AN11175

DALI master using LPC134x

Rev. 1 — 1 March 2012 Application note

Document information

Info	Content
Keywords	LPC134x, ARM, Cortex M3, DALI, USB, Building lighting
Abstract	This application note describes the use of the NXP LPC134x Cortex M3 microcontroller to create a DALI Master.

Revision history

Rev	Date	Description
1	20120301	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. Document purpose

The purpose of this document is to explain the LPC1343 DALI to USB demo board. Both the hardware and software are described in detail.

This document is intended for technical persons, such as system architects and hardware/software engineers, interested in designing/developing a DALI master using NXP microcontrollers.

For creating a DALI setup with the LPC1343 DALI to USB demo board, refer to "UM10553 DALI getting starting guide"[3].

2. Introduction

The DALI master is an example implementation for a USB enabled DALI controller, built using the NXP LPC134x microcontroller. The DALI master features a USB Human Interface Device (HID) interface, which means that no dedicated software driver needs to be installed on the host PC. The USB, GPIO and 32-bit timer/counter of the LPC134x MCU are the main hardware blocks used to implement the DALI master.

The software can be built with

- LPCXpresso v4.1.0_190, and
- IAR Embedded Workbench for ARM v6.20.4.

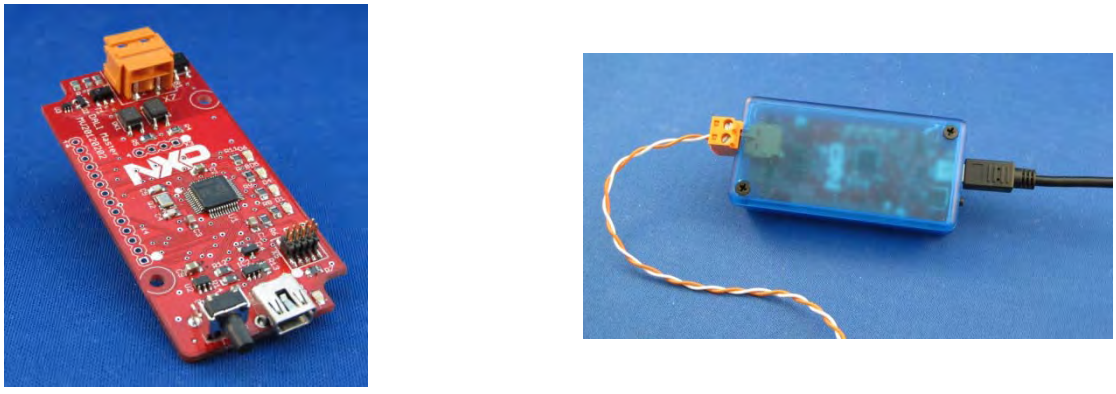

Fig 1. DALI Master assembled PCB

For background information on the DALI standard, refer to [1].

3. DALI master hardware

3.1 Overview

The schematic of the DALI master is shown below.

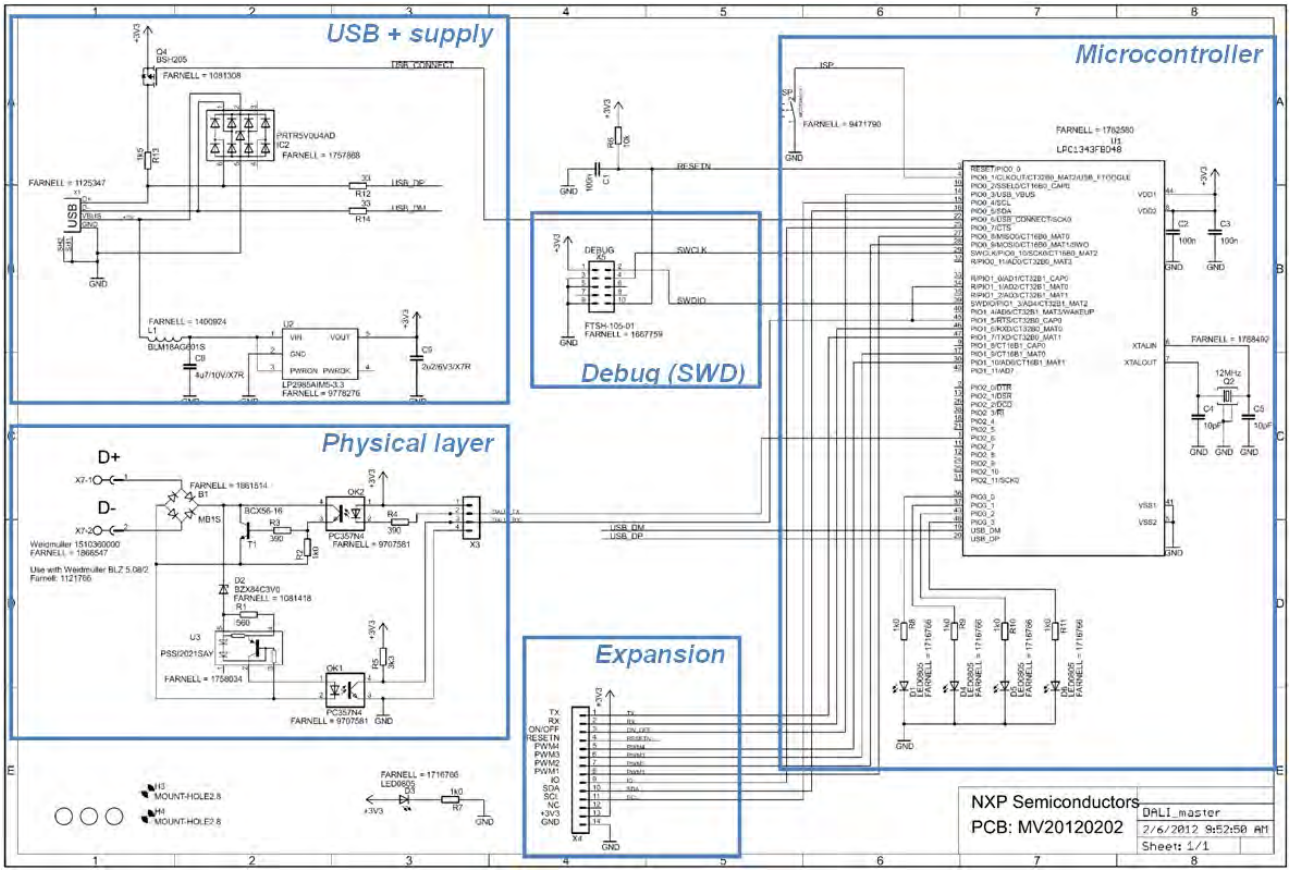

Fig 2. DALI master schematic overview

3.2 Physical layer

The physical layer is an example circuit and does not fully comply with the DALI specification. The circuit is not protected for mains voltage and therefore does not comply with the overvoltage protection test (62386-102 © IEC:2009: 12.1.4.2 Test sequence 'Overvoltage protection').

The DALI bus connects to the connections D+ and D- of the DALI connector. The connector is a Weidmuller header used with a Weidmuller socket block. The connection is polarity independent.

The DALI physical layer is isolated using opto-couplers from the rest of the electronics.

The design is made polarity independent by using bridge rectifier B1. Both terminals D are interchangeable.

The transmission part of the DALI slave is created with T1, R2, R3, OK2 and R4. The microcontroller drives signal DALI1_TX at 0 V or 3.3 V. For high signals of DALI1_TX, Optocoupler OK2 connects junction R2 and R3 to the DALI bus, which creates a drive current for the base of T1 that starts to conduct and short circuit the DALI bus via bridge B1. When the signal DALI1_TX is low, transistor T1 does not conduct and the bus is in the ‘high’ state. Resistor R3 (390 Ohm) allows T1 to become saturated and creates a low voltage level on the bus while having a low voltage drop on T1. This keeps the power dissipation of T1 limited as it should be able to sink the maximum DALI bus current of

250 mA; for this reason T1 should have a high h_{FE} of minimum 100. In this way, no additional mounting area for cooling area is needed.

The reception path is created around optocoupler OK1, U3, R1, R5 and Zener diode D2. When the DALI bus is idle (high), a constant current source of about 1 mA is created using U1 and R1. This current is used to drive optocoupler OK1 that signals the level of the DALI bus to the microcontroller via DALI1_RX. The current source limits the maximum current load the circuit creates when not in transmission mode. Zener D2 and bridge rectifier B1 drop the received bus voltage to a level to guarantee that a low level voltage of 6.5 V does not drive the optocoupler.

The circuit around OK1 and R3 creates an inverted signal to the microcontroller; a high DALI bus level will connect DALI1_RX to low. A low DALI bus level creates a high signal on DALI1_RX.

All components are chosen to withstand several factors (70 V to 80 V) of the highest allowed DALI bus voltage level of 22.5 V. The physical layer does not contain overvoltage protection or suppression components; overvoltage protection is the user's responsibility.

The optocouplers create isolation between the microcontroller side and the DALI bus. The isolation is sufficient for evaluation of the DALI software stack when the microcontroller is connected in a non-isolated way to the mains supply. Any re-use of this design should be made compliant to the isolation requirements as specified in [IEC62383-101] section 5.4[1].

3.3 Microcontroller

The system is built around NXP's LPC1343 device, which is a Cortex-M3 running at frequencies of up to 72 MHz.

Included are a USB 2.0 Full-Speed device controller, 32 kB on-chip Flash, 8 kB SRAM, UART, SSP, I2C, ADC, etc.

Debugging and flashing connection is provided by header X5, which complies with the 10-pin SWD standard supported by many flash and software tools.

If the device is connected to USB while the ISP push button is pressed, the DALI Master connects as a Mass Storage device to the PC. This allows programming via drag and drop of a binary image onto the DALI Master via a file explorer.

Four LEDs are provided for general purpose use (e.g. status information). One of the LEDs is used to signal DALI bus transmission when the DALI master sends data to the bus.

3.4 USB + Supply

The USB interface uses a mini-AB socket.

The connection to the USB is accomplished by bringing USB_D+ (for a full-speed device) HIGH through a 1.5 kOhm pull-up resistor via a FET. The USB SoftConnect feature can be used to allow software to finish its initialization sequence before deciding to establish connection to the USB. Re-initialization of the USB bus connection can also be performed without having to unplug the cable.

Extra ESD protection is provided with IC2.

3.5 Expansion

An expansion header X4 (optional) can be mounted to provide extra connections to the microcontroller.

3.6 Board Layout

The layout of the board (74.0 mm x 34.0 mm) is given below.

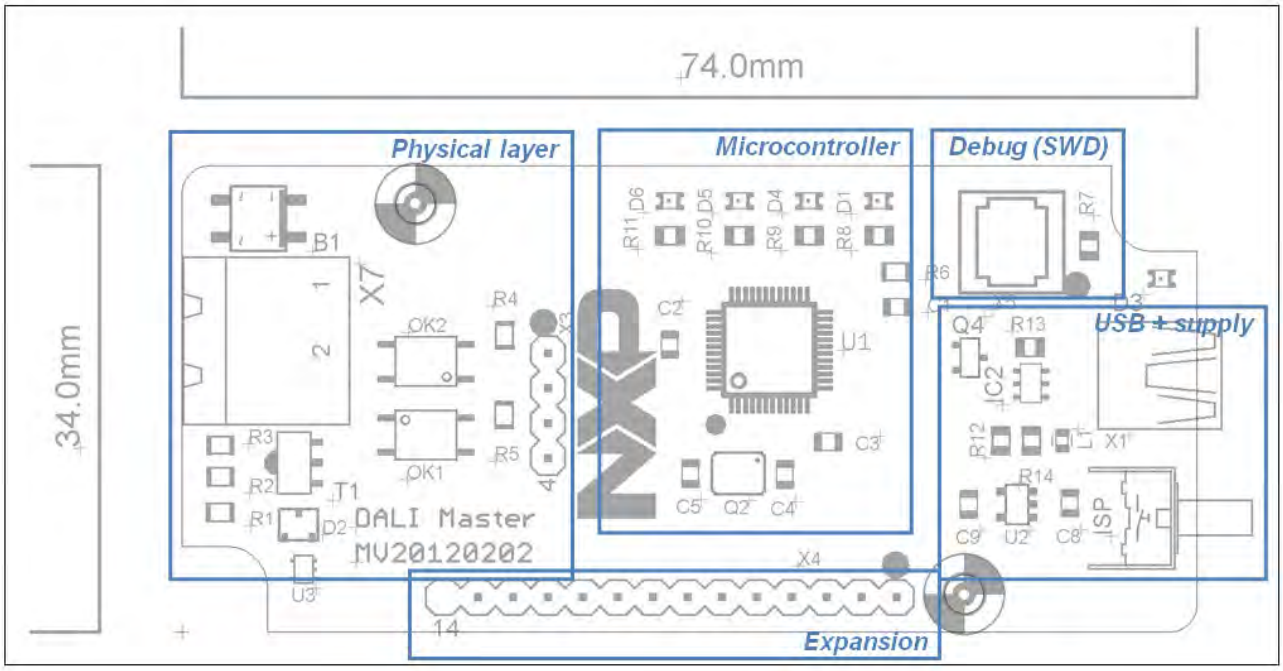

Fig 3. Board layout component placement

Table 1. DALI master parts list

Part	Value	Device	Package	Description	Farnell
B1	MB1S	MB1S	SOIC-4	Bridge Rectifier	1861514
C1	100n	C-EUC0603	C0603	CAPACITOR, European symbol	
C2	100n	C-EUC0603	C0603	CAPACITOR, European symbol	
C3	100n	C-EUC0603	C0603	CAPACITOR, European symbol	
C4	10pF	C-EUC0603	C0603	CAPACITOR, European symbol	
C5	10pF	C-EUC0603	C0603	CAPACITOR, European symbol	
C8	4u7/10V/X7R	C-EUC0603	C0603	CAPACITOR, European symbol	
C9	2u2/6V3/X7R	C-EUC0603	C0603	CAPACITOR, European symbol	
D1	LED0805	LEDCHIPLED_0805	CHIPLED_0805	LED	1716766
D2	BZX84C3V0	BZX84CSMD	TO236	Z DIODE	1081418
D3	LED0805	LEDCHIPLED_0805	CHIPLED_0805	LED	1716766
D4	LED0805	LEDCHIPLED_0805	CHIPLED_0805	LED	1716766
D5	LED0805	LEDCHIPLED_0805	CHIPLED_0805	LED	1716766
D6	LED0805	LEDCHIPLED_0805	CHIPLED_0805	LED	1716766
IC2	PRTR5V0U4AD	PRTR5V0U4AD	ESD-PROTECTION_SOT457R_PHILIPS		1757868
ISP	MCDTSA6-5K	MCDTSA6	TACT90L		9471790
L1	BLM18AG601S	L0603	P0603		1400924
OK1	PC357N4	PC357N4TJ00F	PC357N	Optocoupler, Phototransistor Output	9707581
OK2	PC357N4	PC357N4TJ00F	PC357N	Optocoupler, Phototransistor Output	9707581
Q2	12MHz	CRYSTAL4	XTAL4		1788492
Q4	BSH205	BSH205	SOT23	P-channel enhancement mode MOS transistor	1081308
R1	560	R-EU_R0603	R0603	RESISTOR, European symbol	
R2	1k0	R-EU_R0603	R0603	RESISTOR, European symbol	
R3	390	R-EU_R0603	R0603	RESISTOR, European symbol	
R4	390	R-EU_R0603	R0603	RESISTOR, European symbol	
R5	3k3	R-EU_R0603	R0603	RESISTOR, European symbol	
R6	10k	R-EU_R0603	R0603	RESISTOR, European symbol	
R7	1k0	R-EU_R0805	R0805	RESISTOR, European symbol	
R8	1k0	R-EU_R0805	R0805	RESISTOR, European symbol	

Part	Value	Device	Package	Description	Farnell
R9	1k0	R-EU_R0805	R0805	RESISTOR, European symbol	
R10	1k0	R-EU_R0805	R0805	RESISTOR, European symbol	
R11	1k0	R-EU_R0805	R0805	RESISTOR, European symbol	
R12	33	R-EU_R0603	R0603	RESISTOR, European symbol	
R13	1k5	R-EU_R0603	R0603	RESISTOR, European symbol	
R14	33	R-EU_R0603	R0603	RESISTOR, European symbol	
T1	BCX56-16	BC868-NPN-SOT89-BCE	SOT89-BCE	NPN Transistor	
U1	LPC1343FBD48	LPC1343FBD48	LQFP48	NXP Semiconductors	1762580
U2	LP2985AIM5-3.3	LP2985AIM5	SOT23-5	V REG LDO +3.3V	9778276
U3	PSSI2021SAY	PSSI2021SAY	SOT353	Constant current source in SOT353 package	1758034
X1	USB	USB	USB-MB-H2		1125347
X3	Pin header 4x1	MA04-1R	MA04-1R	PIN HEADER	
X4	Pin header 14x1	MA14-1	MA14-1	PIN HEADER	
X5	FTSH-105-01	FTSH-105-01	CON-SAMTEC_FTSH-105-01-XXX-DV-K		1667759
X7	Header 2p			Weidmuller 1510360000	1866547
X7a	Socket block 2p			Weidmuller BLZ 5.08/2	1121766

3.7 DALI master schematics

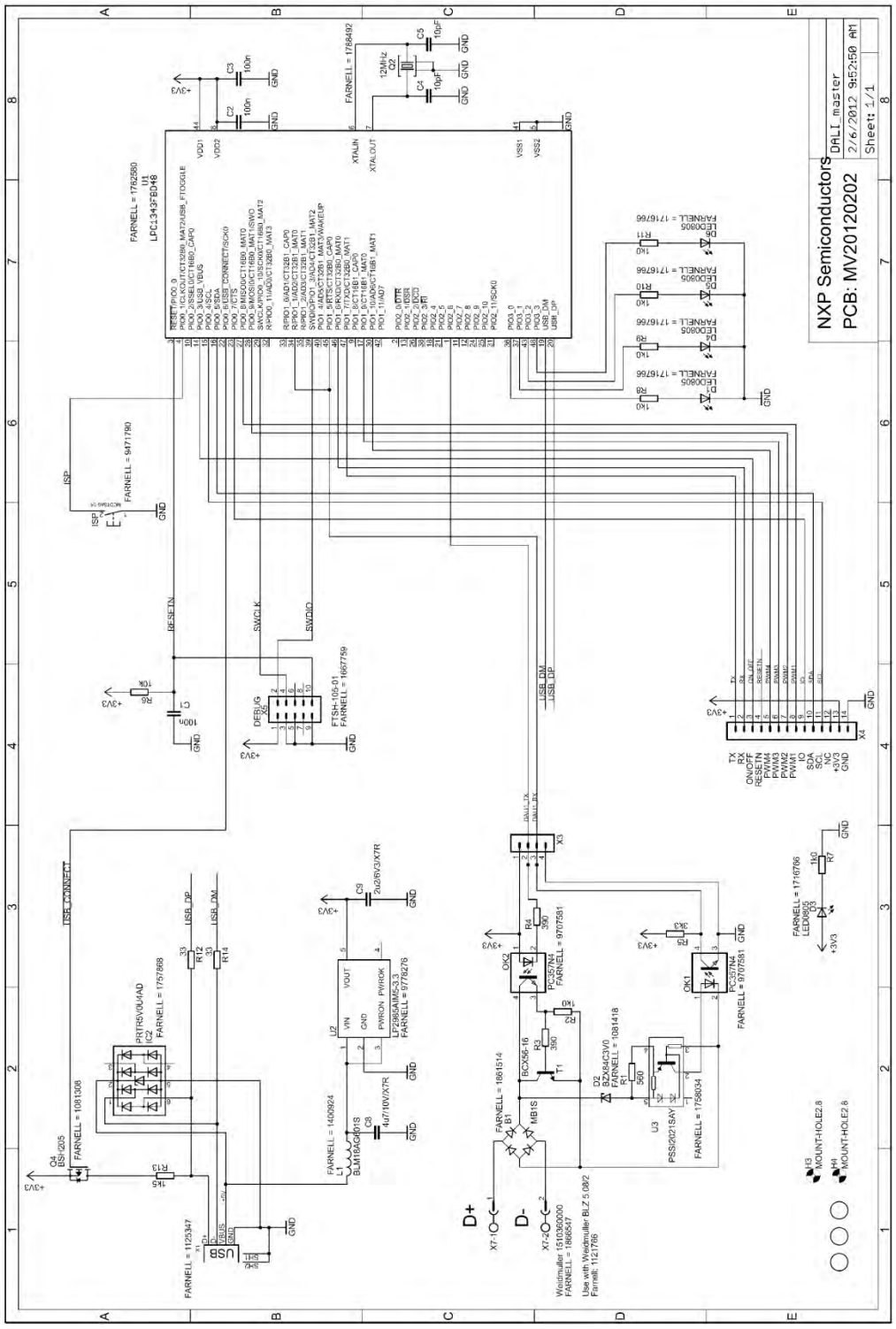

Fig 4. Schematics

4. DALI master software

4.1 Architecture

The software of the DALI master uses the USB HID software driver located in the boot ROM of the LPC134x. The software running on the DALI master does not incorporate a Real Time Operating System; this allows the memory footprint of the firmware to remain as small as possible. The software is written in the “C” programming language.

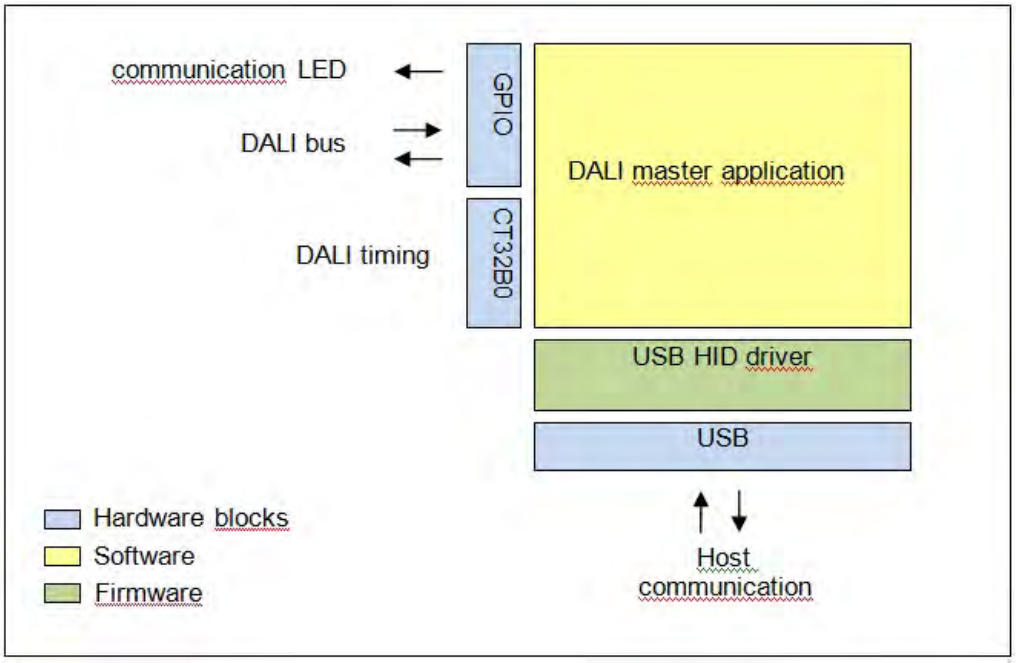

Fig 5. Software architecture of DALI master

The software of the DALI master enables it to function as a controller on the connected DALI bus.

4.2 Main software execution flow

Fig 6 shows the flowchart of the DALI master software. The DALI master initialization is described in section 4.3. The host can invoke a command via USB to the DALI master which puts this command on the DALI bus. When an answer is expected, the DALI master gets the answer from the DALI bus and passes it back to the host via the USB interface.

Interfacing from the DALI master application with the USB HID software driver is via two callback functions: DALI_SetOutReport() and DALI_GetInReport(). The OutReport holds the forward frame (one address byte, plus one data byte) as sent by the host. The InReport holds the backward frame to be sent to the host. When the DALI master is busy on the DALI bus, the LED on the PCB will be on, and it won’t accept a new command from the host. The value of the forward frame defines whether the DALI master will put the command only one time or two times (repeat) on the DALI bus.

The actual data that is put on the DALI bus is defined as follows:

- One start bit (logic 1, bi-phase coded)
- One address byte (bi-phase coded)
- One data byte (bi-phase coded)
- Two stop bits (idle line)

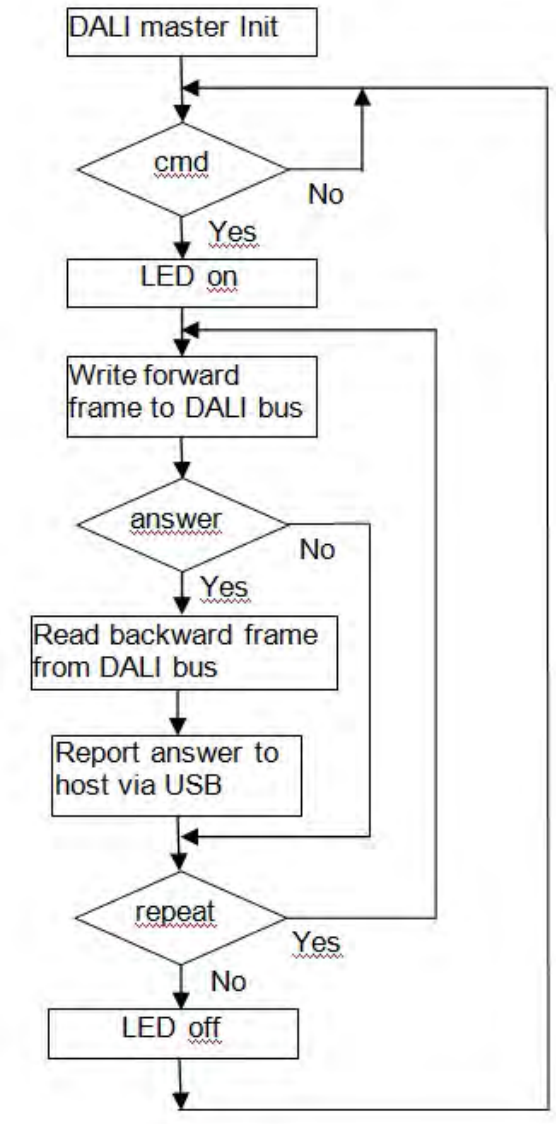

Fig 6. Flowchart of DALI master

The forward frame is bi-phase (Manchester) encoded before it is put on the DALI bus. A logic 1 bit of the forward frame is encoded as symbol “01”; a logic 0 bit of the forward frame is encoded as symbol “10”. The DALI information rate is 1200 bit/s, and the symbol rate on the DALI bus is 2400 symbols/s. Fig 7 shows a DALI transmission between two devices on the bus.

One 32-bit timer is used to put the symbols on the DALI bus at the correct time. PIO pin P2.6 of the LPC134x drives the DALI transmit line; PIO pin P1.1 monitors the DALI receive line.

The first symbol (first half of start bit) is put on the DALI bus by the main loop of the DALI master software and the remaining 37 symbols (one start bit symbol + 32 forward frame symbols + four stop bits symbols) are put on the DALI bus by the timer interrupt handler. When transmitting all the symbols on the DALI bus is finished, the timer interrupt handler captures the optional response symbols from a DALI slave.

The capture unit (part of the 32-bit timer hardware block in the LPC134x) detects a change in the line value on the DALI receive pin, which then generates an interrupt. This causes the timer interrupt handler to store the line value and the time offset since the start of the DALI slave answer. When the timer signals the end of the answering time window, the timer interrupt handler takes care of DALI bus idle timing. When the timer interrupt handler finishes its work, the main loop continues to decode the received symbols from the DALI slave and returns this answer back to the host.

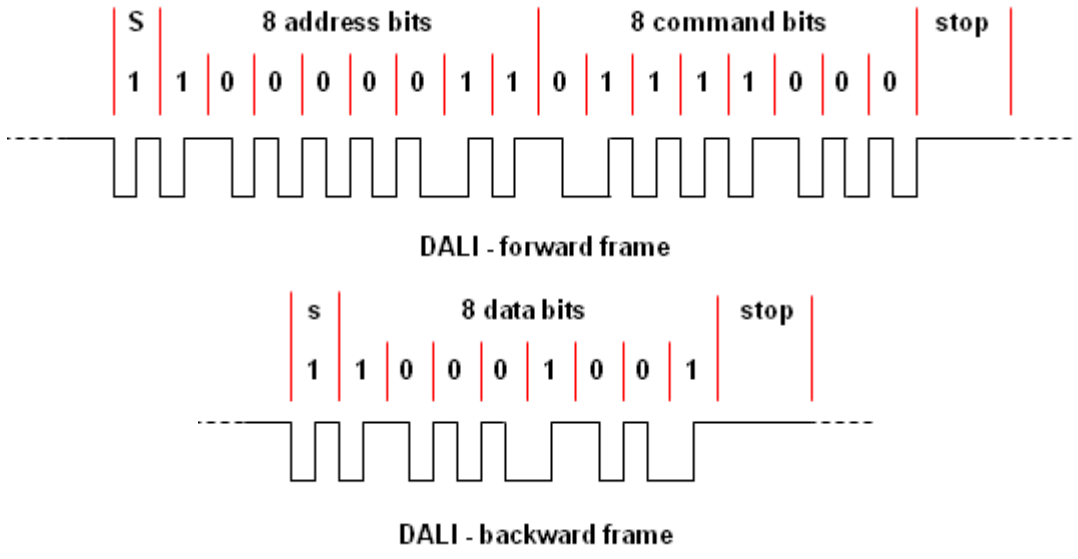

Fig 7. Example of DALI data on the bus

4.3 Initialization sequence

The following occurs when the DALI master is powered: the LPC134x microcontroller starts up, the stack pointer is loaded with the value from address 0:3, and the program counter is loaded with the value from address 4:7 (the address of the reset handler). This results in the calling of main().

The main() function directly calls the function SystemInit(), and after that the function DALI_Thread() is called. SystemInit() sets up the main PLL that is used to set the proper CPU clock, as configured by LPC_CORE_CLOCKSPEED_HZ in 'app_config.h'. Function SystemInit() is from the CMSIS library. The function DALI_Thread() implements the endless loop which handles the USB communication and transmits/receives data to/from the DALI bus. Before the endless loop is entered, the LPC I/O pins for driving the DALI bus are setup, the timer hardware block is initialized, the timer interrupt is enabled, the USB HID driver is initialized, and the LPC output pin for controlling the LED is configured.

4.4 USB HID driver

The USB HID driver is placed in the boot ROM of the LPC134x, and can't be changed by the user. The DALI master software has a very thin glue layer to initialize this USB HID driver, and to register the callback functions InReport() and OutReport().

4.5 Interrupts

The following table shows the interrupts that are handled by the software.

Table 2. DALI master used interrupts

Interrupt	Description
USB_IRQ	Generated by the USB HW block. Used for transmitting/receiving data to and from the host.
TIMER_IRQ	Generated by the 32-bit timer HW block. Used for transmitting/receiving data to and from the DALI bus.

4.6 Software source code files

The software tree holding the source code files of the DALI master is shown in Fig 8.

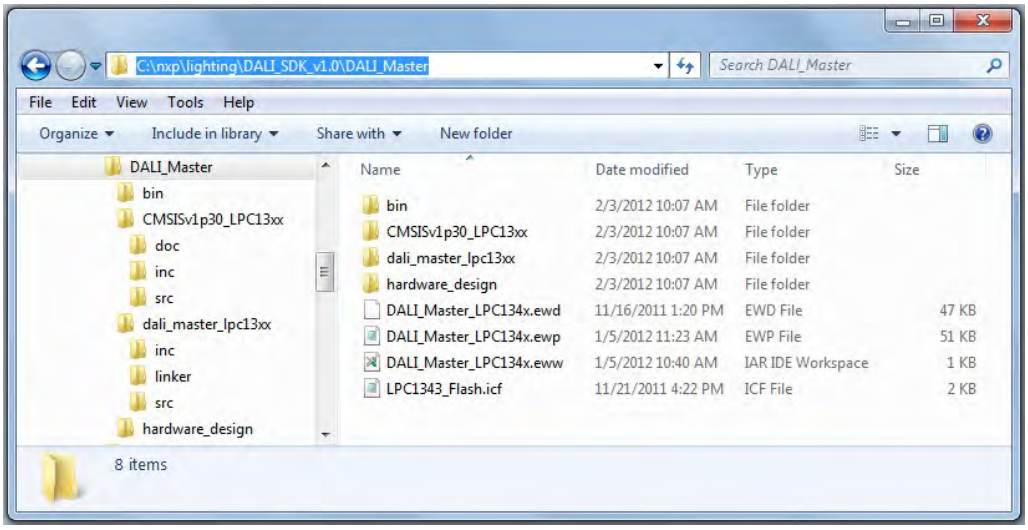

Fig 8. Software tree of DALI master

All DALI master source code is contained in the directory DALI_Master, which consists of a “CMSIS” directory containing the CMSIS source code; a “dali_master_lpc13xx” directory containing the source code of the DALI master; and the project files of the IAR IDE.

Fig 9 shows the contents of the “dali_master_lpc13xx\src” directory, while Fig 10 shows the contents of the “dali_master_lpc13xx\inc” directory.

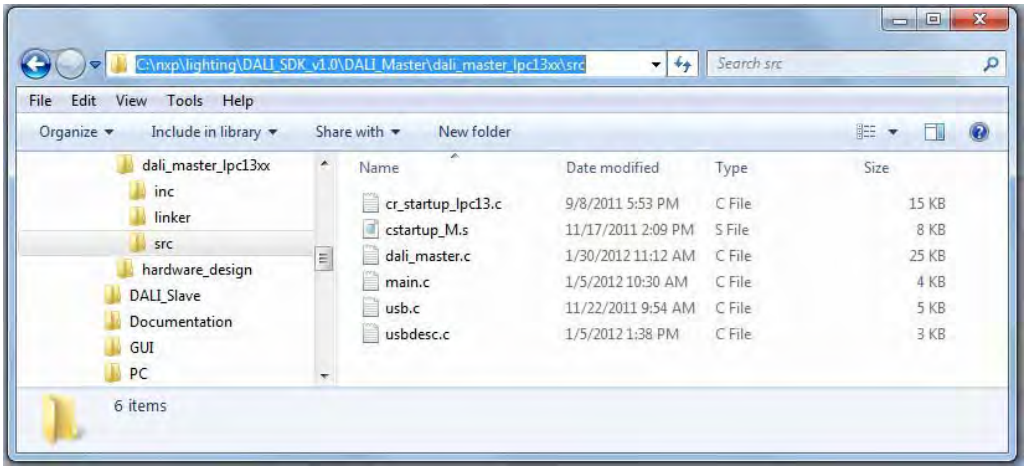

Fig 9. DALI master source code files

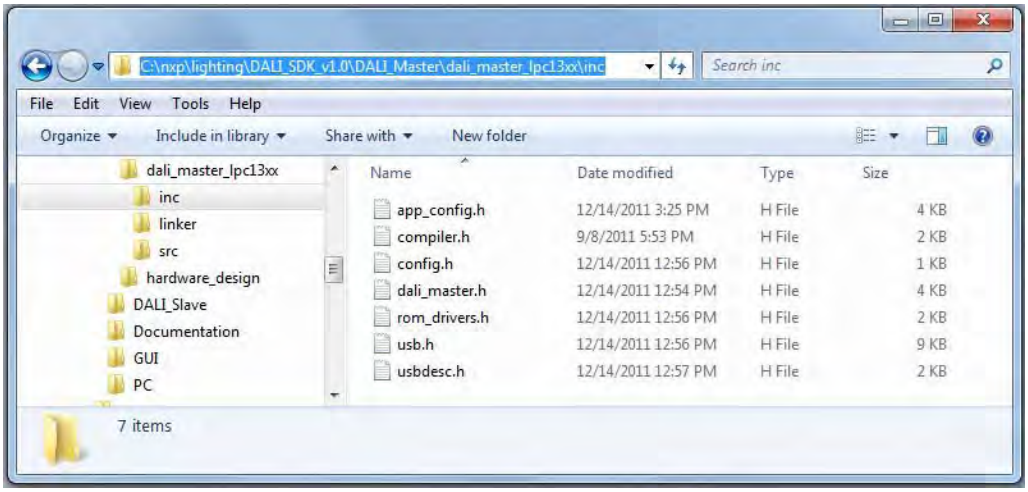

Fig 10. DALI master header files

A brief description of these source code files is shown in Table 3.

Table 3. DALI master source code file description

File name	Description
app_config.h	By means of this file the DALI master application can be configured
compiler.h	This file contains compiler specific definitions
config.h	This file contains USB HID device info configuration settings
cr_startup_lpc13.c	Assembly file containing the Reset vector and exception vector table (needed for the LPCXpresso build only)
cstartup_M.s	Assembly file containing the Reset vector and exception vector table (needed for the IAR build only)
dali_master.c	This file contains the main loop, and DALI bus plus host message handling
dali_master.h	This file contains the exported functions of dali_master.c
main.c	This file contains the application entry point
rom_drivers.h	This file contains setup structures for the USB HID driver in the boot ROM
usb.c	This file contains the glue layer to initialize this USB HID driver
usb.h	This file contains USB definitions
usbdesc.c	This file contains the USB descriptor
usbdesc.h	This file contains the USB descriptor definition

4.7 Building the software

The software tree includes project files for LPCXpresso v4.1.0_190.

When using LPCXpresso for building the DALI master, use workspace location <your install path>\DALI_Master and import the existing project at the location <your install path>\DALI_Master\dali_master_lpc13xx. Make sure to uncheck the "Copy projects into workspace" checkbox.

Next to LPCXpresso the software tree also includes project files for IAR Embedded Workbench for ARM v6.20.4. When the IAR workbench is installed, open the project by double clicking the file 'DALI_Master_LPC134x.eww'.

The software can be configured via the source file 'app_config.h', which holds one configuration option as listed in Table 4.

Table 4. DALI master configuration options

Define	Description
LPC_CORE_CLOCKSPEED_HZ	Sets the CPU clock.

The CPU clock is configured at 48 MHz (a multiple of 12 MHz), which gives more than enough performance needed for the DALI master.

Table 5 shows the firmware sizes (in bytes) of a RELEASE build of the DALI master.

Table 5. DALI master firmware sizes

Firmware sizes for DALI master	IAR EWARM v6.20.4		LPCXpresso v4.1.0_190	
	Flash [Bytes]	RAM [Bytes]	Flash [Bytes]	RAM [Bytes]
Release build	1884 + 108	124	1936	124

In the IAR IDE a program stack of 512 bytes is reserved in RAM (not shown in Table 5); in LPCXpresso this program stack size is not fixed. Further, the USB HID driver in the boot ROM needs 304 bytes in RAM (at a fixed location) which is also not shown in Table 5.

5. Document management

5.1 Abbreviations

Table 6. Abbreviations

Acronym	Description
CMSIS	Cortex Microcontroller Software Interface Standard
CPU	Central Processing Unit
CT	Counter Timer
GPIO	General Purpose Input/Output
HW	Hardware
IDE	Integrated Development Environment
IRQ	Interrupt Request
LED	Light Emitting Diode
MCU	Micro Controller Unit
NVM	Non Volatile Memory
PC	Personal Computer
PCB	Printed Circuit Board
PIO	Input/Output Pin
PLL	Phase Locked Loop
SW	Software
USB	Universal Serial Bus

5.2 Referenced documents

Table 7. Referenced documents

Title		Version	Author	Issue Date
[1]	IEC62386-102: Digital addressable lighting interface, General requirements – control gear	Edition 1.0 2009-6	IEC	2009
[2]	UM10375.pdf LPC1311/13/42/43 User manual	Rev. 01.02	NXP	20100330
[3]	UM10553 DALI Getting started guide	V1.0	NXP	2012

6. Legal information

6.1 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.

6.2 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. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

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 and its suppliers accept 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 competent 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.

6.3 Trademarks

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

7. List of figures

8. List of tables

9. Contents

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

Date of release: 1 March 2012
Document identifier: AN11175