

Figure 18 Complete Setup of DMX512 System

The user can now control the LED color on each board from the transmitting device. For example, when the slot value of address 04 is increased, the second board will display brighter red, while increasing the slot value of address 08 will make the third board display brighter green.

6 Summary

This application note has described the implementation of a simple DMX512 transmitting device using the XC836 Easy Kit. For a complete DMX512 solution, please also refer to the AP08131 detailing the receiving device.

7 References

[1] ANSI ESTA E1.11 - 2008 “Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories”

[2] XC82x User Manual version 1.2

[3] XC83x User Manual version 1.1

[4] AP08108 “Programming the BMI Value in the XC82x and XC83x Products”

[5] AP08131 “DMX512 Receving Device using XC836”

[6] AP08102 “DALI Control Gear Software Stack”

[7] AP08114 “DALI Control Device using XC836”

[8] AP08104 “Guide to using the DALI LightNet Tool”

[9] AP08105 “DALI Demo using Touch Sense Control”

Appendix - RS485 Interface Board Schematic

Figure 19 and Figure 20 show the schematic and Layout of of RS485 Interface board.

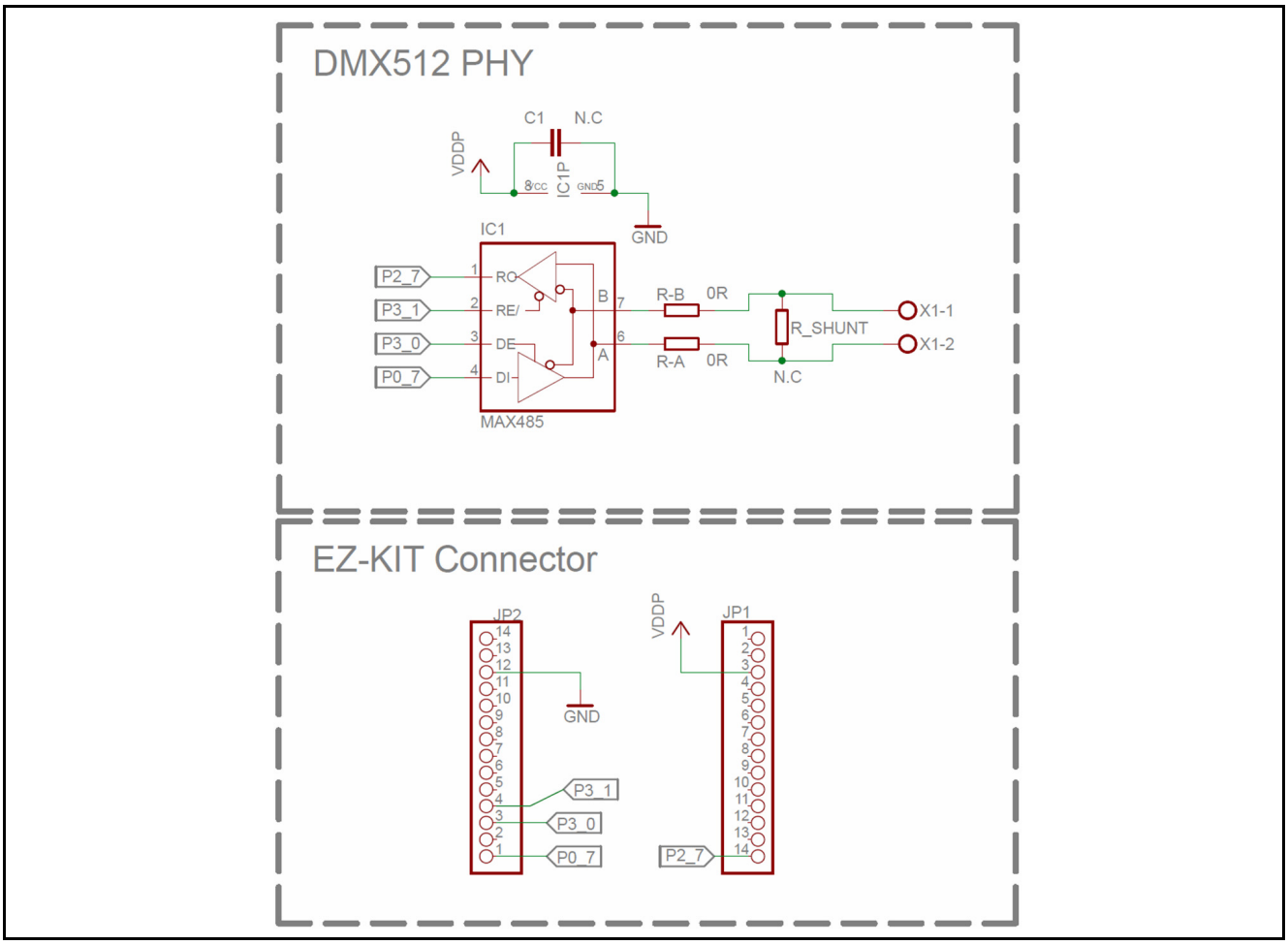

Figure 19 RS485 Interface Board Schematic

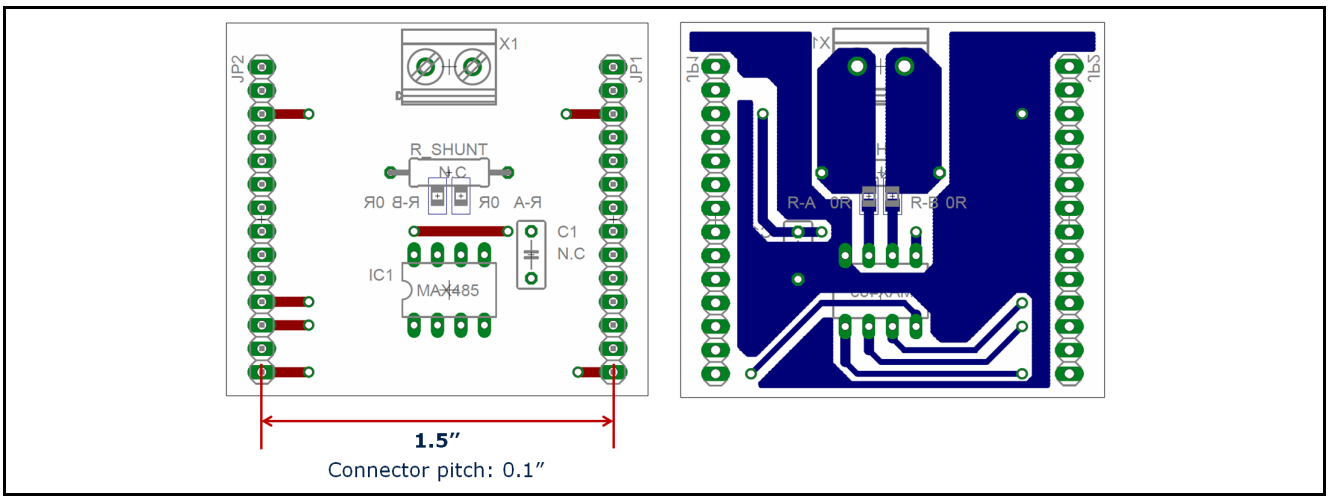

Figure 20 RS485 Interface Board Layout

Appendix - DMX512 Software Stack Flowchart

The following **Figure 21**, **Figure 22** and **Figure 23** shows the flowchart from each peripheral that is used to create DMX512 Master Code.

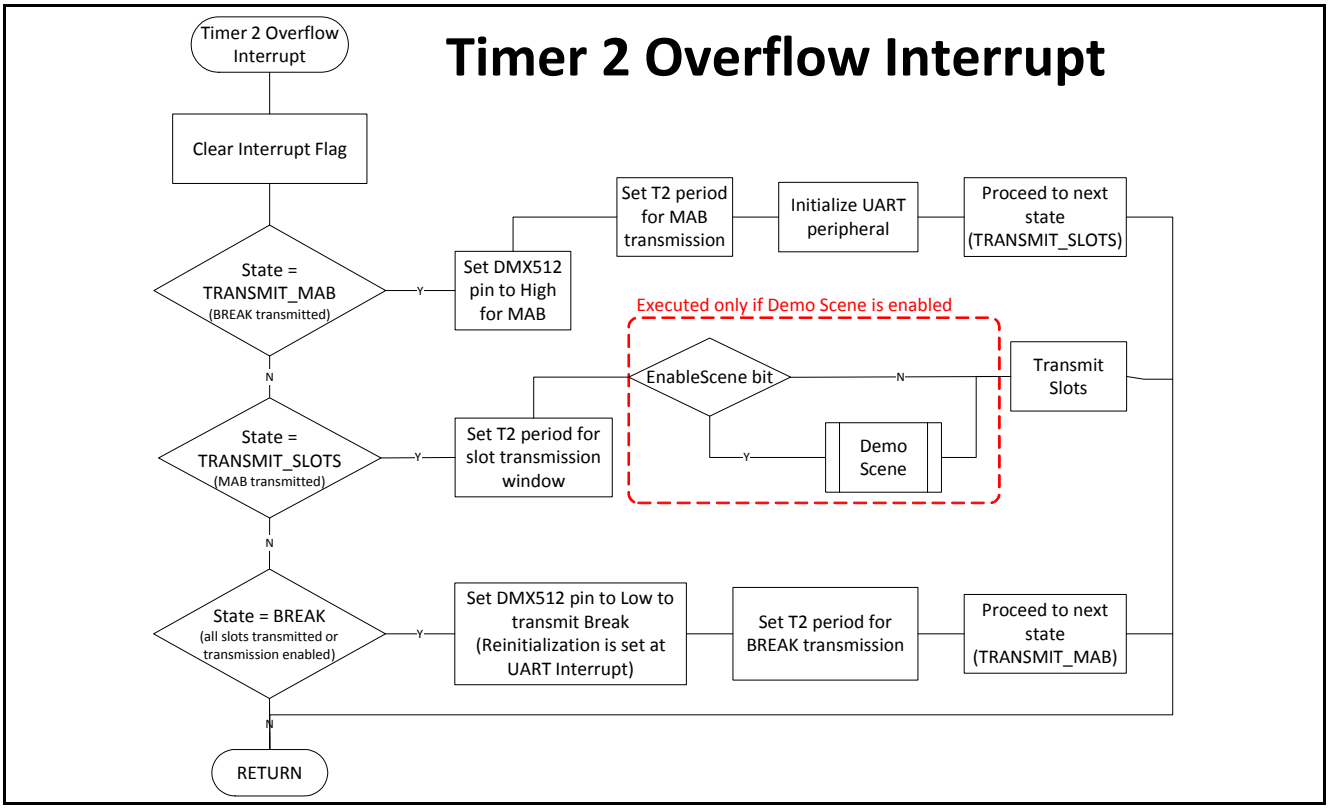

Figure 21 Timer 2 Flowchart

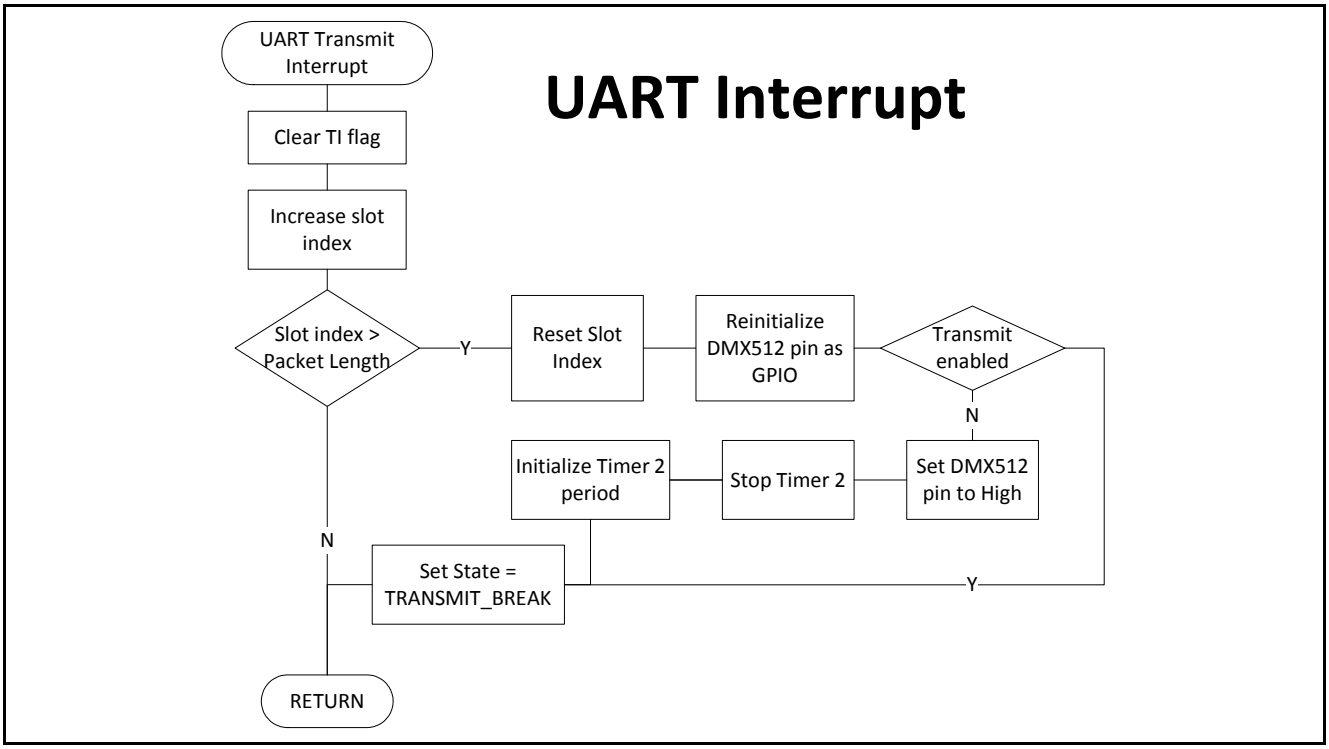

Figure 22 UART Flowchart

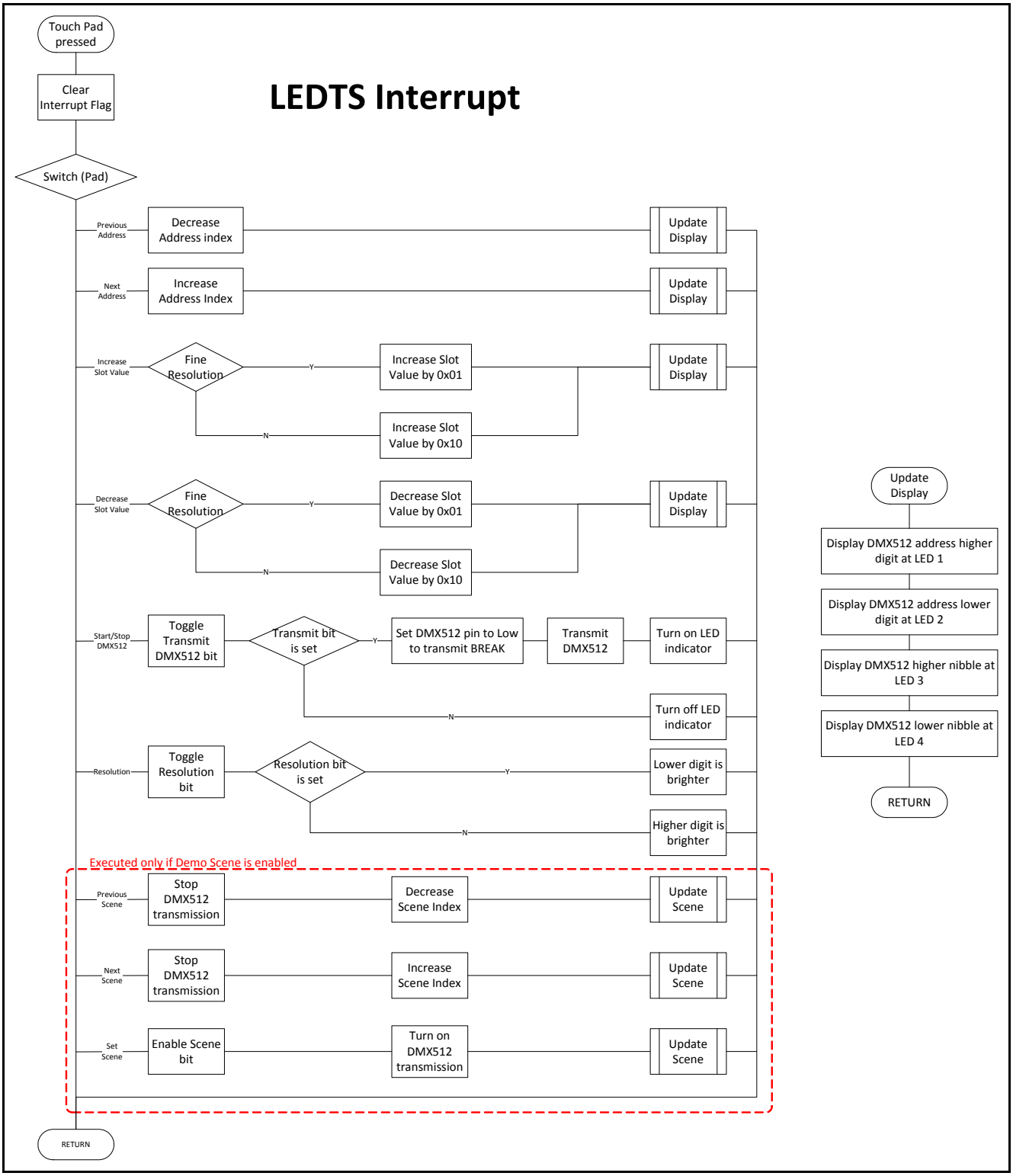

Figure 23 LEDTS Flowchart

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

Published by Infineon Technologies AG