UM10495

TDA5051A lighting master-slave demo board OM13314

Rev. 2 — 11 May 2012 User manual

Document information

Info	Content
Keywords	TDA5051A, LPC1114, PCF8883, zero crossing and UART style synchronization, lighting demo
Abstract	This document is a user manual for the TDA5051A Power Line Modem (PLM) master-slave lighting controller demo OM13314.

Revision history

Rev	Date	Description
v.2	20120511	User manual; second release. • Figure 4 “Schematic for 4 ´ PCF8883 capacitive proximity sensor switch” updated (to improve readability) • Figure 24 “TDA5051A lighting demo board schematic” updated (to improve readability)
v.1	20110816	User manual; initial release.

Contact information

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

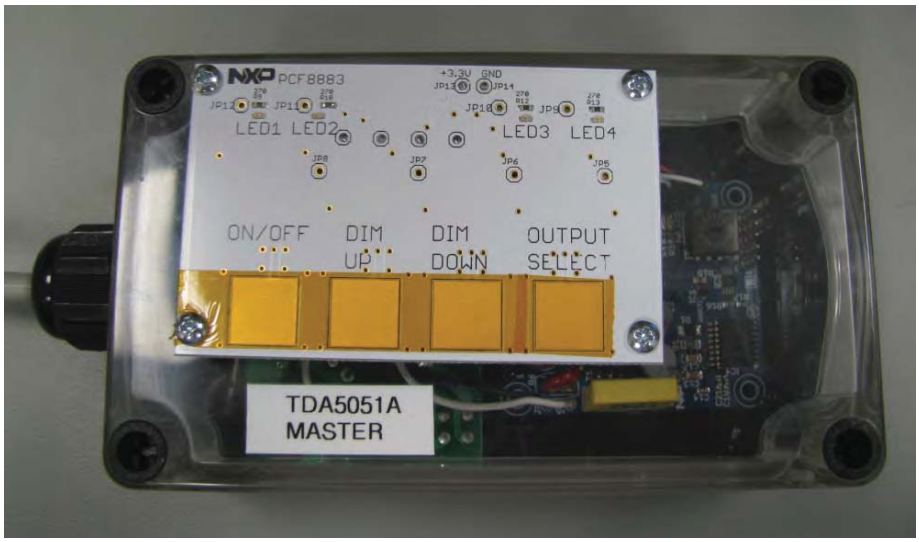
For sales office addresses, please send an email to: **salesaddresses@nxp.com**

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

1. Introduction

The TDA5051A lighting control demo consists of a master controller and a slave lighting controller. The master controller uses the High-Bay board, which consists of the LPC1114 microcontroller, the TDA5051A PLM IC, and power management circuitry. An AC-DC converter is used to provide +13 V DC supply voltage. The master employs 4 × PCF8883 capacitive proximity switches to provide On/Off, Dim Up, Dim Down, and Select outputs of the remote lighting slave controller. The slave controller consists of the High-Bay board, an AC-DC converter to provide +13 V supply voltage, and an LED array. Both the master and the slave are housed in a plastic box with 110 V AC power cords.



002aag521

Fig 1. TDA5051A master lighting controller

002aag522

Fig 2. TDA5051Aslave lighting controller

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

2. PLM demo board

The TDA5051A lighting demo board is designed to let customers evaluate the TDA5051A in a realistic application. The TDA5051A lighting demo board schematic is found in Section 12 “Appendix B — TDA5051A lighting demo board schematic”. The demo board includes a the TDA5051A, an LPC1114 microcontroller, a PCA9632 I^2C-bus to PWM converter, and software containing some pre-defined functions to brighten and dim the four available LED outputs from the PCA9632. The parameters used by these functions can be easily changed by changing a configuration header file with parameter values used by the pre-defined functions. To further customize the application, the driver functions used in the demo firmware can be easily modified.

2.1 Power line communications

The TDA5051A is an ASK modem chip designed for power line communications. In this application, the carrier frequency is 125 kHz. However, the frequency can be changed with the specified range by changing the frequency of the crystal or clock input. The TDA5051A consists of an AGC amplifier and ADC on the front with digital band-pass filtering and demodulation of the received signal. It also contains a lookup ROM and DAC to provide the proper wave shape and envelope for transmitted data. The internal amplifier then drives an external network to couple the transmit data onto the AC lines.

2.2 Demo application overview

The TDA5051A demo board is designed for easy setup and ease of operation. The demo functions are executed by pressing one of four control buttons located externally to the demo board. TDA5051A demo functions supported by the demo board and firmware include the following:

Off/On — Sends an On/Off command to the slave via the PLM

Dim Up — Sends a Dim Up command to the slave via the PLM

Dim Down — Sends a Dim Down command to the slave via the PLM

Select — Rotates the selected channel from the following:

- GROUP
- PWM0
- PWM1
- PWM2
- PWM3

Some of the other features of the TDA5051A demo board are:

- Programmable device address (slave configuration)
- Requires 12 V input, has onboard +5 V and +3.3 V regulators
- Flexible re-programming, firmware updating
  - SWD
  - RS-232

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

This document will also discuss the firmware required to implement the demos. One set of firmware was written utilizing the zero crossing of the AC line as a method to synchronize transmission. The other set of firmware was written to use a start bit/stop bit synchronization similar to a UART to allow transmission over DC lines that have no zero crossing information.

3. Hardware requirements

The hardware requirements for both firmware implementations are similar when using the AC line as transmission medium. Due to the synchronization differences of the two different firmware implementations, the UART style would not need the opto-coupler that is on the TDA5051A demo board. Considerations for other implementations will be discussed in a later section. Even though the same demo board is used for both the master and the slave, not all components are used in both configurations as discussed later in Section 4 and Section 5.

3.1 Master hardware configuration

To implement the master the hex address switch SW1 is not needed. If SW1 is populated, set the switch to the '0' position (all 'open'). This allows four external switches to be connected to the same pins on the microcontroller as SW1 without generating a conflict. The external switches mate with the SV2 connector on the demo board. A DC power supply and connection to the AC line are also needed for the master. Additional details can be found in Section 4.

3.2 Slave configuration

To implement the slave for the SSL demo, the TDA5051A demo board, four LEDs with current limiting resistors, an AC line connection, and a DC power input are required to implement a PLM slave for the SSL demo. The slave actions in response to received commands are discussed in Section 5.

4. Master hardware description — TDA5051A demo board

In the master configuration, the TDA5051A demo board reads switch closures and generates commands that are transmitted by the TDA5051A over the AC lines. The details of how the LPC1114 handles these tasks will be discussed in the firmware portion of the Users Manual.

4.1 TDA5051A demo board — general

The NXP TDA5051A and the LPC1114 are the heart of the board with the TDA5051A providing the modem interface to the power lines and the LPC1114 providing a high-performance programmable processor at low power and low cost. It handles all of the 'intelligent' functions of the board and manages the peripheral interfaces.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

4.2 TDA5051A demo board — I/O, master configuration

The TDA5051A demo board provides a number of switches and visual indicators to control and monitor the Power Line communications. The details on how these are used in the master configuration are discussed in the following sections.

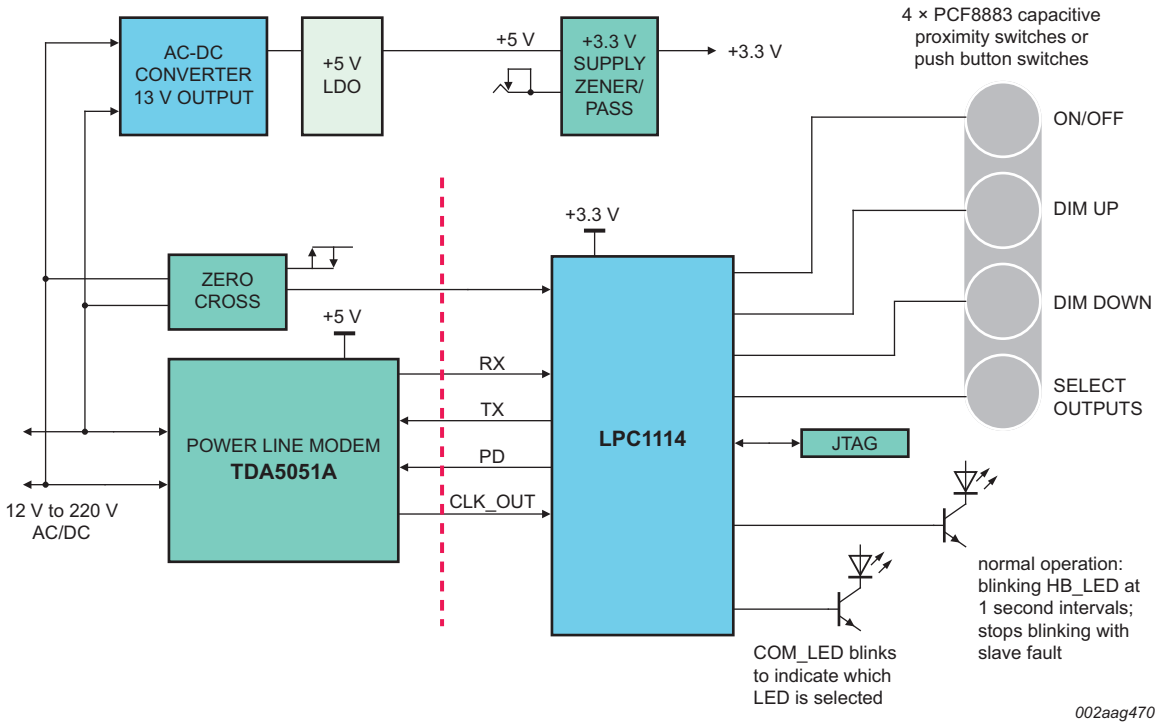


Fig 3. Block diagram of master lighting controller

4.2.1 Onboard visual indicators

Table 1 shows the onboard LEDs and their function in the master configuration.

Table 1. Onboard LEDs and function

LED label	LED function	Control drive source	Notes
HB_LED	processor heartbeat	LPC1114 - PIO3_2	blinks when operational
COM_LED	Com status	LPC1114 - PIO3_4	blinks when 'select' is pressed[1]

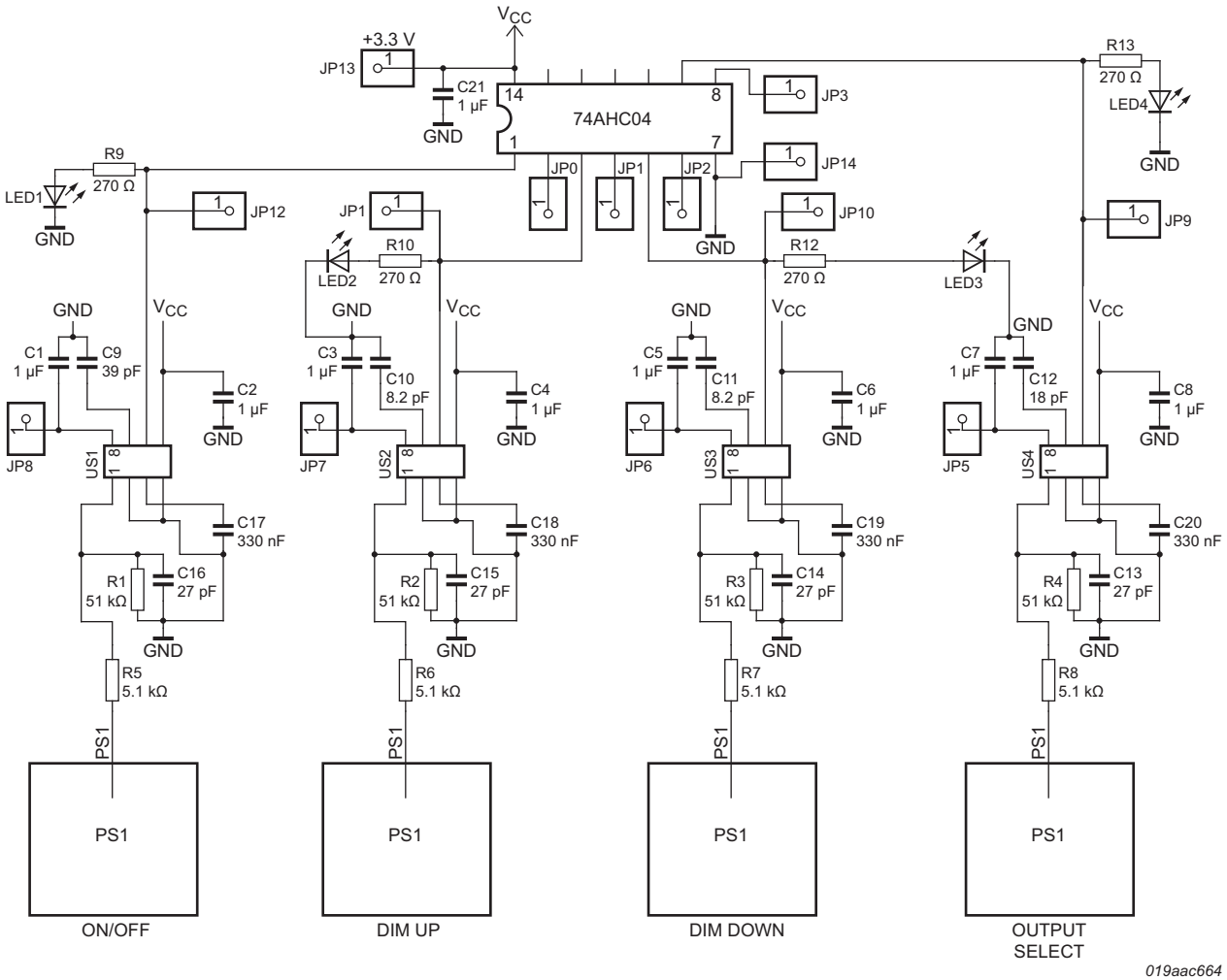
[1] When pressing the external 'select' switch, the Com LED will blink to indicate which item is being selected as detailed in Section 4.2.2.

4.2.2 External switch interface

The master configuration requires four external switches to be connected to header SV2. This may be implemented with a 4 × PCF8883 capacitive proximity sensor switch circuit shown in Figure 4. The connector pin assignments are shown in Table 2.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.



(1) JP0 connects to pin 1 of SV2 header.

(2) JP1 connects to pin 2 of SV2 header.

(3) JP2 connects to pin 3 of SV2 header.

(4) JP3 connects to pin 4 of SV2 header.

(5) GND (ground) connects to pin 5 of SV2 header.

(6) V_{CC} connects to +3.3 V supply.

Fig 4. Schematic for 4 × PCF8883 capacitive proximity sensor switch

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

Table 2. SV2 header pin assignments

Header pin	Switch function	Signal destination	Notes
1	Off/On	LPC1343 - PIO1_8	sends an On/Off command
2	Dim Up	LPC1343 - PIO1_9	sends a Dim Up command
3	Dim Down	LPC1343 - PIO1_10	sends a Dim Down command
4	Select	LPC1343 - PIO1_11	rotates the selected channel[1]
5	-	ground	

[1] Rotates the selected channel from the following and blinks the COM_LED when changed as noted below:
GROUP → 1 Com LED long blink
PWM0 → 1 Com LED short blink
PWM1 → 2 Com LED short blinks
PWM2 → 3 Com LED short blinks
PWM3 → 4 Com LED short blinks

4.2.3 Demo board connections — master configuration

A TDA5051A demo board, four switches, an AC line connection, and a DC power input are required to implement a master for the SSL demo. The functions assigned to the switch inputs are discussed in a later section.

The interface connections to the TDA5051A demo board are shown in Figure 5.

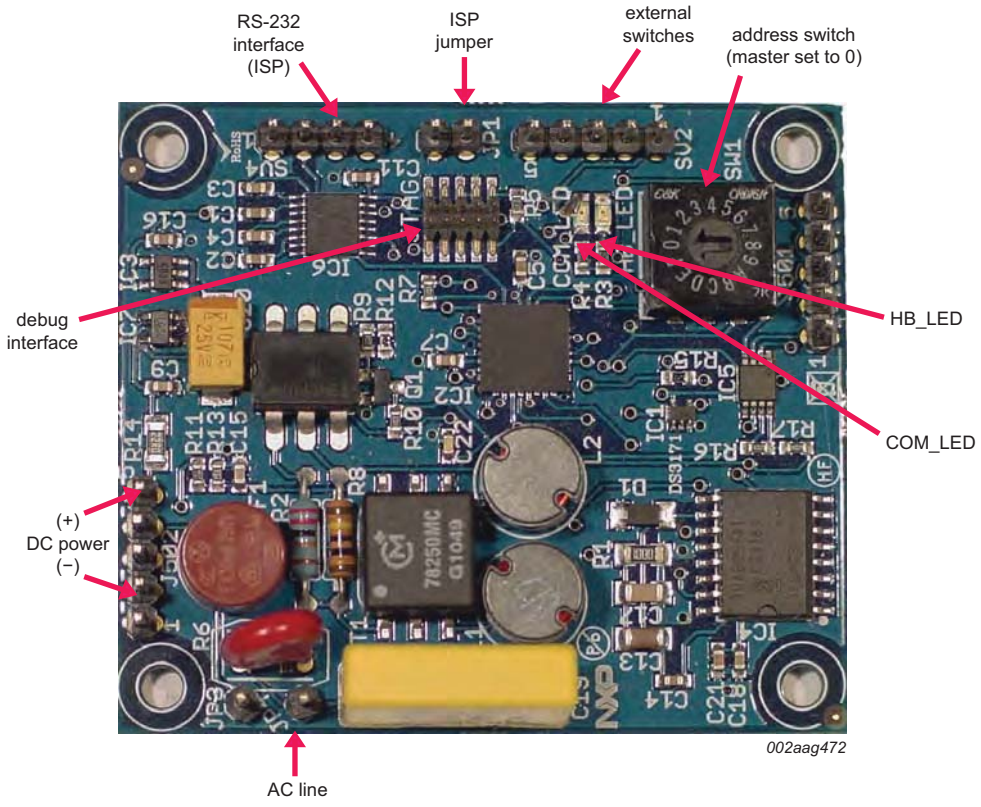


Fig 5. TDA5051A demo board interface connections

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

5. TDA5051A demo board — slave configuration

In the slave configuration, one of the tasks of the LPC1114 microcontroller is to detect when a command has been received over the power lines and execute the command. In the demo firmware provided, the commands received over the power lines result in sending an I^2C message to the PCA9632 LED PWM controller to control the output brightness of four externally connected LEDs.

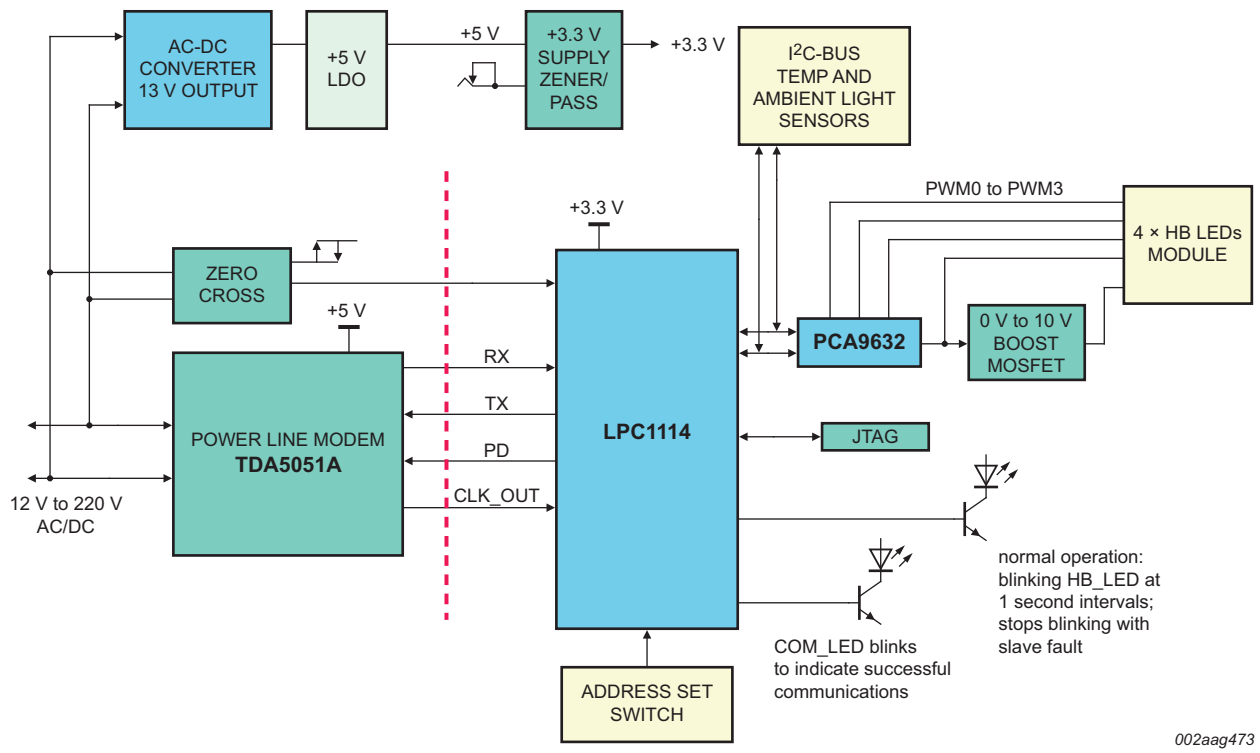


Fig 6. Block diagram of slave lighting controller

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

5.1 TDA5051A demo board — I/O, slave configuration

The TDA5051A demo board provides a number of switches and visual indicators to control and monitor the Power Line communications. The details on how these are used in the slave configuration are discussed in the following sections.

5.1.1 Onboard visual indicators — slave configuration

Table 3 shows the onboard LEDs and their function in the slave configuration.

Table 3. Onboard LEDs, function in slave configuration

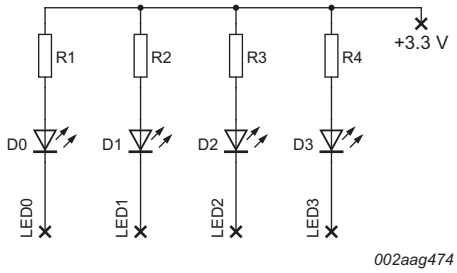
LED label	LED function	Control drive source	Notes
HB_LED	processor heartbeat	LPC1114 - PIO3_2	blinks when operational
COM_LED	Com status	LPC1114 - PIO3_4	blinks when communicating

5.1.2 External LED interface

The LED outputs from the PCA9629 can drive LEDs directly, up to a specified limit, or it can provide PWM controlled dimming for external high current LED power supplies. In the demo application, an external array of four white, low power LEDs is driven from the PCA9632. The LEDS are biased with current limiting resistors and an external connection to a power supply on the demo board (see the schematic in Figure 7). The connection to the external LEDs is via header J501 on the TDA5051A demo board with the pin assignments is given in Table 4.

Table 4. Pin assignments for external LED connection via header J501

Header pin	Function	Signal destination	Notes
1	PWM3	LED3	via external CL resistor to +3.3 V
2	GND	supply GND	
3	PWM2	LED2	via external CL resistor to +3.3 V
4	PWM1	LED1	via external CL resistor to +3.3 V
5	PWM0	LED0	via external CL resistor to +3.3 V



X = connections to 3.3 V supply and header J501 on TDA5051A demo board.

Fig 7. Schematic for external array of four white LEDs

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

5.1.3 Demo board connections — slave configuration

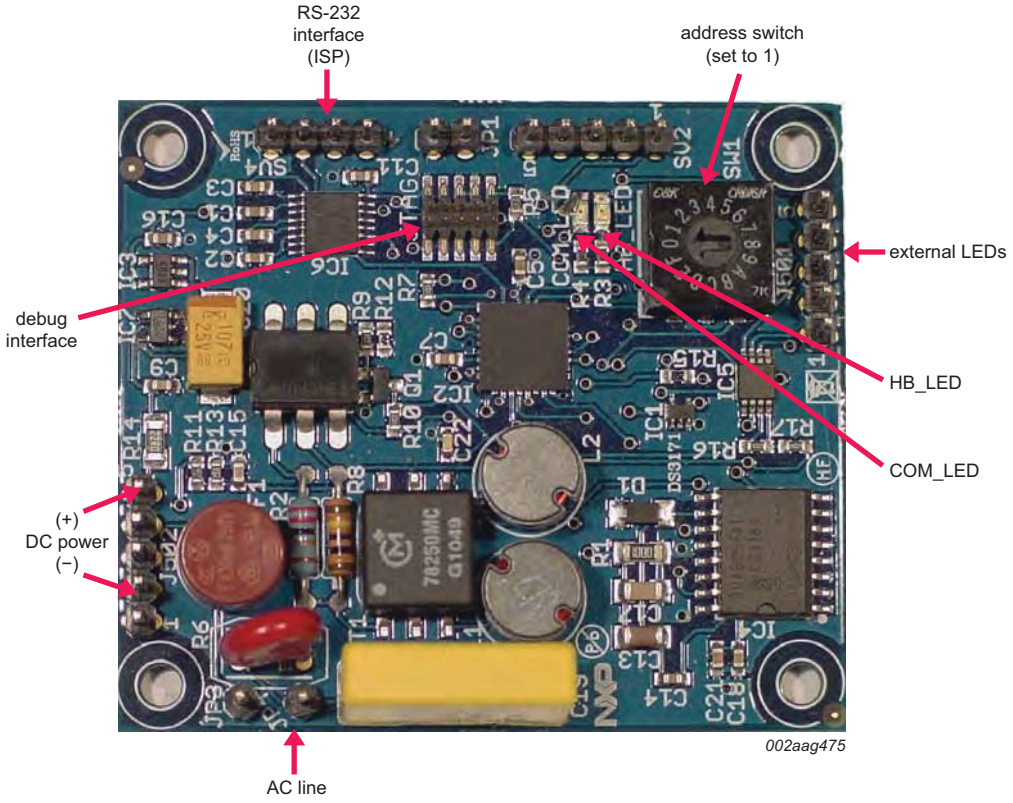


Fig 8. TDA5051A demo board connections for slave configuration

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

6. Protocol requirements

The objective of this demo software is to provide a protocol structure that defines the bit level frames and message frames to provide a number of capabilities including the following:

- Source and destination addresses allowing multiple masters and multiple slaves
- Parity and checksum error detection and message re-transmission to increase reliability of communications
- Sub-address to communicate with multiple channels within a fixture
- Bidirectional communications to allow getting status or configuration information from slaves

The message level protocol for implementations using zero crossing or start bit synchronization is the same, however the bit level frames are different as discussed in the next section.

6.1 Bit frame protocol considerations — zero crossing synchronization

By using the zero crossing of the AC signal to synchronize the bit frame, we can structure the protocol to avoid the peaks of the AC line where the noise is greatest and the line impedance is the lowest. Most low power factor equipment draws the most current at the peaks. As a result, the line impedance may drop significantly when the rectifier diodes in the equipment are conducting at the peaks of the AC line. This can affect the amount of signal coupled onto the AC line potentially causing errors. By having the bit frame start 1 ms after the peak of the AC sine wave and ending the bit frame 1 ms before the AC peak, the noisy, low-impedance portion of the AC line can be avoided, increasing the reliability of the communications link. For received data, 16× sample clock is generated and five samples are taken in the middle of the bit time. The received state of the sampled bit is determined by the majority of the five samples taken. This helps reduce reception errors due to short noise spikes that may occur at a sample time, but is averaged out over several samples. The bit frame for the zero crossing implementation will be detailed in Section 7.

6.2 Bit frame protocol considerations — start bit (UART style) synchronization

For some applications such as DC links (solar panels, DC power distribution), a zero crossing signal is not available. Also, when coupling between different phases, the zero crossing time of the master is not coincident with the zero crossing of the slave. For these applications, firmware was developed that uses a start bit to synchronize the reception of the data so a zero crossing is not required. After the start bit, a frame bit is needed to provide message level synchronization. The frame bit is then followed by eight data bits, a parity bit and a stop bit. In the same way as was done with the zero crossing firmware, a 16× receive clock is generated and five samples taken in the middle of the bit time. The received state of the sampled bit is determined by the majority of the five samples taken to increase noise immunity as noted in the previous section.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

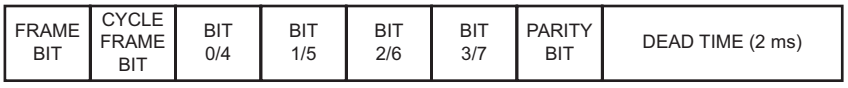
7. Protocol specification

Two sets of firmware were generated for the TDA5051A demo board for the two different synchronization methods discussed in the previous section. Each set includes master and slave firmware. The following protocol specifications were generated based on the requirements listed in Section 6.

7.1 Protocol specification — zero crossing synchronization

7.1.1 Bit level protocol — zero crossing synchronization

In transmitting over the AC lines, there is a region approximately 2 ms wide centered on the peaks of the AC sine wave where line disturbances have the greatest chance of corrupting the data transmission and reception. One of the line disturbances is due to added noise from any number of sources that are active during this time. With low power factor power supplies, the rectifiers only conduct at the peaks of the AC signal. This can result in lowering of the line impedance during this period. This can lower the amount of signal coupled on to the line by the master as well as reducing the amount of signal received by the slave. When synchronizing to the zero crossing, the bit level transmission frame can be offset from the detected zero crossing. The protocol then specifies 2 ms of dead time centered over the peaks of the AC line to avoid the ‘disturbance region’ of the AC line. At 60 Hz, a half-cycle of the AC line is 8.33 ms in duration. Avoiding the 2 ms disturbance region leaves 6.33 ms to transmit the bit frame. This does not leave enough time to transmit a byte of data (with framing and parity bits). As a result, the half-byte is transmitted every half-cycle. The start of the frame is offset from the zero crossing by about 5.16 ms (about 1 ms after the AC peak). The format of the bit level frame is detailed in Figure 9.



If the frame bit is 0, it indicates the following bits should contain a valid frame byte flagging the beginning of a message frame. When the Frame Bit is 1, the bits [0:3] during the first half cycle and bits [4:7] in the second half cycle should contain a valid frame byte. The value of the parity bit is then calculated including the frame bit and bits [0:3] or [4:7] depending on the half-cycle indicated by the cycle frame bit.

If the frame bit is 1, and a frame byte has already been received, this indicates the following bits should contain a data byte or checksum byte depending on the position after the frame bit. The value of the parity bit is then calculated including the frame bit and bits [0:3] or [4:7] depending on the half-cycle indicated by the cycle frame bit.

If the cycle frame bit is ‘0’, this indicates this frame corresponds to the first half-cycle of the transmitted data and contains data/frame bits [0:3].

If the cycle frame bit is ‘1’, this indicates this frame corresponds to the second half-cycle of the transmitted data and contains data/frame bits [4:7].

The dead time allows time for the slave to process the current data transmitted as well as allowing the master and the slave to avoid transmitting during the peaks of the AC line. At 60 Hz, the time of a half-cycle is 8.33 ms. Providing 2 ms of dead time leaves 6.33 ms to transmit 7 bits, resulting in a data rate of 1105 bit/s.

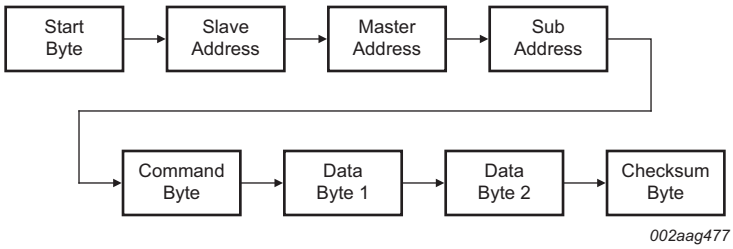
Fig 9. Bit level frame format

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

7.1.2 Message level protocol

The message level protocol was designed to provide a number of features. The message structure has the ability work with multiple masters and multiple slave by having a master and slave address included in the frame. Error handling is also included by providing a message checksum along with the frame by frame parity to detect errors. The message level protocol is detailed in Figure 10 starting with the master transmit message frame.



The Start Byte is a specific value for a master transmission and indicates the beginning of a message.

The slave address is the destination and for lighting applications is typically a lighting fixture. It could also potentially address a bridge device such as a PLM/DALI bridge, or PLM/RS-232 bridge to allow communications between protocols.

The master address is present to support multi masters. For example, for home lighting, there may be masters in every room. So, a slave could be programmed to only respond to the master in the room it is located in. It could also respond to broadcast command to turn off or dim lights at certain hours, or when a residence is vacated for energy management.

The sub address could be a specific light or color in the lighting fixture. Or, in the case of a Dali Bridge device, this could be the Dali address byte.

The command byte tells the slave how to respond. The proposed protocol uses similar command assignments as Dali. In a Dali bridge device, this would be the Dali command. If similar command assignments are used in the PLM protocol, no command translation would be required in a Dali bridge device.

Data Byte1 / Data Byte 2 are available if the command indicates the slave should use the data that follows. Depending on the specified command, the data bytes can be interpreted as two 8-bit values or one 16-bit value. In a PLM/RS-232 bridge device, this number of bytes could be expanded to allow for more data bytes if needed. Larger frames are not intended to be supported in the initial implementation for the lighting application.

The checksum byte contains the two's complement checksum of the previous seven bytes. The slave should sum the start byte and all remaining bytes including the checksum. If the result is '0', the frame has been received without a checksum error.

Fig 10. Master transmit message frame

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

The slave will respond to the master if the command is received with a valid start byte.

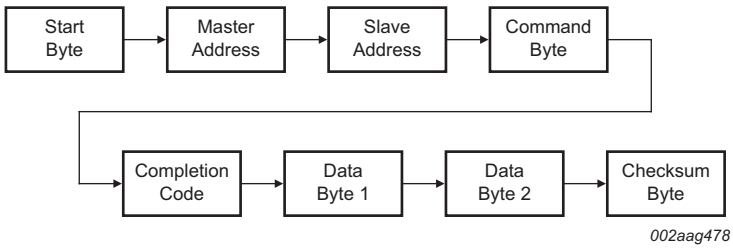


Fig 11. Slave receive message frame

A completion code of '0' indicates the master's command was received with parity bits correct, the checksum correct, a valid start byte, and the correct number of bytes received. Completion codes could be assigned to parity or checksum errors so statistics can be obtained for these if desired. Other completion codes for special error conditions could also be defined if needed. If the master receives no response, the master will transmit the message up to three times (configurable) before aborting the attempt. The data bytes in the response frame could provide the ability to query the slave about its status, configuration, etc. Even though the frame structure could support slave queries, the application code described here only implements basic communications and command functions to demonstrate capabilities while leaving enough flexibility to allow the user to enhance the code as needed for their own applications.

7.2 Protocol specification — start bit (UART style) synchronization

7.2.1 Bit level protocol — start bit (UART style) synchronization

There are applications where zero crossing synchronization is not feasible. This specification was developed and firmware was written to address those situations. A start bit transition and additional data samples are used to synchronize the reception of the data so a zero crossing is not required. After the start bit, a frame bit is used to provide message level synchronization. The frame bit is then followed by eight data bits, a parity bit and a stop bit. In the same way as was done with the zero crossing firmware, a 16× receive clock is generated and five samples are taken of all bit in the middle of the bit period. The received state of the sampled bit is determined by the majority of the five samples taken to increase noise immunity as noted in the previous section.

Synchronization at the bit level is performed in the same manner as a UART. The idle state of the 'soft' PLM UART RX and TX pins on the microcontroller is 'HIGH'. Since this matches the TDA5051A idle state, we can directly connect the TX pin on the microcontroller to the $\overline{\text{DATA\_IN}}$ pin on the TDA5051A. And, likewise, we can connect the 'soft' PLM UART RX pin on the micro to the $\overline{\text{DATA\_OUT}}$ pin on the TDA5051A. So the 'HIGH' and 'LOW' states referred to in the specification pertain to the logic levels of the interface between the microcontroller and the TDA5051A. The bit level frame is shown in Figure 12.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

The Start bit is always LOW. This is used as synchronization as the stop bit and any idle time after the stop bit are defined to be HIGH. As a result, there is always a HIGH-to-LOW transition at the beginning of a bit frame to synchronize the reception of the rest of the bits in the bit frame. This means the master transmit and slave receive baud rates must be set equal to be able to sample the received signal at the correct times. In addition to using the 'HIGH-to-LOW' transition to synchronize the receive sampling, the start bit is also sampled half a bit period after the 'HIGH' to 'LOW' transition. This is to verify a valid start condition exists.

If the frame bit is 0, it indicates the following bits should contain a valid frame byte flagging the beginning of a message frame. When the Frame Bit is 1, the bits [0:7] in the field should contain a valid frame byte. The value of the parity bit is then calculated including the frame bit and bits [0:7].

If the frame bit is 1, it indicates the following bits should contain a data byte or checksum byte depending on its position after the frame bit. The value of the parity bit is then calculated including bits [0:7].

The Stop bit is always HIGH, providing time for the slave to process the current data transmitted and to provide at least 1 bit period of idle state to facilitate detection of the next Start bit.

Fig 12. Bit level frame

8. Master firmware overview

To implement the Master application, three commands were defined and enumerated: Dim Up, Dim Down, and On/Off. These commands are initiated by pressing one of three external switches or touch sensors (active LOW) that are connected to Port pins P1_8, P1_9, P1_10 of the LPC1114 microcontroller. The fourth switch connected to P1_11 and will rotate through the different control channels available and set the selected channel number in the outgoing PLM transmit frame. The details of the protocol and the command and response frame structure are discussed later.

P1_8: Off/On — Sends an On/Off command to the slave via the PLM

P1_9: Dim Up — Sends a Dim Up command to the slave via the PLM

P1_10: Dim Down — Sends a Dim Down command to the slave via the PLM

P1_11: Select — Rotates the selected channel from the following:

- GROUP
- PWM0
- PWM1
- PWM2
- PWM3

More details of how these commands are interpreted will follow in the Slave section (Section 9).

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

There are two LED indicators on the TDA5051A used by the master. The ‘HB LED’ is for the Heartbeat function to indicate that the microcontroller is functioning when blinking. The ‘COM LED’ indicates which of the functions is selected when the select switch is activated by the following indications:

One short blink – PWM0 is selected

Two short blinks – PWM1 is selected

Three short blinks – PWM2 is selected

Four short blinks – PWM3 is selected

One long blink – GROUP is selected

The Master transmission contains a slave address to select which device will recognize its commands. Since there are no additional inputs on the TDA5051A demo board to set this address, it is ‘hard coded’ to be 0x01.

8.1 Master firmware implementation — zero crossing synchronization

The master firmware for the TDA5051A SSL demo application contains a number of ‘C’ application modules combined with system and core code. The modules are described in Section 8.1.1 through Section 8.1.12.

8.1.1 Main_loop.c

This module calls all the initialization code for all the other modules and provides the While (1) loop to test for application flags. The flags tested by the master include byte level and message level receive flags, transmit status, and transmit error flags. If the flag gets set, the test function will call application function in the related module.

8.1.2 16b_timer1.c

Timer 16b1 on the LPC1114 provides the receive sample timing and is set to interrupt at a rate 16× transmit data rate. The middle five samples of each bit time are used to determine the logic state of the received bit by state of a majority of the samples. The start of the sample time is determined by an offset from the zero crossing.

8.1.3 UART.c

This was used as a debug channel during development.

8.1.4 IO.c

This configures the I/O of the LPC1114 including setting the functions and direction of the I/O pins. The I/O interfacing with the zero crossing opto couple generates an interrupt on both edges of the signal. The pulse width of the optocoupler output is measured to estimate the zero crossing as described later. This module also contains the functions to detect the closure of a command switch and take the appropriate action or sent the corresponding command to the slave.

8.1.5 Command.c

This was used for initial debug.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

8.1.6 NVIC.c

This sets the priority and enables required interrupts in the Nested Vectored Interrupt Controller (NVIC). The enable interrupts include timers 32b0, 32b1, 16b0, 16b1, and external interrupt EINT0.

8.1.7 SysTick.c

The SysTick timer provides the heartbeat and delay function timing. The transmit time-out counter is also implemented using the systick interrupt.

8.1.8 32b_Timer0.c

The 32b0 timer provides the transmit timing for the master transmitter. This is a 1× clock synchronized by an offset from the zero crossing. This provides a timing reference for the output data.

8.1.9 PLM_Master.c

This handles the message level communications for the PLM master, providing synchronization, transmitting commands, receiving acknowledgements, and handling error conditions. This module also contains the External interrupt handler that is active on both edges of the zero crossing opto-coupler. This is used to determine the estimated zero crossing using timer 32b1 as described in Section 8.1.10.

8.1.10 32b_Timer1.c

This is used to time the start of frame offset from the zero crossing. Part of this involves timing the pulse width of the zero crossing pulse from the opto-coupler. Since the opto-coupler output is symmetrical around the actual zero crossing, the time from the second edge of opto-coupler pulse to the actual zero crossing can be estimated by dividing the pulse width by two. This is subtracted from the number of counts required for the 5.16 ms offset and loaded into timer 32b1 to set the offset time from the second edge. The timer 32b1 interrupt then sets the start time of the offset bit frame.

8.1.11 Timer16b_0.c

The master uses this as the debounce timer for the switch inputs.

8.1.12 PLM_UART.c

This contains the routines that handle the bit level synchronization and data transmission and reception. This uses the timing references from the 16b1 and 32b2 timers to implement a soft UART specifically for the interfacing to the TDA5051A.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

8.2 Master firmware implementation — start bit (UART style) synchronization

The Master configuration for implementing the start bit synchronization method is the same as the zero crossing method at the high level. The switch commands and indicator actions are the same for the firmware implementations of both synchronization methods. The differences discussed in the protocol section are detailed when discussing how the synchronization methods are handled in the individual firmware modules.

The master firmware for the TDA5051A SSL demo application contains a number of 'C' application modules combined with system and core code. The modules are described in Section 8.2.1 through Section 8.2.12.

8.2.1 Main_loop.c

This module calls all the initialization code for all the other modules and provides the While (1) loop to test for application flags. The flags tested by the master include byte level and message level receive flags, transmit status flag to see if all bytes in message have been transmitted, and transmit error flag. If the flag gets set, the test function will call an application function in the related module to act.

8.2.2 16b_timer1.c

Timer 16b1 on the LPC1114 provides the receive sample timing and is set to interrupt at a rate 16× transmit data rate. The middle five samples of each bit time are used to determine the logic state of the received bit by state of a majority of the samples. The start of the sample time is determined by an offset from the zero crossing.

8.2.3 UART.c

This is only used for debug.

8.2.4 IO.c

This configures the I/O of the LPC1114 including setting the functions and direction of the I/O pins. This also configures the RX pin used by the PLM UART as an external interrupt that detects the HIGH-to-LOW transition at the beginning of the start bit. This module also contains the functions to detect the closure of a command switch and take the appropriate action or sent the corresponding command to the slave.

8.2.5 Command.c

This is only used for debug.

8.2.6 NVIC.c

This sets the priority and enables required interrupts in the Nested Vectored Interrupt Controller (NVIC). The enabled interrupts include Timer 32b0, Timer 16b1, and the external interrupt.

8.2.7 SysTick.c

The SysTick timer provides the heartbeat and delay function timing.

8.2.8 32b_Timer0.c

The 32b0 timer provides the transmit timing for the master transmitter.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

8.2.9 PLM_Master.c

This handles the message level communications for the PLM master, providing synchronization, transmitting commands, receiving acknowledgements, and handling error conditions. This module also contains the External interrupt handler that detects the start bit for bit level synchronization.

8.2.10 32b_Timer1.c

This is used to time the start of frame offset from the zero crossing. Part of this involves timing the pulse width of the zero crossing pulse from the opto-coupler. Since the opto-coupler output is symmetrical around the actual zero crossing, the time from the second edge of opto-coupler pulse to the actual zero crossing can be estimated by dividing the pulse width by two. This is subtracted from the number of counts required for the 5.16 ms offset and loaded into timer 32b1 to set the offset time from the second edge. The timer 32b1 interrupt then sets the start time of the offset bit frame.

8.2.11 Timer16b_0.c

The master uses this as the debounce timer for the control switch inputs.

8.2.12 PLM_UART.c

This contains the routines that handle the bit level synchronization, data transmission, and reception. This uses the timing references from the 16b1 and 32b2 timers to implement a soft UART specifically for the interfacing to the TDA5051A.

9. Slave firmware description

To implement the Slave application, an on-board hex switch is used to set the slave address. This is connected to port pins P1_8, P1_9, P1_10, and P1_11. The address assigned to the hex switch is specified as follows:

P1_8: 2^8

P1_9: 2^4

P1_10: 2^2

P1_11: 2^1

The sum of the addresses for the logic ‘HIGH’ inputs results in the slave address recognized for commands. For example, when the hex switch is set to ‘1’, P1_11 will be ‘HIGH’ and P1_8 – P1_10 will be ‘LOW’.

Remark: Since the master hard codes a slave address of ‘0x01’, make sure the Hex switch on the slave is set to this address to be able to respond to master commands.

There are two LED indicators on the Slave. The ‘HB LED’ is for the Heartbeat function to indicate that the microcontroller is functioning. The ‘COM LED’ blinks when a valid frame byte has been received to indicate communications activity.

The Slave configuration for implementing the start bit synchronization method is the same as the zero crossing method at the high level. The command responses and indicator actions are the same for the firmware implementations of both synchronization methods. The differences are detailed in the individual module sections when discussing how the different synchronization methods are handled.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

9.1 Slave firmware description — zero crossing synchronization

The slave firmware for the TDA5051A SSL demo application contains a number of 'C' application modules combined with system and core code. The modules are described in Section 9.1.1 through Section 9.1.14.

9.1.1 Main_loop.c

This module calls all the initialization code for all the other modules and provides the While (1) loop to test for application flags. The flags tested by the master include byte level and message level receive flags, transmit status flag to see if all bytes in message have been transmitted, and transmit response flag. If the flag gets set, the test function will call an application function in the related module to act.

9.1.2 16b_timer1.c

Timer 16b1 on the LPC1114 provides the receive sample timing and is set to interrupt at a rate 16× transmit data rate. The middle five samples of each bit time are used to determine the logic state of the received bit by state of a majority of the samples.

9.1.3 UART.c

This was used for initial debug.

9.1.4 IO.c

This configures the I/O of the LPC1114 including setting the functions and direction of the I/O pins. The I/O interfacing with the zero crossing opto-coupler also generates an interrupt on both edges of the signal. This pulse width of the opto-coupler output is measured to estimate the zero crossing as described later.

9.1.5 Command.c

This is only used for debug.

9.1.6 NVIC.c

This sets the priority and enables required interrupts in the Nested Vectored Interrupt Controller (NVIC). The enable interrupts include timers 32b0, 32b1, 16b0, 16b1, and external interrupt EINT0.

9.1.7 SysTick.c

The SysTick timer provides the heartbeat and delay function timing.

9.1.8 32b_Timer0.c

The 32b0 timer provides transmit timing allowing the slave to respond back to a master.

9.1.9 I2C.c

This contains the driver code for the I^2C-bus peripheral on the LPC1114. The I^2C interface is used by the slave to communicate with the PCA9632 I^2C-bus to PWM LED driver. The application specific code is contained in the 'I2C Messages.c' module described in Section 9.1.10.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

9.1.10 I2C Messages.c

This module contains the functions that interpret the incoming commands and channel assignments and sends corresponding I^2C messages to the PCA9632. The PCA9632 is initialized in the 'Group Dim Mode' where a 190 Hz fixed frequency 'Group' signal is superimposed with the 6.25 kHz Individual brightness control signals PWM0, PWM1, PWM2, and PWM3. The 'Select' function on the master sets the outgoing channel number in the PLM Master transmit frame. This channel number is read by the slave to determine which register on the PCA9632 is addressed when it sends out I^2C commands. The corresponding registers on the PCA9632 and a short description are shown in Table 5.

Table 5. PCA9632 corresponding registers

Register	Range of values	Description
GRPPWM	0 to 15	this is used as a global brightness control allowing the LED outputs to be dimmed with the same value
PWM0	0 to 63	Individual Brightness control for PWM0 output
PWM1	0 to 63	Individual Brightness control for PWM1 output
PWM2	0 to 63	Individual Brightness control for PWM2 output
PWM3	0 to 63	Individual Brightness control for PWM3 output

This mode is useful to be able to set the brightness 'mix' of the different outputs, then increasing or decreasing all outputs at the same time while still maintaining the same 'mix'. Refer to the PCA9632 data sheet (Ref. 1) for more information. The brightness levels of both the GRPPWM and PWMx registers are initialized at a mid level of brightness. To get maximum duty cycle from the outputs, increase both group and individual levels via the master 'Dim Up' and 'Dim Down' and 'Select'. The Off/On command from the master results in the slave sending an I^2C message to the PCA9632 to enable or disable all or none of the LEDx output drives via the LEDOUT register on the PCA9632. The actual levels sent to the GRPPWM and PWMx registers are incremented via functions that provide a simple piece-wise linear approximation of a logarithmic dimming curve. As the eye sensitivity is logarithmic, providing some approximation of this gives the Dim Up and Dim Down commands a more natural response. The dimming curve functions are located in the PLM_Slave module discussed in Section 9.1.11.

9.1.11 PLM_Slave.c

This handles the message level communications for the PLM slave, providing synchronization, receiving commands, transmitting acknowledgements, and handling error conditions. The received 'Dim Up' and 'Dim Down' commands result in PWM values that are incremented or decremented through simple dimming curve functions contained in this module as discussed in the previous section. This module also contains the External interrupt handler that is active on both edges of the zero crossing opto-coupler. This is used to determine the estimated zero crossing using timer 32b1 as described in Section 9.1.12.

9.1.12 32b_Timer1.c

This is used to time the start of frame offset from the zero crossing in the same manner as the master. Part of this involves timing the pulse width of the zero crossing pulse from the opto-coupler. Since the opto-coupler output is symmetrical around the actual zero crossing, the time from the second edge of opto-coupler pulse to the actual zero crossing can be estimated by dividing the pulse width by two. This is subtracted from the number of

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

counts required for the 5.16 ms offset and loaded into timer 32b1 to set the offset time from the second edge. The timer 32b1 interrupt then sets the start time of the offset bit frame.

9.1.13 Timer16b_0.c

The master uses this as the debounce timer for the switch inputs.

9.1.14 PLM_UART.c

This contains the routines that handle the bit level synchronization, data transmission, and reception. This uses the timing references from the 16b1 and 32b2 timers to implement a soft UART specifically for the interfacing to the TDA5051A.

9.2 Slave firmware description — start bit (UART style) synchronization

The slave firmware for the TDA5051A SSL demo application contains a number of 'C' application modules combined with system and core code. The modules implemented for start bit synchronization are described in Section 9.2.1 through Section 9.2.12.

9.2.1 Main_loop.c

This module calls all the initialization code for all the other modules and provides the While (1) loop to test for application flags. The flags tested by the master include byte level and message level receive flags, transmit status flag to see if all bytes in message have been transmitted, and transmit response flag. If the flag gets set, the test function will call an application function in the related module to act.

9.2.2 16b_timer1.c

Timer 16b1 on the LPC1114 provides the receive sample timing and is set to interrupt at a rate 16× transmit data rate. The middle five samples of each bit time are used to determine the logic state of the received bit by state of a majority of the samples.

9.2.3 UART.c

This is only used for debug.

9.2.4 IO.c

This configures the I/O of the LPC1114 including setting the functions and direction of the I/O pins. This also configures the RX pin used by the PLM UART as an external interrupt that detects the HIGH-to-LOW transition at the beginning of the start bit This module also contains the functions to detect the closure of a command switch and take the appropriate action or sent the corresponding command to the slave.

9.2.5 Command.c

This is only used for debug.

9.2.6 NVIC.c

This sets the priority and enables required interrupts in the Nested Vectored Interrupt Controller (NVIC). The enable interrupts include timers 32b0, 32b1, 16b0, 16b1, and external interrupt EINT0.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

9.2.7 SysTick.c

The SysTick timer provides the heartbeat and delay function timing.

9.2.8 32b_Timer0.c

The 32b0 timer provides the transmit timing for the master transmitter.

9.2.9 I2C.c

This contains the driver code for the I^2C peripheral on the LPC1114. The I^2C interface is used by the slave to communicate with the PCA9632 I^2C-bus to PWM LED driver. The application specific code is contained in the I2C Messages.c module described in Section 9.2.10.

9.2.10 I2C Messages.c

This module contains the functions that interpret the incoming commands and channel assignments and sends corresponding I^2C messages to the PCA9632. The PCA9632 is initialized in the 'Group Dim Mode' where a 190 Hz fixed frequency 'Group' signal is superimposed with the 6.25 kHz Individual brightness control signals PWM0, PWM1, PWM2, and PWM3. The 'Select' function on the master sets the outgoing channel number in the PLM Master transmit frame. This channel number is read by the slave to determine which register on the PCA9632 is addressed when it sends out I^2C commands. The corresponding registers on the PCA9632 and a short description are shown in Table 6.

Table 6. PCA9632 corresponding registers

Register	Range of values	Description
GRPPWM	0 to 15	this is used as a global brightness control allowing the LED outputs to be dimmed with the same value
PWM0	0 to 63	Individual Brightness control for PWM0 output
PWM1	0 to 63	Individual Brightness control for PWM1 output
PWM2	0 to 63	Individual Brightness control for PWM2 output
PWM3	0 to 63	Individual Brightness control for PWM3 output

This mode is useful to be able to set the brightness 'mix' of the different outputs, then increasing or decreasing all outputs at the same time while still maintaining the same 'mix'. Refer to the PCA9632 data sheet (Ref. 1) for more information. The brightness levels of both the GRPPWM and PWMx registers are initialized at a mid level of brightness. To get maximum duty cycle from the outputs, increase both group and individual levels via the master 'Dim Up' and 'Dim Down' and 'Select'. The Off/On command from the master results in the slave sending an I^2C message to the PCA9632 to enable or disable all or none of the LEDx output drives via the LEDOUT register on the PCA9632. The actual levels sent to the GRPPWM and PWMx registers are incremented via functions that provide a simple piece-wise linear approximation of a logarithmic dimming curve. As the eye sensitivity is logarithmic, providing some approximation of this gives the Dim Up and Dim Down commands a more natural response. The dimming curve functions are located in the PLM_Slave module discussed in Section 9.2.11.

9.2.11 PLM_Slave.c

This handles the message level communications for the PLM slave, providing synchronization, receiving commands, transmitting acknowledgements, and handling error conditions. The received 'Dim Up' and 'Dim Down' commands result in PWM values

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

that are incremented or decremented through simple dimming curve functions contained in this module as discussed in the previous section. This module also contains the External interrupt handler used to detect the HIGH-to-LOW transition of the start bit at the beginning of a bit frame.

9.2.12 PLM_UART.c

This contains the routines that handle the bit level synchronization, data transmission, and reception. This uses the timing references from the 16b1 and 32b2 timers to implement a soft UART specifically for the interfacing to the TDA5051A.

10. Demo setup and operation

10.1 Demo setup and operation — zero crossing synchronization

10.1.1 Setup requirements, master — zero crossing synchronization

Hardware needed (master):

- TDA5051A demo board
- 13 V DC power supply
  - Internally, +5 V and +3.3 V are generated from this using linear regulators.
- 4 mechanical or ‘cap sense’ switches
- power supply for ‘cap sense’ switch assembly
  - If ‘cap sense’ switches are used
  - Not required when using mechanical switches
- AC line connection
- Cabling to mate with headers on demo board for:
  - AC line
  - DC power
  - Switch interface
  - Cap sense supply (as required)

10.1.2 Setup requirements, slave — zero crossing synchronization

Hardware needed (slave):

- TDA5051A demo board
- 13 V DC power supply
- 4 LEDs with current limit resistors
- Power supply for LED assembly
- AC line connection
- Cabling to mate with headers on demo board for:
  - AC line
  - DC power
  - LED assembly with current limit resistors
  - LED power (as required)

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

10.1.3 Demo configuration — zero crossing synchronization

10.1.3.1 Master

Assemble the hardware pieces described in Section 10.1.1. If the firmware has not been loaded into the LPC1114 on the PLM demo board master, program the devices with Zero Crossing Master code, V2_09 using one of the programming methods discussed in Section 11 “Appendix A — Programming instructions”.

Master Switch setting: The master does not use the hex switch, however, it is connected to the same pins on the microcontroller as the control switches. Make sure the hex switch on the slave is set to ‘0’, the open state, to prevent any conflict with the external switches.

Hook up the power, AC line, and switches to the TDA5051A demo board as shown in Figure 13.

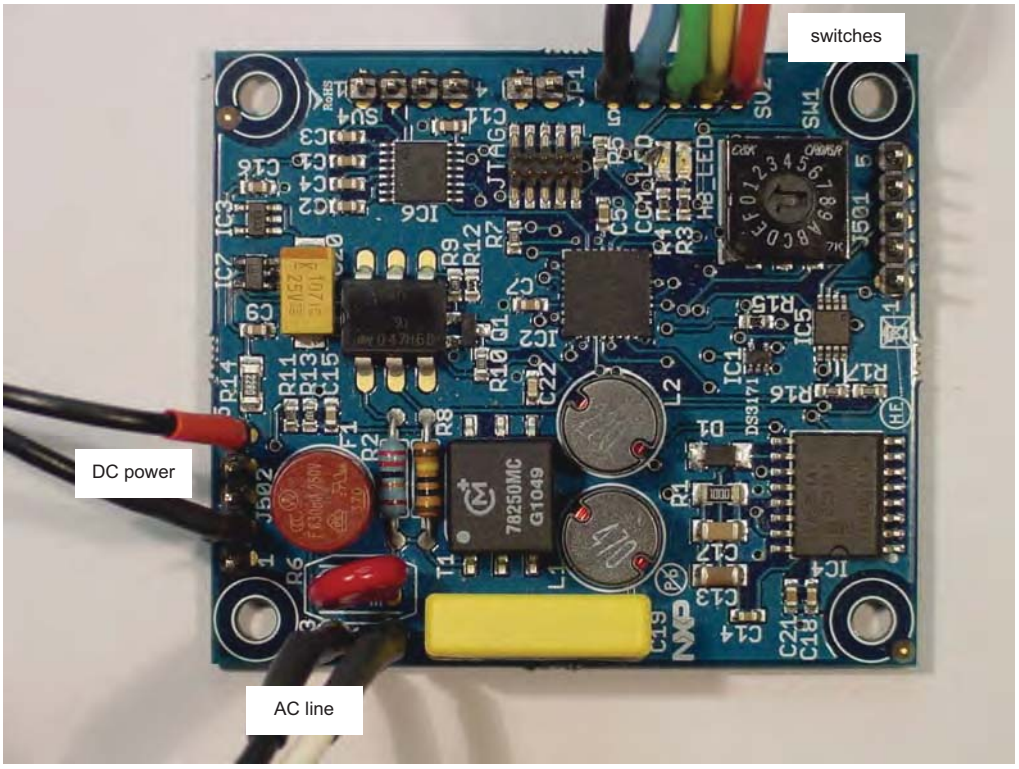


Fig 13. Hook up of power, AC line and switches to TDA5051A demo board (master)

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

10.1.3.2 Slave

Assemble the hardware pieces described in Section 10.1.2. If the firmware has not been loaded into the LPC1114 on the PLM demo board slave, program the devices with Zero Crossing Slave code, V2_09 using one of the programming methods discussed in Section 11 “Appendix A — Programming instructions”.

Slave Switch setting: The firmware is set up to transmit to Slave address ‘1’. Make sure the hex switch on the slave is set to ‘1’.

Hook up the power, AC line, and LEDs to the TDA5051A demo board as shown in Figure 14.

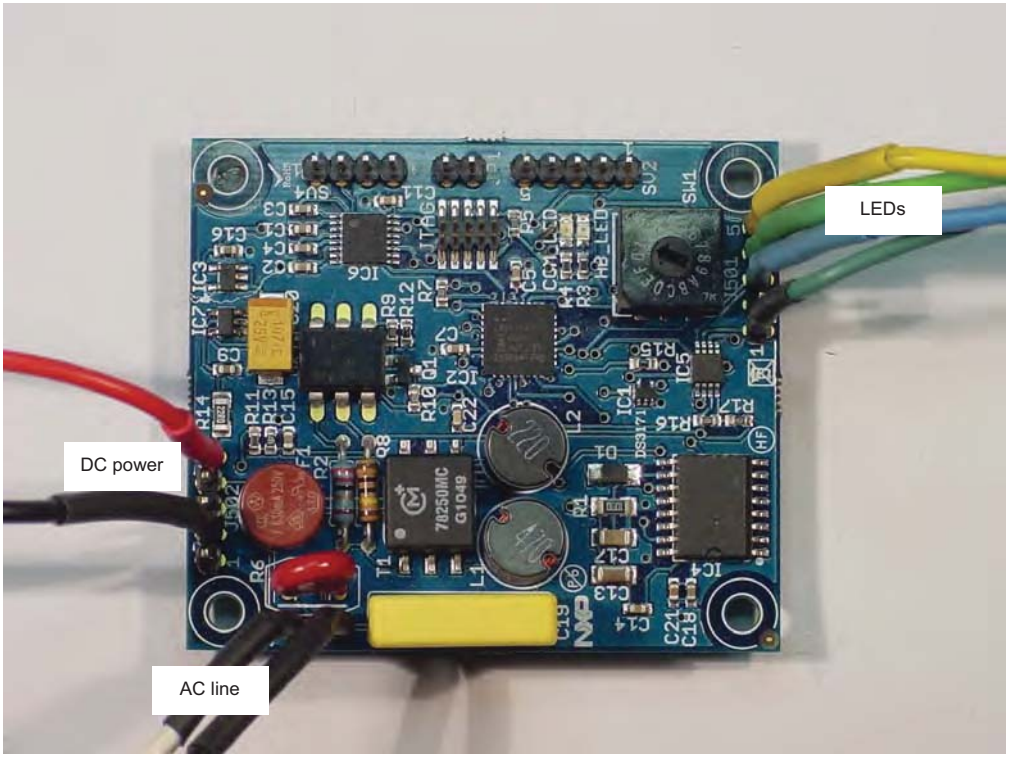


Fig 14. Hookup of DC power, AC line and LEDs to TDA5051A High-Bay demo board (slave)

10.1.4 Demo operation — zero crossing synchronization

With both master and slave powered and connected to the line, the HB LED should blink on both units. The slave should initialize with the LEDs illuminated at about ¼ of the maximum. When the ‘Dim Up’ switch is pressed, the LEDs connected to the PCA9632 outputs will get brighter. When the data is being received by the slave, the ‘COM_LED’ will blink. When the ‘Dim Down’ switch is pressed, the LEDs connected to the PCA9632 outputs will get dimmer. When this data is being received by the slave, the ‘COM_LED’ will blink. The On/Off switch sends the On/Off command to the slave.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

The ‘Select’ switch will select different dim modes. The default setting for the ‘Select’ switch is ‘group’. In this mode, a Dim Up or Dim Down command will affect all LED PWM outputs. There are 16 steps from full ON to full OFF in this mode. Pressing this switch once will select PWM output 0. The COM_LED on the master will blink once when PWM0 is selected. Only PWM0 will be affected by a Dim Up or Dim Down in this mode. There are 64 steps from ON to OFF in this mode. Pressing the switch again will select PWM1 with 64 steps. The COM_LED on the master will blink twice when PWM1 is selected. Pressing the switch again will select PWM2 with 64 steps. The COM_LED on the master will blink three times when PWM2 is selected. Pressing the switch again will select PWM3 with 64 steps. The COM_LED on the master will blink four times when PWM3 is selected. Pressing the switch again will bring you back to the group mode. The COM_LED on the master will blink 1 long blink when GROUP is selected. The relative levels of PWM0-PWM3 are preserved when dimming up or down when returning to Group mode. Additional details on the group dimming mode can be found in the PCA9632 data sheet.

10.1.5 Troubleshooting — zero crossing synchronization

In case of problems, the first thing to verify is that the HB_LEDs on both master and slave are blinking. This means the unit is powered up and the microcontroller is functioning.

Verify the AC line is connected to both master and slave. Without an AC line the TDA5051A demo board will not try to communicate since it needs to detect a zero crossing to synchronize transmission and reception.

On the Master, verify the COM_LED blinks when pressing ‘Select’. If not, verify the switch connections.

If everything looks good on the master, observe the slave to see if the COM LED blinks when a ‘Dim Up’ or ‘Dim Down’ command is received. This will verify the master is communicating with the slave.

If everything looks good up to this point, but the dim up and dim down are still not functional, verify the LED connections to the TDA5051A demo board. If the LEDs need external power, verify the LED power is operational.

If still not functional, verify the LED0-LED3 outputs with an oscilloscope to determine if a PWM signal is present.

If the PWM signal is present, the problem is probably bad LEDs or other problems with the LED assembly.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

10.2 Setup requirements — start bit (UART style) synchronization

10.2.1 Setup requirements, master — start bit (UART style) synchronization

Hardware needed (master):

- TDA5051A demo board
- 13 V DC power supply
  - Internally, +5 V and +3.3 V are generated from this using linear regulators.
- 4 mechanical or ‘cap sense’ switches
- power supply for ‘cap sense’ switch assembly
  - If ‘cap sense’ switches are used
  - Not required when using mechanical switches
- AC or DC line connection
  - DC lines may need a small network at each end to provide a controlled impedance to the PLM demo board.
- Cabling to mate with headers on demo board for:
  - AC or DC line and networks
  - DC power
  - Switch interface
  - Cap sense supply (as required)

10.2.2 Setup requirements, slave — start bit (UART style) synchronization

Hardware needed (slave):

- TDA5051A demo board
- 13 V DC power supply
- 4 LEDs with current limit resistors
- Power supply for LED assembly
- AC or DC line connection
- Cabling to mate with headers on demo board for:
  - AC or DC line and networks
  - DC power
  - LED Assembly with current limit resistors
  - LED power (as required)

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

10.2.3 Demo configuration — start bit (UART style) synchronization

10.2.3.1 Master

Assemble the hardware pieces described in Section 10.2.1. If the firmware has not been loaded into the LPC1114 on the PLM demo board master, program the devices with UART Style Master code, V1_1 using one of the programming methods discussed in Section 11 “Appendix A — Programming instructions”.

Master Switch setting: The master does not use the hex switch, however, it is connected to the same pins on the microcontroller as the control switches. Make sure the hex switch on the slave is set to ‘0’, the open state, to prevent any conflict with the external switches.

Hook up the Power, AC line, and switches to the TDA5051A demo board as shown in Figure 15.

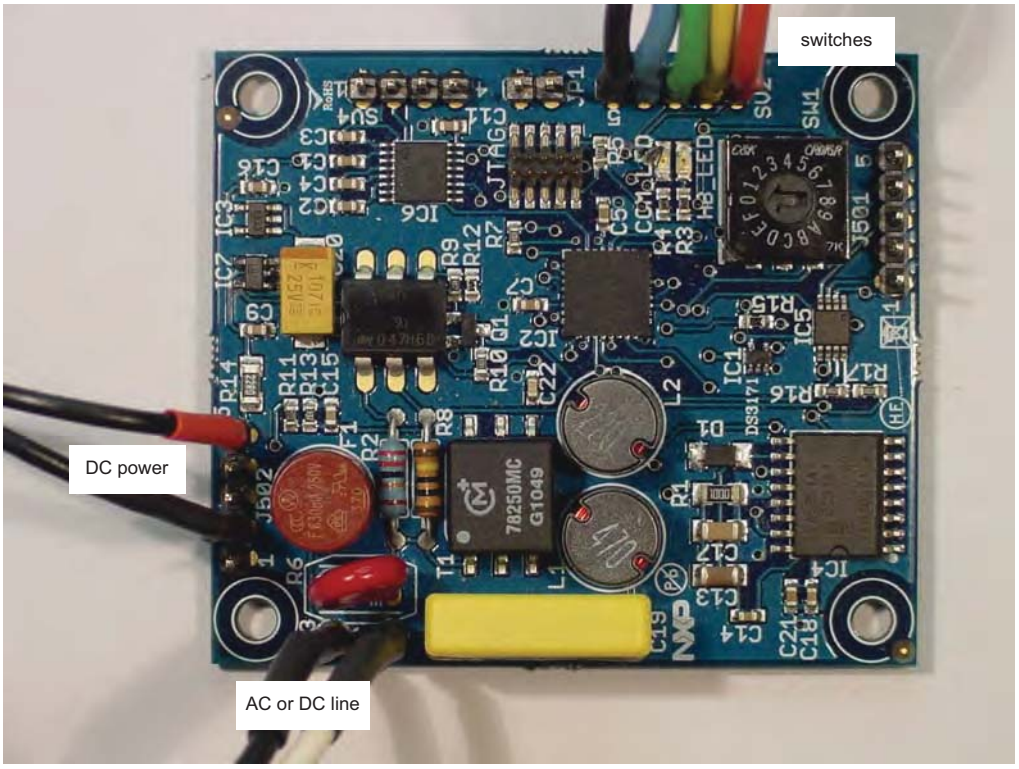


Fig 15. Hook up of power, AC line and switches to the TDA5051A demo board (master)

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

10.2.3.2 Slave

Assemble the hardware pieces described in Section 10.2.2. If the firmware has not been loaded into the LPC1114 on the PLM demo board slave, program the devices with UART Style Slave code, V1_1 using one of the programming methods discussed in Section 11 “Appendix A — Programming instructions”.

Slave Switch setting: The firmware is set up to transmit to Slave address ‘1’, so make sure the hex switch on the slave is set to ‘1’.

Hook up the power, AC line, and LEDs to the TDA5051A demo board as shown in the following:

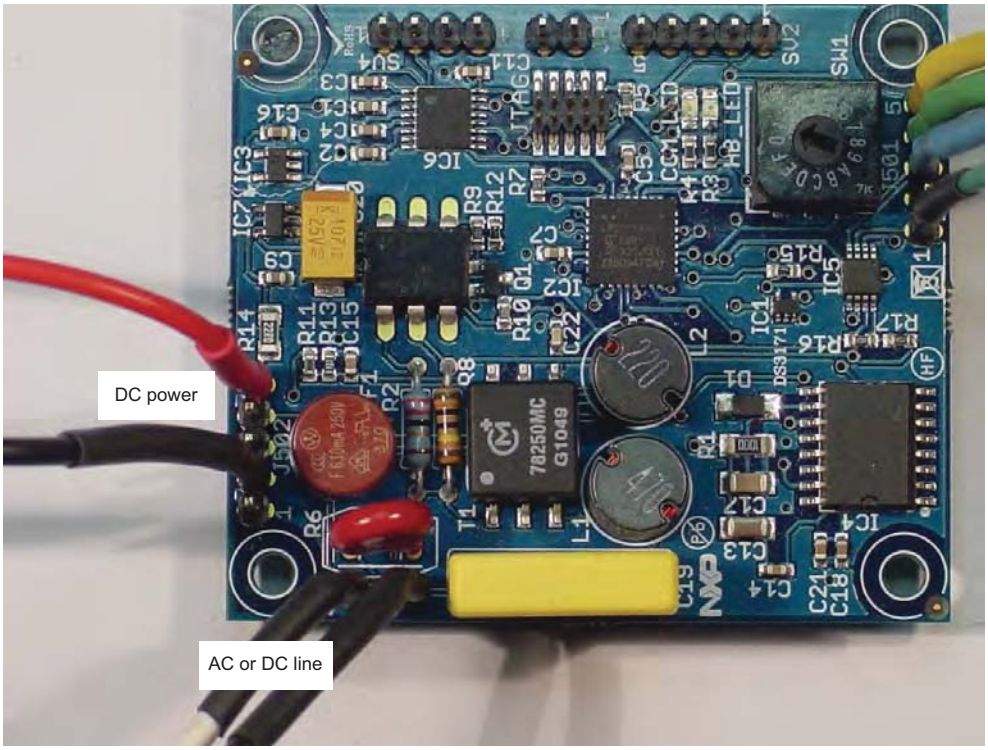


Fig 16. Hookup of power, AC line and LEDs to TDA5051A demo board (slave)

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

10.2.4 Demo operation — start bit (UART style) synchronization

With both master and slave powered and connected to the line, the HB_LED should blink on both units. The slave should initialize with the LEDs illuminated at about ¼ of the maximum. When the ‘Dim up’ switch is pressed, the LEDs connected to the PCA9632 outputs will get brighter. When the data is being received by the slave, the ‘COM_LED’ will blink. When the ‘Dim down’ switch is pressed, the LEDs connected to the PCA9632 outputs will get dimmer. When this data is being received by the slave, the ‘COM_LED’ will blink. The On/Off switch sends the On/Off command to the slave.

The ‘Select’ switch will select different dim modes. The default setting for the ‘Select’ switch is ‘group’. In this mode, a Dim Up or Dim Down command will affect all LED PWM outputs. There are 16 steps from full ON to full OFF in this mode. Pressing this switch once will select PWM output 0. The COM_LED on the master will blink once when PWM0 is selected. Only PWM0 will be affected by a Dim Up or Dim Down in this mode. There are 64 steps from ON to OFF in this mode. Pressing the switch again will select PWM1 with 64 steps. The COM_LED on the master will blink twice when PWM1 is selected. Pressing the switch again will select PWM2 with 64 steps. The COM_LED on the master will blink three times when PWM2 is selected. Pressing the switch again will select PWM3 with 64 steps. The COM_LED on the master will blink four times when PWM3 is selected. Pressing the switch again will bring you back to the Group mode. The COM_LED on the master will blink one long blink when GROUP is selected. The relative levels of PWM0-PWM3 are preserved when dimming up or down when returning to Group mode. Additional details on the group dimming mode can be found in the PCA9632 data sheet.

Remark: when using this to communicate over DC lines it is likely that a simple inductive-resistive network is needed in series at the source and the load to give a low DC resistance to minimize any I2R losses, but provide a known impedance at 125 kHz for the TDA5051A to communicate.

10.2.5 Troubleshooting — start bit (UART style) synchronization

In case of problems, the first thing to verify is that the HB_LEDs on both master and slave are blinking. This means the unit is powered up and the microcontroller is functioning.

Verify the connections to the power are complete.

On the Master, verify the COM_LED blinks when pressing ‘Select’. If not, verify the switch connections.

If everything looks good on the master, observe the slave to see if the COM_LED blinks when a ‘Dim UP’ or ‘Dim Down’ command is received. This will verify the master is communicating with the slave.

If everything looks good up to this point, but the dim up and dim down are still not functional, verify the LED connections to the TDA5051A demo board. If the LEDs need external power, verify the LED power is operational.

If still not functional, verify the LED0-LED3 outputs with an oscilloscope to determine if a PWM signal is present.

If the PWM signal is present, the problem is probably bad LEDs or other problems with the LED assembly.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

11. Appendix A — Programming instructions

The TDA5051A demo board can be programmed via an RS-232 interface onboard, or via the Serial Wire debug interface available on the LPC1114.

11.1 RS-232 interface

A MAX3232 RS-232 interface is provided on the TDA5051A to provide a programming interface to program the device via the ISP (In-System Programming) capability. The RS-232 signals are brought out on connector SV4 with the pin assignments as follows:

Pin 1 – Ground

Pin 2 – Receive

Pin 3 – Transmit

Pin 4 – LPC1114 Reset

To put the device into ISP mode, insert a jumper into JP1 (ISP_MODE) and reset the device by grounding the reset pin and releasing, or cycling the power to provide a power-on reset. The ISP configuration is shown in Figure 17.

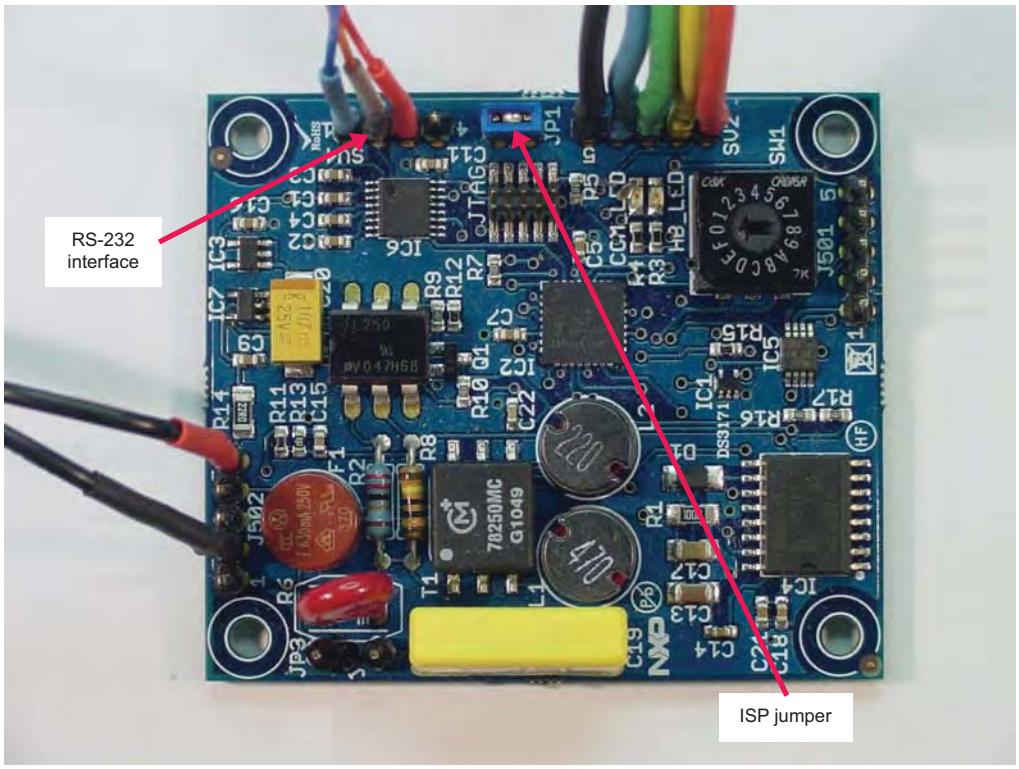
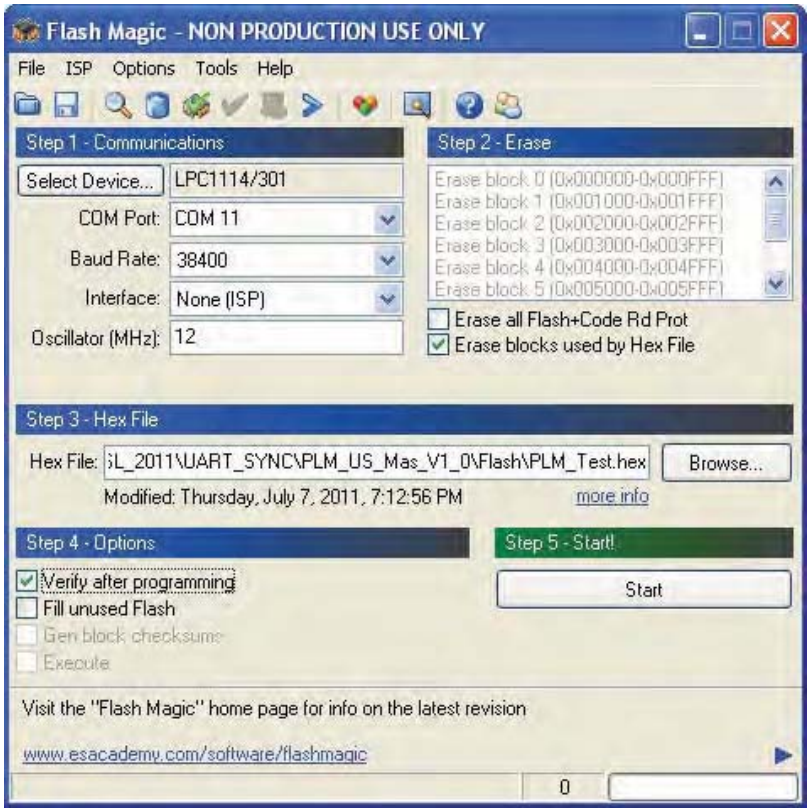


Fig 17. ISP configuration

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

A PC utility is then needed to send the program code to the microcontroller. The utility used is Flash Magic (www.nxp.com/redirect/flashmagictool.com). This is a free download. Install the utility and open it up. Configure the utility as follows:



002aag485

Fig 18. Configure Flash Magic utility

For Step 1, The COM port is the RS-232 port assignment for your PC, usually COM1. Fill in the other entries as shown in Figure 18.

For Step 2, check the box as shown in Figure 18.

For Step 3, browse to the hex file provided for programming. In addition to the above set up values, pull down the options menu and select ‘Advanced Options’. The communications and hardware configurations should be set as shown in Figure 19 and Figure 20.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

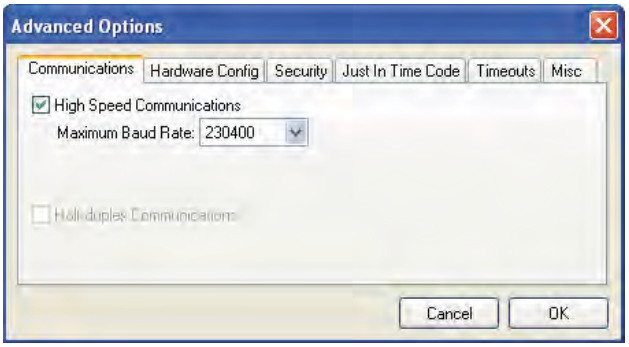


Fig 19. Communications configuration

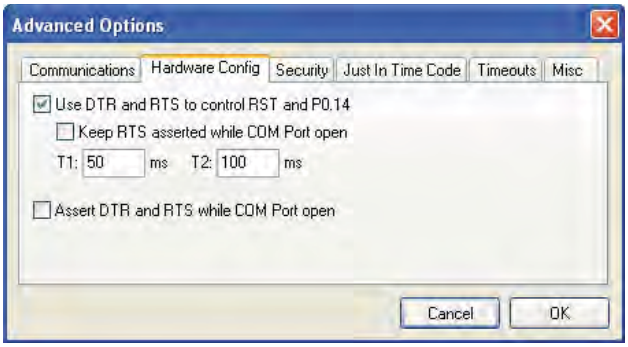
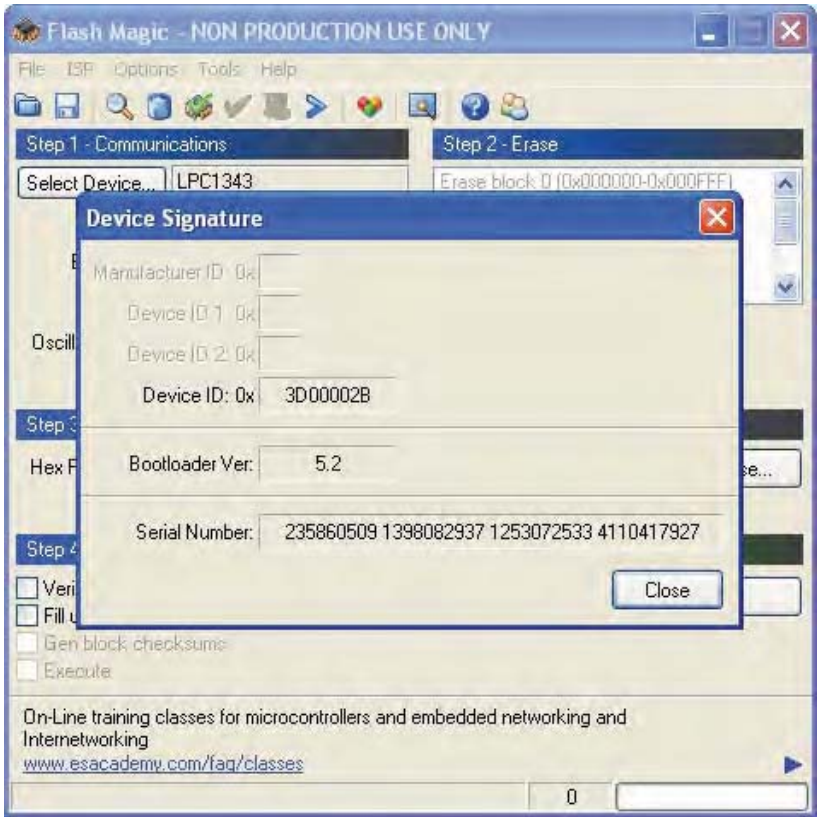


Fig 20. Hardware configuration

To verify Flash Magic is connected, pull down the 'ISP' tab and select 'Read Device Signature'. If Flash Magic returns a signature similar to that shown in Figure 21, then communications is established.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.



002aag488

Fig 21. 'Read Device Signature' result

To complete programming, verify the correct hex file is loaded and press 'Start' (Step 5 in the Flash Magic Window shown in Figure 18).

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

11.2 SW programming interface

The board may also be programmed via SW (Serial Wire). Many of the IDE vendors support SWD programming. The SWD cable should be removed after programming to ensure proper operation. The debug connection for SWD programming is shown in Figure 22.

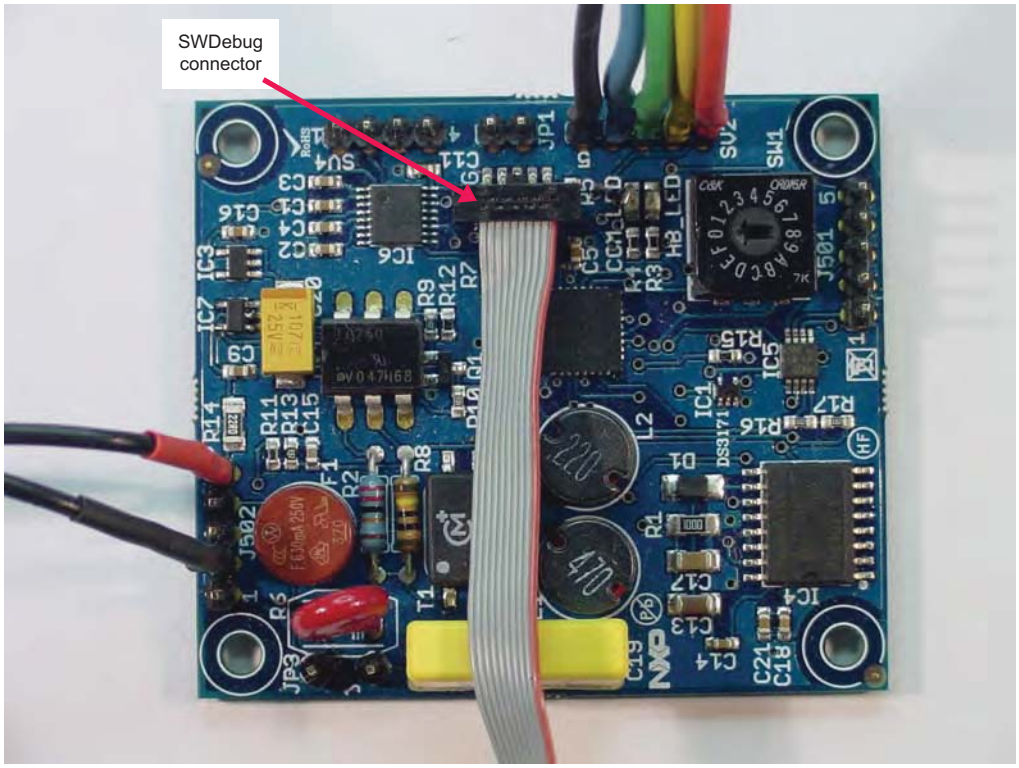
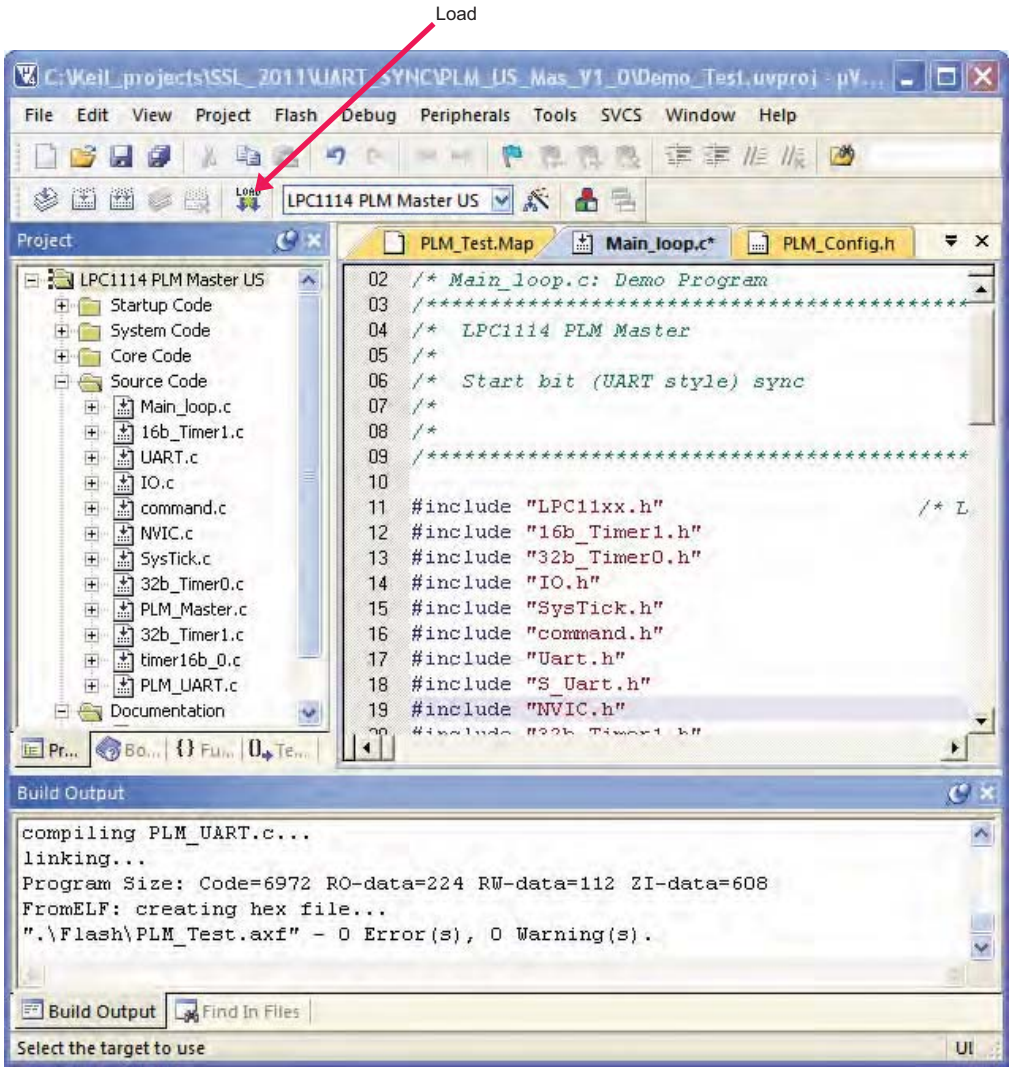


Fig 22. Debug connection for SWD programming

To program via SWD, open up Keil uVision4, MDK-ARM version 4.21 or higher. Make sure the correct project is loaded and compiled. The press the ‘Load’ button as shown in Figure 23.

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.



002aag490

Fig 23. 'Load' button

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

12. Appendix B — TDA5051A lighting demo board schematic

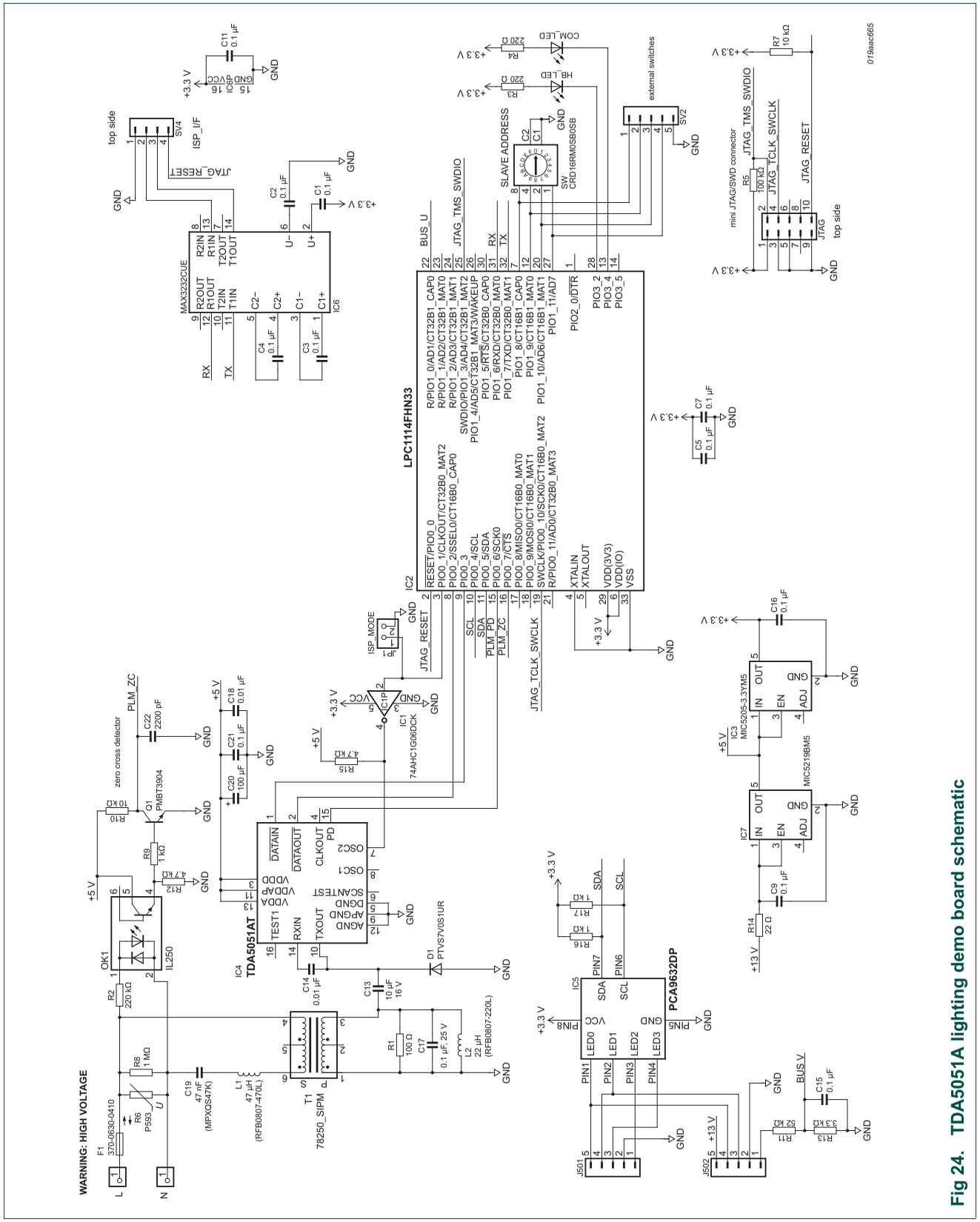


Fig 24. TDA5051A lighting demo board schematic

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

13. Abbreviations

Table 7. Abbreviations

Acronym	Description
ADC	Analog-to-Digital Converter
AGC	Automatic Gain Control
ASK	Amplitude Shift Keying
CL	Current Limiting (resistor)
DAC	Digital-to-Analog Converter
I^2C-bus	Inter Integrated Circuit bus
I/O	Input/Output
IDE	Integrated Development Environment
LED	Light Emitting Diode
NVIC	Nested Vectored Interrupt Controller
PLM	Power Line Modem
PWM	Pulse Width Modulation
ROM	Read Only Memory
SSL	Solid State Lighting
SWD	Serial Wire Debug
UART	Universal Asynchronous Receiver/Transmitter

14. References

[1] **PCA9632, 4-bit Fm+ I^2C-bus low power LED driver —** NXP Semiconductors; Product data sheet; www.nxp.com/documents/data_sheet/PCA9632.pdf

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

15. Legal information

15.1 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.

15.2 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.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

15.3 Trademarks

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

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

16. Contents

continued >>

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2012. All rights reserved.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section ‘Legal information’.

© NXP B.V. 2012. All rights reserved.

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

Date of release: 11 May 2012
Document identifier: UM10495