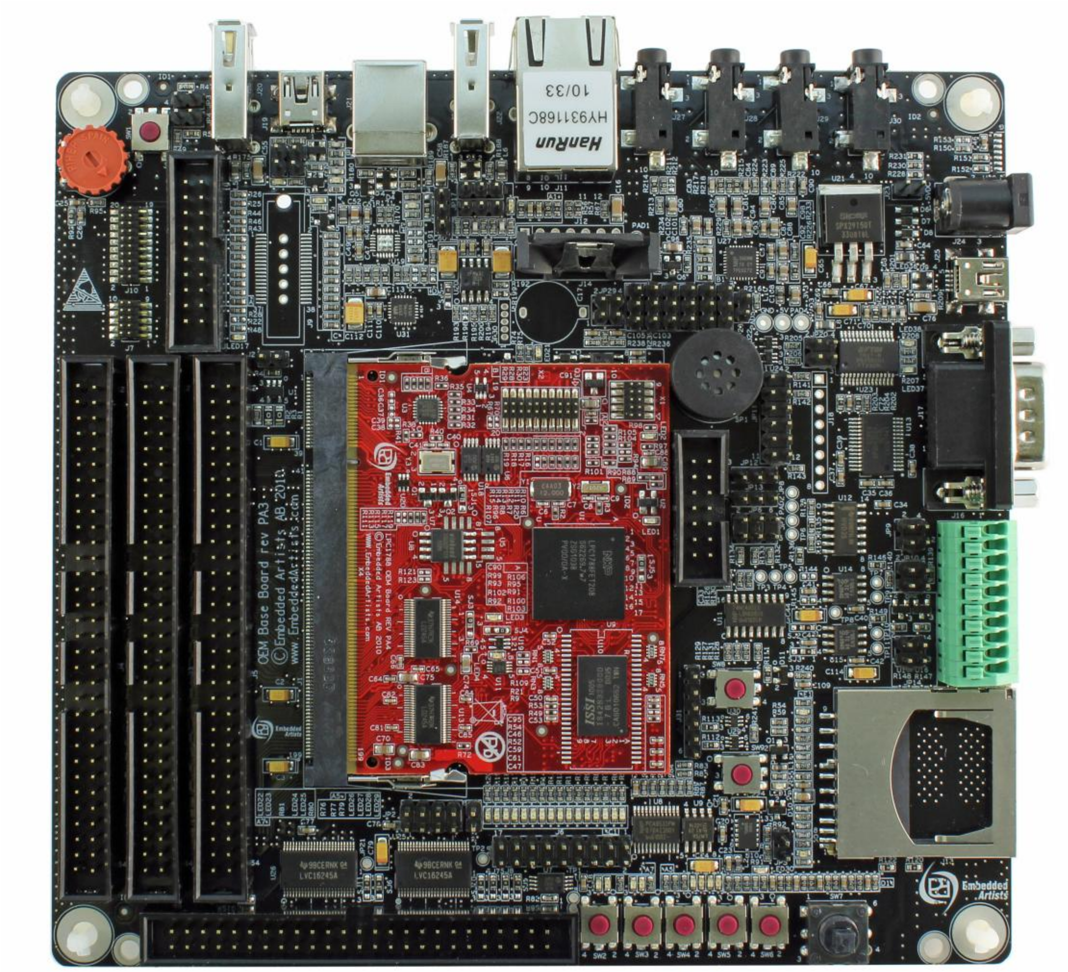
LPC3250 Developer's Kit v2
User's Guide



Get Up-and-Running Quickly and
Start Developing Your Application On Day 1!

Embedded Artists AB
Davidshallsgatan 16
211 45 Malmö
Sweden

info@EmbeddedArtists.com
http://www.EmbeddedArtists.com

Disclaimer

Embedded Artists AB makes no representation or warranties with respect to the contents hereof and specifically disclaim any implied warranties or merchantability or fitness for any particular purpose. Information in this publication is subject to change without notice and does not represent a commitment on the part of Embedded Artists AB.

Feedback

We appreciate any feedback you may have for improvements on this document. Please send your comments to support@EmbeddedArtists.com.

Trademarks

All brand and product names mentioned herein are trademarks, services marks, registered trademarks, or registered service marks of their respective owners and should be treated as such.

Table of Contents

1 Document Revision History

Revision	*Date*	*Description*
A	2012-06-18	First official revision of document for the OEM Base Board
PB1	2012-10-16	Clarified I2S jumper positions in Figure 25.
PB2	2013-06-05	Correction about which connector to use for USB Device on USB2.
B	2017-04-28	Added information that accelerometer MMA7455LT is no longer mounted.

2 Introduction

Thank you for buying Embedded Artists' *LPC3250 Developer's Kit* based on NXP's ARM926EJ-S LPC3250 microcontroller.

This document is a User's Guide that describes the *LPC3250 OEM Board* and the *OEM Base Board* hardware design, which together form the *LPC3250 Developer's Kit.*

Note that there is a previous version of the *LPC3250 Developer's Kit* that was based on the *QVGA Base Board*. This manual is for the new *OEM Base Board* (not the same as the *QVGA Base Board*).

2.1 Features

Embedded Artists' *LPC3250 OEM Board* lets you get up-and-running quickly. The small form factor OEM board offers many unique features that ease your learning curve and program development. The board has been designed for OEM applications with volume discount available. The features of the LPC3250 OEM Board are:

- NXP's ARM926EJ-S LPC3250 microcontroller in BGA package, with 256 KByte internal RAM
- External data memory: 64 MB DDR SDRAM (16-bit databus width)
- External FLASH memories: 128 MB (1Gbit) NAND FLASH and 4 MB (32Mbit) SPI-NOR FLASH
- 13.0000 MHz crystal for cpu
- 32.768kHz RTC crystal
- 256 Kbit I2C E2PROM for storing non-volatile parameters
- Buffered 16-bit data bus for external expansion
  - Voltage translation buffers. 3.3V external bus can be used.
- 100/10M Ethernet PHY/interface based on National/TI DP83848
- 200 pos expansion connector (SODIMM-200 format, 0.6mm pitch)
  - Almost all LPC3250 pins available (except dedicated pins for on-board memories and internal powering)
- 3.15-3.3V powering
- Onboard reset generation
- 5 LEDs
- Compact SODIMM size: 66 x 48 mm
  - Eight layer PCB design for compact design and best EMC performance

There is an accompanying *OEM Base Board* that can be used for initial prototyping work. The features of the board are:

- Interfaces and Connectors
  - 200 pos, 0.6mm pitch SODIMM connector for *OEM Board*
  - LCD expansion connector with control signals for touch screen interface
  - Expansion connector with all *OEM Board* signals
  - Ethernet connector (RJ45)
  - MMC/SD interface & connector

  - USB2: Device or Host interface & connector
  - Provision for NXP JN5148 RF module (former Jennic) interface (RF module not included)
  - Full modem RS232
  - RS422/485 interface & connector
  - Provision for IrDA transceiver interface (transceiver not mounted)
  - I2S audio codec (mic in, line in, line out, headphone out)
  - JTAG connector
  - Trace/ETM connector (not mounted)
- Power
  - Power supply, either via USB or external +5V DC
  - Coin cell powering supported (CR1025 battery not included) for RTC and LED on ALARM output.
- Other
  - *OEM Board* current measuring
  - Parallel NOR flash on external memory bus
  - 16-bit register and LEDs on external memory bus
  - 5-key joystick
  - 3-axis accelerometer (I2C connected) - not mounted on boards after March 2017
  - LM75 temperature sensor (I2C connected)
  - 5 push-button keys (four via I2C and one on P2.10)
  - 9 LEDs (8 via I2C and one on P2.10)
  - Analog input
  - USB-to-serial bridge on UART #0 (FT232R) and ISP functionality
  - Reset push-button and LED
  - Speaker output on analog output from *OEM Board*, or from I2S audio codec
  - Compact size: 160x150 mm

2.2 ESD Precaution

Please note that the *LPC3250 OEM Board* and *OEM Base Board* come without any case/box and all components are exposed for finger touches – and therefore extra attention must be paid to ESD (electrostatic discharge) precaution.

Make it a habit always to first touch the metal surface of one of the USB or Ethernet connectors for a few seconds with both hands before touching any other parts of the boards. That way, you will have the same potential as the board and therefore minimize the risk for ESD.

Never touch directly on the *LPC3250 OEM Board* and in general as little as possible on the *OEM Base Board*. The push-buttons on the *OEM Base Board* have grounded shields to minimize the effect of ESD.

Note that Embedded Artists does not replace boards that have been damaged by ESD.

2.3 General Handling Care

Handle the *LPC3250 OEM Board* and *OEM Base Board* with care. The boards are not mounted in a protective case/box and are not designed for rough physical handling. Connectors can wear out after excessive use. The *OEM Base Board* is designed for prototyping use, and not for integration into an end-product.

For boards with LCD, do not exercise excessive pressure on the LCD glass area. That will damage the display. Also, do not apply pressure on the flex cables connecting the LCD/touch screen. These are relatively sensitive and can be damaged if too much pressure is applied to them.

Note that Embedded Artists does not replace boards where the LCD has been improperly handled.

2.4 LPC3250 Core Voltage Precaution

The core voltage for the LPC3250 can be dynamically changed, via I2C commands sent to the LTC3447 voltage converter. The voltage can be set to up to 2V. Note that this is above the limits of the core voltage. Read the LPC3250 datasheet for details (the absolute maximum core voltage allowed is 1.39V).

Note that Embedded Artists do not replace LPC3250 OEM boards where the core voltage ($V_{DD(CORE)}$) has been raised above 1.39 Volt. It is the user's responsibility not to exceed the voltage specification found in the datasheet.

2.5 CE Assessment

The *LPC3250 Developers Kit* (consisting of the *LPC3250 OEM Board* and *OEM Base Board*) is CE marked. See separate *CE Declaration of Conformity* document.

The *LPC3250 Developers Kit* is a class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

EMC emission test has been performed on the *LPC3250 Developers Kit*. Standard interfaces like Ethernet, USB, serial have been in use. General expansion connectors where internal signals are made available (for example processor pins) have been left unconnected. Connecting other devices to the product via the general expansion connectors may alter EMC emission. It is the user's responsibility to make sure EMC emission limits are not exceeded when connecting other devices to the general expansion connectors of the *LPC3250 Developers Kit*.

Due to the nature of the *LPC3250 Developers Kit* – an evaluation board not for integration into an end-product – fast transient immunity tests and conducted radio-frequency immunity tests have not been executed. Externally connected cables are assumed to be less than 3 meters. The general expansion connectors where internal signals are made available do not have any other ESD protection than from the chip themselves. Observe ESD precaution.

Note that the *LPC3250 OEM* board is classified as a component and is hence not CE marked separately. It can perform different functions in different integrations and it does not have a direct function. It is therefore not in the scope of the CE Directive. An end product, where an *OEM Board* is integrated into, is however very likely to need CE marking.

2.6 Other Products from Embedded Artists

Embedded Artists have a broad range of LPC and i.MX based boards that are very low cost and developed for prototyping / development as well as for OEM applications. Modifications for OEM applications can be done easily, even for modest production volumes. Contact Embedded Artists for further information about design and production services.

2.6.1 Design and Production Services

Embedded Artists provide design services for custom designs, either completely new or modification to existing boards. Specific peripherals and I/O can be added easily to different designs, for example, communication interfaces, specific analog or digital I/O, and power supplies. Embedded Artists has a broad, and long, experience in designing industrial electronics in general and with NXP's LPC microcontroller and i.MX application processor families in specific.

2.6.2 OEM / Education / QuickStart Boards and Kits

Visit Embedded Artists' home page, www.EmbeddedArtists.com, for information about other *OEM / Education* / *QuickStart* boards / kits or contact your local distributor.

3 LPC3250 OEM Board Design

Please read the *LPC3250 OEM Board* datasheet and associated schematic for information about the board. Some additional information about the *LPC3250 OEM Board* is presented below.

3.1 LPC3250 OEM Board Schematics

3.1.1 Schematic Page 2: Crystals

The microprocessor crystal frequency is 13.0000 MHz, which is the recommended frequency from NXP. An internal PLL can create many other frequencies from this, like 208 MHz and 266 MHz.

The LPC3250 has an internal real-time clock (RTC) block that can be used to provide real-time and alarm function. A 32.768 kHz crystal gives the base frequency for the RTC. The RTC block can be powered via a separate supply (for example from a battery or high-capacity capacitor). The 32.768 kHz can also be used as main oscillator via a PLL: 32.768 kHz x 397 = 13.009 MHz.

Note that the clocking structure is different from the LPC1xxx/2xxx family. It is a more complex structure but also much more versatile and flexible. There is no shortcut but to read the LPC3250 User's Manual in detail to understand the options and settings.

3.1.2 Schematic Page 2: Booting

The LPC3250 starts executing from an on-chip ROM, containing the bootloader. Note that the LPC3250 does not contain any on-chip FLASH memory. Program code must be loaded from an external source into the on-chip SRAM.

The default boot is from an external memory (see LPC3250 User's Manual for details). Program code is typically stored in NAND or SPI-NOR flash.

By pulling pin GPI_01/SERVICE_N low, UART boot mode is activated. This is a method for downloading code from the PC, for example for programming the bootloader for the first time. Note that pin GPI_01/SERVICE_N can be pulled low by pressing SW6-key on the OEM Base Board.

3.1.3 Schematic Page 2: SPI NOR FLASH

There is a 32Mbit (4 MByte) NOR flash connected to the SPI bus. Embedded Artists can choose to mount, either S25FL032P from Spansion, AT45DB321 from Atmel, or other, on the board depending on component availability at the time of production. Mounted chips will be supported by the LPC3250 bootloader. However, commands used to program the memory differ. Chip id should always be read out to determine exact type mounted.

3.1.4 Schematic Page 2: Reset Generation

The reset generation is handled by a standard voltage supervisor chip, CAT811R from Catalyst Semiconductor. The reset signal will be held active (i.e., low) until the supply voltages, +3.3V, is within margins (above 2.63V). The reset duration is typically 200 mS (consult the CAT811R datasheet for exact details). The output reset signal is push/pull output that is converted to an open-collector / open-drain output via the 74LVC1G125 buffer. An external reset source can pull the reset signal low (with an open-collector/open-drain output). The RESET_N input on the LPC3250 has a 1.2V voltage range (voltage domain: VDD_RTC). A 74LVC1G125 buffer makes sure this voltage range is not exceeded by the external signal RESET_IN (which has 3.3V range).

3.1.5 Schematic Page 2: I2C E2PROM

There is a 256 kbit E2PROM accessible via the I2C interface (I2C1). The LPC3250 has two on-chip I2C communication channels (I2C1 and I2C2). More peripheral units are easily connected to the two-wire I2C bus, just as long as the addresses do not collide. The address of the 256kbit E2PROM is 0xA0, which is also indicated in the schematic.

There are 2.2 kohm pull-up resistors (pull-ups are always needed on I2C busses) on the board on both I2C channels.

3.1.6 Schematic Page 3: 1.2V and 1.8V Fixed Voltages

The LPC3250 requires three fixed voltages; 1.2V for the core, 1.8V for the memory interface and 3.15-3.3V for the rest of the i/o interfaces. The 1.2V and 1.8V voltages are generated by two LM3671MF step-down switching regulators from National Semiconductor. These regulators are capable of generating 600mA, which by far exceed the needed current by the LPC3250 and other components on the LPC3250 OEM Board. The 3.15-3.3V voltage is the input voltage to the LPC3250 OEM Board, see below.

The Real-time clock also needs a 1.2V power, which is generated by a LDO (MIC5232).

3.1.7 Schematic Page 3: 1.2V Adjustable Voltage

The core 1.2V voltage is adjustable and is generated by the step down switching regulator LTC3447 from Linear Technologies. This regulator is capable of generating 600mA, which also by far exceed the needed current by the LPC3250. The adjustment is done via an I2C channel, I2C1 in the LPC3250 case. The core voltage can be adjusted as a power save feature. By lowering the voltage (down to 0.9V), the total power consumption can be lowered but the clock frequency of the core must then also be lowered. This is a trade-off that is important for hand held/portable equipment. ***Note that it is the user's responsibility not to program the LTC3447 to generate too high core voltage, which is possible.*** The LTC3447 can generate voltages up to 2V, which by far exceed the limits for the LPC3250.

See the LPC3250 datasheet for exact details about voltage ranges, but it is in the region of 1.1-1.39V. Also see the LTC3447 datasheet for details about how to adjust the voltage (it is a write-only register). The I2C address for the LTC3447 is indicated in the schematic.

Note that a core voltage of 1.35V should be programmed when working with the external DDR SDRAM.

3.1.8 Schematic Page 3: Input Voltage

The input voltage to the LPC3250 OEM Board is given by the requirements of the LPC3250. The recommended input voltage range is 3.15V to 3.3V. The input supply must be stable but there are no special needs for bulk capacitors close to the power pins on the expansion connectors. The needed capacitors are placed close to the switched step down switching regulators on the LPC3250 OEM Board.

Note that the LPC3250 OEM Board is sensitive to input noise on the input voltage. The peak-to-peak noise should be below 10mV. A linear regulator to feed the input voltage is strongly recommended.

3.1.9 Schematic Page 4: External Memories

Page 4 of the schematic contains the external memory interface and the external memories. The memory interface uses a 16-bit databus and operates at 1.8V level, which minimizes power consumption.

3.1.10 Schematic Page 4: DDR SDRAM

A 512 MBit (64 MByte) Mobile DDR SDRAM is used (MT46H32M16LF from Micron). The chip is powered by 1.8V and is organized as 32Mbit x16, i.e. it has 16-bit databus width. The chip is connected to EMC_DYCS0_N (memory bank #0 for dynamic RAM) at address range 0x8000 0000 – 0x9FFF FFFF.

Note that memory bank #1 for dynamic RAM is not available (i.e., signal EMC_DYCS1_N is not used).

3.1.11 Schematic Page 4: NAND Flash

A 1 Gbit (128 MByte) NAND flash is used (K9F1G08 from Samsung). The chip is powered by 3.3V and has 8-bit databus width. The NAND flash builds on a single-level cell (SLC) technology and has a page size of 2112 bytes (2,048 + 64 bytes). Note that the chip is not directly accessible via the memory bus. Instead, all accesses must be done via the on-chip NAND flash controller of the LPC3250.

The NAND FLASH has an optional busy output that can be used for controlling the erase/program operations with better precision. This signal is connected to GPO19 and is also available on the (SODIMM) expansion connector. For compatibility with other OEM boards, the busy signal can also be routed to a suitable (i.e., free) input pin. The *OEM Base Board* can connect the signal to GPIO72 by inserting a jumper between pin 3-4 on JP2. This feature is not needed for the LPC3250 OEM board, where the busy signal is connected tp GPO19.

Also note that the busy status of the chip is available under software control so the hardware signals is not strictly needed.

3.1.12 Schematic Page 4: Buffers to External Interface

The LPC3250 memory interface is available on the expansion connector. The data bus width is 16-bits on the external interface. The relevant signals are buffered. The following four static memory regions are available for external access:

- External static bank #0 (0xE000 0000 – 0xE0FF FFFF)
  16-bit databus width and 16MByte in size.
- External static bank #1 (0xE100 0000 – 0xE1FF FFFF)
  16-bit databus width and 16MByte in size.
- External static bank #2 (0xE200 0000 – 0xE2FF FFFF)
  16-bit databus width and 16MByte in size.
- External static bank #3 (0xE300 0000 – 0xE3FF FFFF)
  16-bit databus width and 16MByte in size.

By default (R44 = 0 ohm, R43 not mounted), signal N_ABUF_EN is pulled low and the two buffers for address and control signals (U13 and U14) are enabled and act as output (from the LPC3250 OEM Board).

The buffered version of the LPC3250 signal OE controls the direction of the data bus buffer (U15). During read operations the buffer acts as an input and during write operations it acts as an output. The data bus buffer is controlled by the signals BLS0 and BLS1, each controlling lower and upper bytes of the 16-bit databus. These signals are active when accessing the external static memory regions.

The buffers are dual voltage buffers and act as level translators between the internal 1.8V signal levels and the external levels. Connect the external bus voltage to VDD_EXT. See the datasheet of 74AVCA164245 for exact details about voltage range. Normally 3.3V powering is used on the external side.

3.2 Memory Layout

The external memory controller on the LPC3250 defines eight memory regions. See table below for details about usage.

Name	***Control signal***	***Address range***	***Memories on LPC3250 OEM Board***	***External memory bus comment***
Static memory #0	CS0	0xE000 0000 – 0xE0FF FFFF		Available for external use.
Static memory #1	CS1	0xE100 0000 – 0xE1FF FFFF		Available for external use.

Static memory #2	CS2	0xE200 0000 – 0xE2FF FFFF		Available for external use. *OEM Base Board* can connect a parallel NOR flash to this chip select.
Static memory #3	CS3	0xE300 0000 – 0xE3FF FFFF		Available for external use. *OEM Base Board* can connect a 16-bit parallel register to this chip select.
Dynamic memory #0	DYCS0	0x8000 0000 – 0x9FFF FFFF	SDRAM (512 MBit = 64 MByte in size)	Cannot be accessed on external memory bus.
Dynamic memory #1	DYCS1			Cannot be accessed.

As seen in the table above, it is only the static memory regions that are available on the external memory bus from the *LPC3250 OEM Board*. The data bus buffers on the *LPC3250 OEM Board* are controlled automatically and only enabled when a static memory region is accessed. The address and control bus buffers are always enabled.

Note that the BLS0 and BLS1 pins must be initialized for these functionalities. Else the buffer control will not work correctly.

3.3 LEDs

P2.10, P2.11 and P2.12 controls three LEDs on the *LPC3250 OEM Board*. The LED driving is isolated via mosfet transistors so P2.10, P2.11 and P2.12 are not loaded because of this.

4 OEM Base Board Design

This chapter contains information about the peripherals and general design of the *OEM Base Board* and how to set the different jumpers on the board. The schematic can be downloaded in pdf format from the support page, and is recommended to have printed out while reading this chapter.

Section naming begins with *SPx*, which is short for *Schematic Page x*.

The picture below gives an overview of the *OEM Base Board* design.

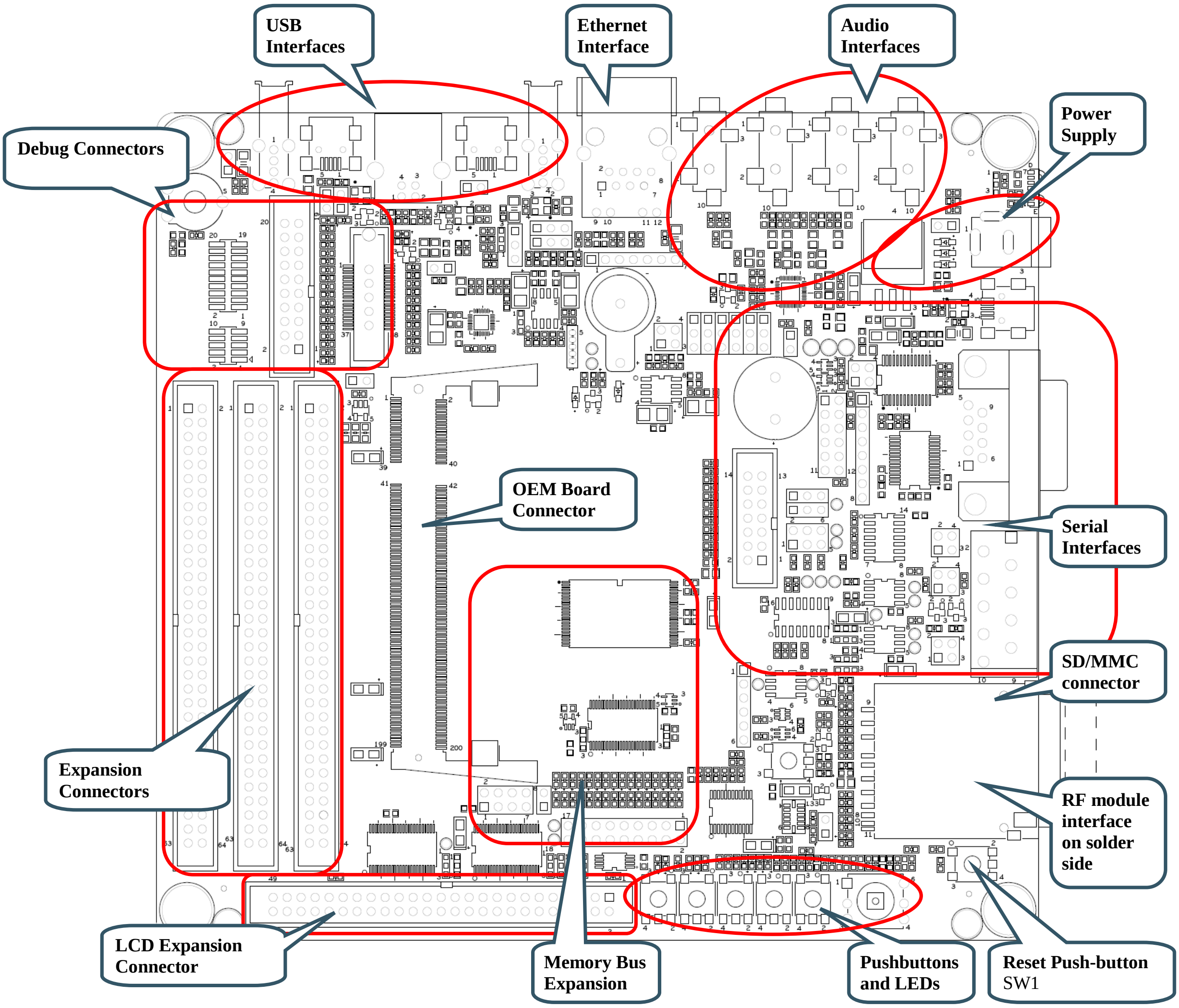


Figure 1 – OEM Base Board Overview

The OEM base board has been designed to support many processors in NXP's LPC family, i.e. different OEM boards. The signal names are general and represent the OEM base board functionality, rather than the LPC3250 signal names. Some functionalities of the OEM Base Board is not supported by the LPC3250, for example the CAN interface and the USB1 interface.

4.1 Modifications to OEM Base Board

The *OEM Base Board* has been designed to be flexible. Most options can be controlled via jumpers but some options might need soldering. Note that modifications to the board are done at own risk and void all warranties.

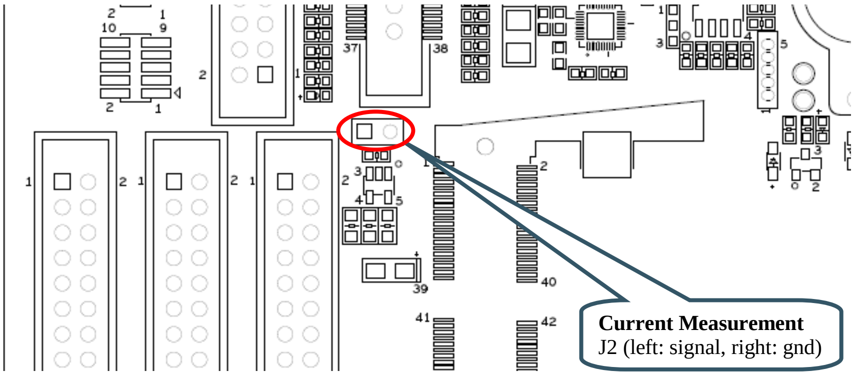
4.2 SP2: OEM Board Connector

The LPC3250 OEM board connector is a standard DDR2 SO-DIMM socket with 200 positions and 0.6mm pitch. The JEDEC standard for this form factor is called MO-224. It has 1.8V keying (which is what DDR2 stands for).

4.3 SP2: Current Measurements

It is possible to accurately measure current consumption of the LPC3250 OEM board. This can be very valuable when working with applications that make use of the low power modes of the LPC3250 processors. The circuit is based on the chip ZXCT1010 from Diodes/Zetex. This chip generates a voltage output proportional to the current through R2. This voltage can be measured over J2. 100mA gives a 500mV output voltage, or expressed differently, 1mV correspond to 0.2mA

It is possible to remove R1, R2 and/or R3 for measuring current with an external multimeter. Note that VCC_MAIN and VCC_BUFFERS are separate power supplies on the LPC3250 OEM board. The combined current is measured over R2 but each of the supplies can also be measured over R1 and R3, respectively.



Current Measurement
J2 (left: signal, right: gnd)

Figure 2 – Current Measurement J2

4.4 SP3: Expansion Connectors

All relevant OEM board signals are available for external use via three 64 pos IDC expansion connectors; J3, J4 and J5. The expansion connectors are close to the SO-DIMM connectors to minimize signal distortion.

Note that some OEM board circuits may need to be disconnected before externally used. Carefully investigate the need for this before using a signal for external expansion.

Note that J4 has gathered all needed signals for expanding the memory bus (16-bit bus expansion). J4 is the expansion connector closest to the board edge.

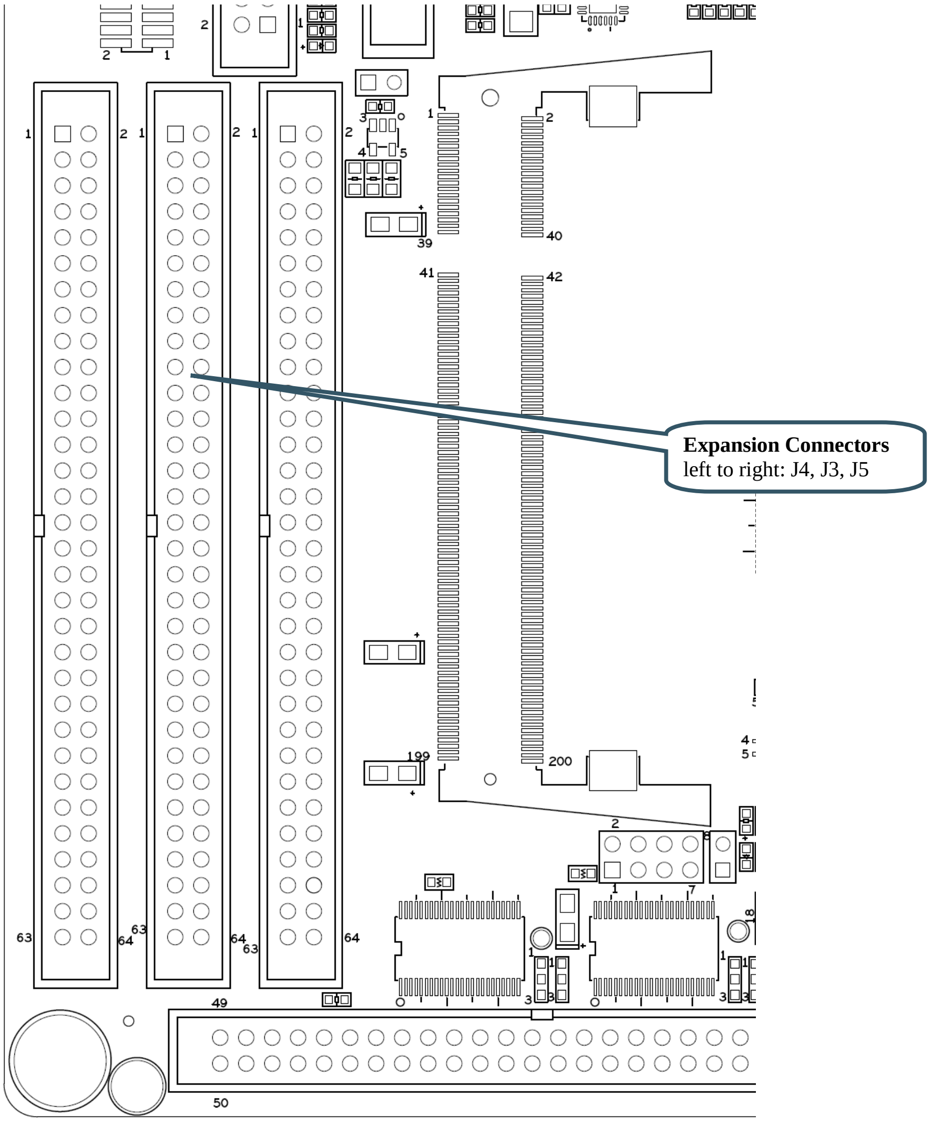


Figure 3 – Expansion Connectors J3, J4, J5

4.5 SP4: External Memory Bus

This part of the OEM Base Board demonstrates how the external memory bus can be used for expansion of custom circuits. Two different devices are connected to the memory bus; a 16-bit parallel NOR flash and a 16-bit register.

In order to be universal between 16- and 32-bit databus width OEM Boards, only the lower 16 bits are connected. This does not affect the LPC3250 OEM board since it only exists in 16-bit databus width.

The 16-bit parallel NOR flash is enabled when JP1 is inserted, see picture below for guidance where to find JP1 on the OEM Base Board. Note that the signal name is cryptic since the OEM Base Board is compatible with many different OEM Boards. For the LPC3250 OEM Board, the signal BCSY is actually signal BCS2. This means that the NOR flash is accessible in memory region: 0xE200 0000 - 0xE2FF FFFF.

Also, a 16-bit register is connected to the external memory bus. The 16 bits in the register directly drives 16 LEDs (a high signal light a LED). The signals are also available on an expansion connector (J6). It can for example connect to a logic analyzer, for high-bandwidth logging. The upper and lower 8 bits are individually writeable. Signals BBLS0/BBLS1 controls the lower and upper 8 bits, respectively. Since the OEM Base Board is universal and supports many different OEM Boards, the chip select signal is either signal BCSX or GPIO69. For the LPC3250 OEM Board, the signal BCSX is actually signal BCS3. This means that the 16-bit register is accessible in memory region: 0xE300 0000 - 0xE3FF FFFF. No jumper in JP2 is needed when working with the LPC3250 OEM Board.

SJ12 shall be in default position (pad 1-2 shorted) to let BCS2 control chip select of the 16-bit register.

SJ1 controls the output enable of the register. By default it is grounded (pad 1-2 is shorted) and hence the register drives the LEDs and expansion connector, J6.

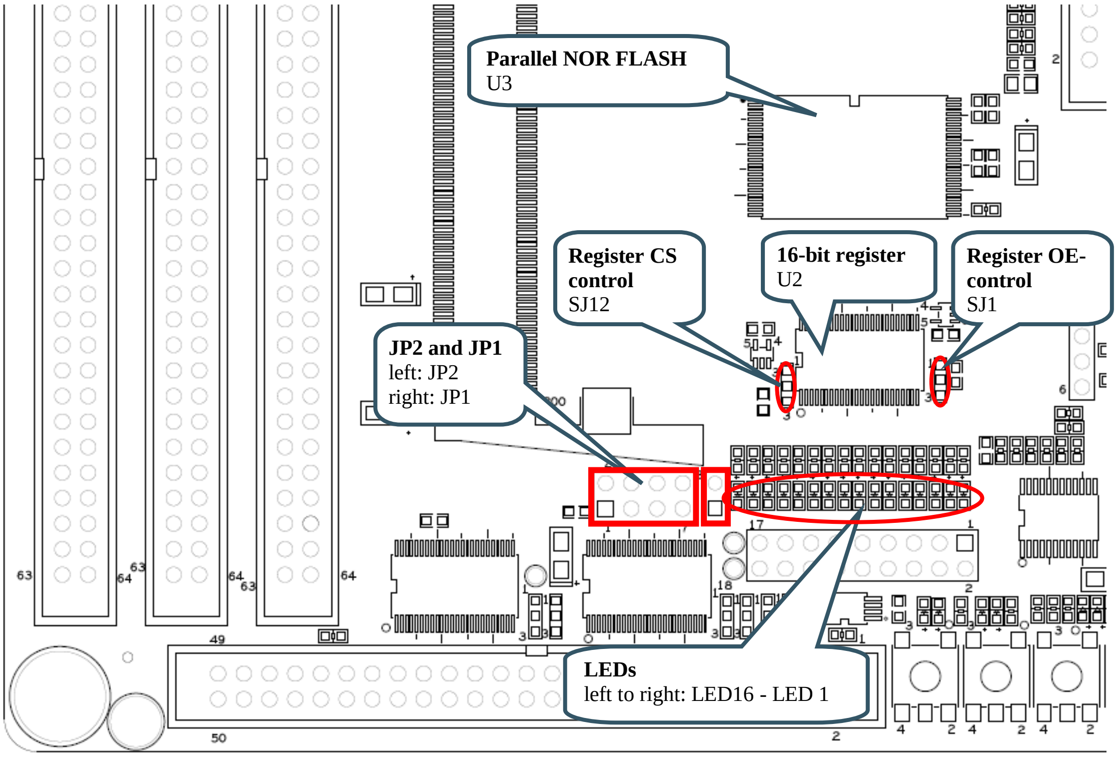


Figure 4 – External Memory Bus Circuit

4.6 SP5: Debug Interfaces

The multiple debug interfaces can look complex, but that is just because the board supports many different OEM Boards. The connectors are:

- J7 – this is the new and smaller footprint standard ARM debug connector. It has 2x5 pins in 50 mil pitch. The connector supports both the SWD and JTAG interfaces. Note that not all, and in particular older, JTAG debug probes do not support the SWD interface standard. Note where pin 1 is found (see picture below) for this connector.
- J8 – this is the old and big footprint standard ARM debug connector. It has 2x10 pins in 100 mil pitch. The connector supports both the SWD and JTAG interfaces. Note that not all, and in particular older, JTAG debug probes do not support the SWD interface.
- J9 – this is the old and big footprint 38 pin Mictor connector for ETM trace for the LPC2478. This connector is not mounted. The connector can be soldered to the board if needed. The connector can be bought from Tyco Electronics Amp and is a 38-way receptacle Mictor connector, 0.025" pitch, part number: 767054-1 or 2-5767004-2 (RoHS compliant). This connector is not used when working with LPC3250 OEM boards.
- J10 – this is the new and smaller footprint standard ARM Cortex-Mx connector for trace. It carries the trace signals as well as the debug signals found on J7. This connector is not used when working with LPC3250 OEM boards.

When working with the LPC3250 OEM Board, J7 is typically used. If an older and big footprint JTAG debug pod is used, J8 can alternatively be used.

JP3, J9 and J10 are not used when working with LPC3250 OEM boards.

JP4 shall be open (no jumper inserted) for enabling the JTAG interface of the LPC3250 OEM board.

Also note that the LPC3250 has a JTAG interface. Not the newer SWD interface.

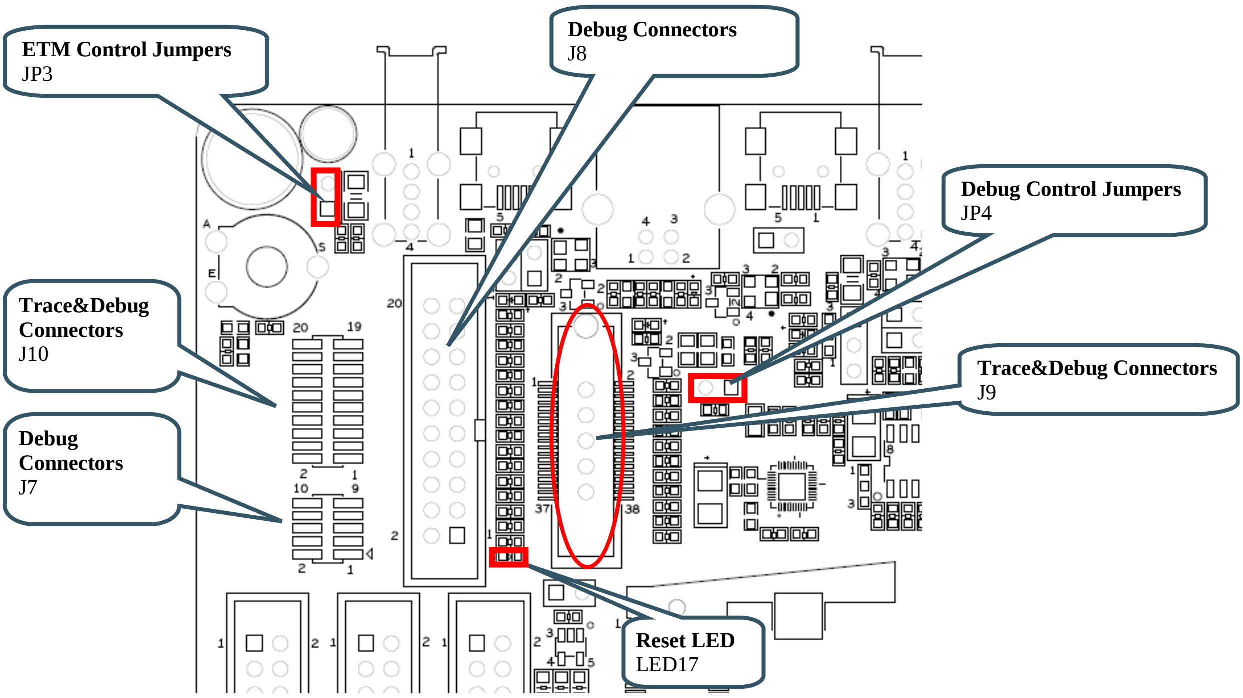


Figure 5 – Debug Interfaces

4.7 SP6: Ethernet Interface

The board has an Ethernet interface, J11, which is a RJ45 connector with integrated magnetics. There is also provision on the board for connecting a PoE interface. All signals can be accessed via expansion pads, J12. Five 0 ohm resistors have to be removed in that case for isolating J11 from the new connector (this is because J11 is not capable of handling a PoE interface so a new RJ45 connector must added). See picture below for where to find the relevant components on the board.

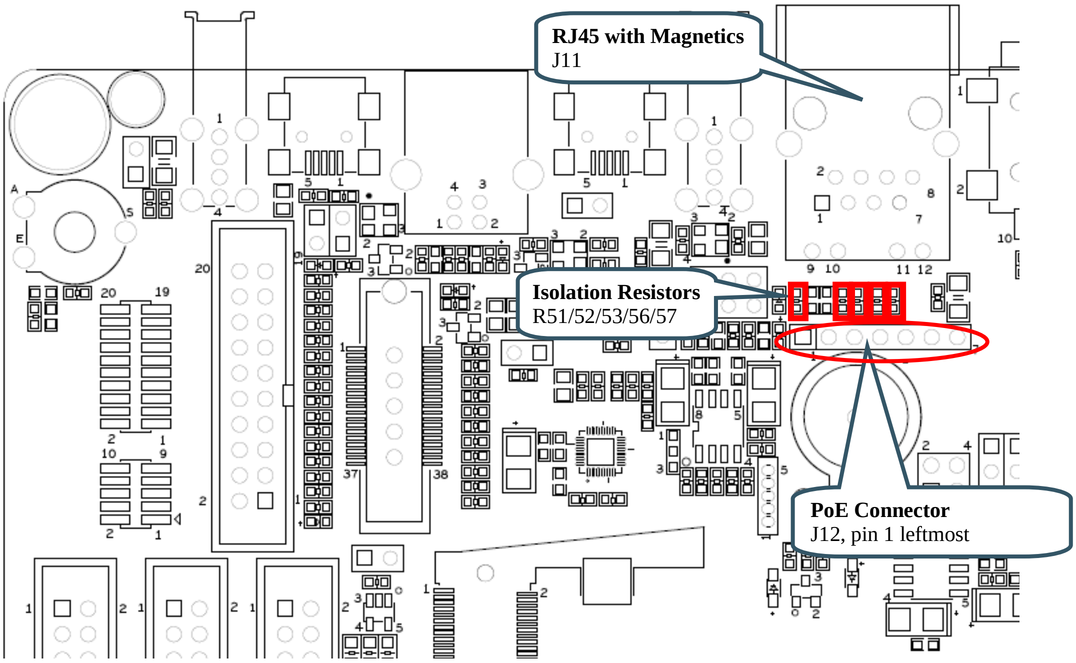


Figure 6 – Debug Interfaces

PoE modules that can be used for testing can for example be found at http://www.silvertel.com/poe_products.htm. Select a version delivering +5V with enough current capability for the specific application in mind. For example, using USB Host with power hungry external devices will require more current. In most situations a 9-10W module will be sufficient.

Besides the PoE module, a RJ45 capable of handling PoE is also needed.

4.8 SP6: SD/MMC Memory Card Interface

The board has a SD/MMC memory card interface, J13. Supply voltage to the external memory card is controlled via Q1. The Card Detection (CD) and Write Protect (WP) signals are connected to the I2C port expander on schematic page 7. There is also visual indication of supply voltage and the CD and WP signals via LEDs, see picture below where to find the LEDs on the board. Note that the Write Protect-LED is actually inverted. It is on when the memory card is not write protected and off when it is write protected.

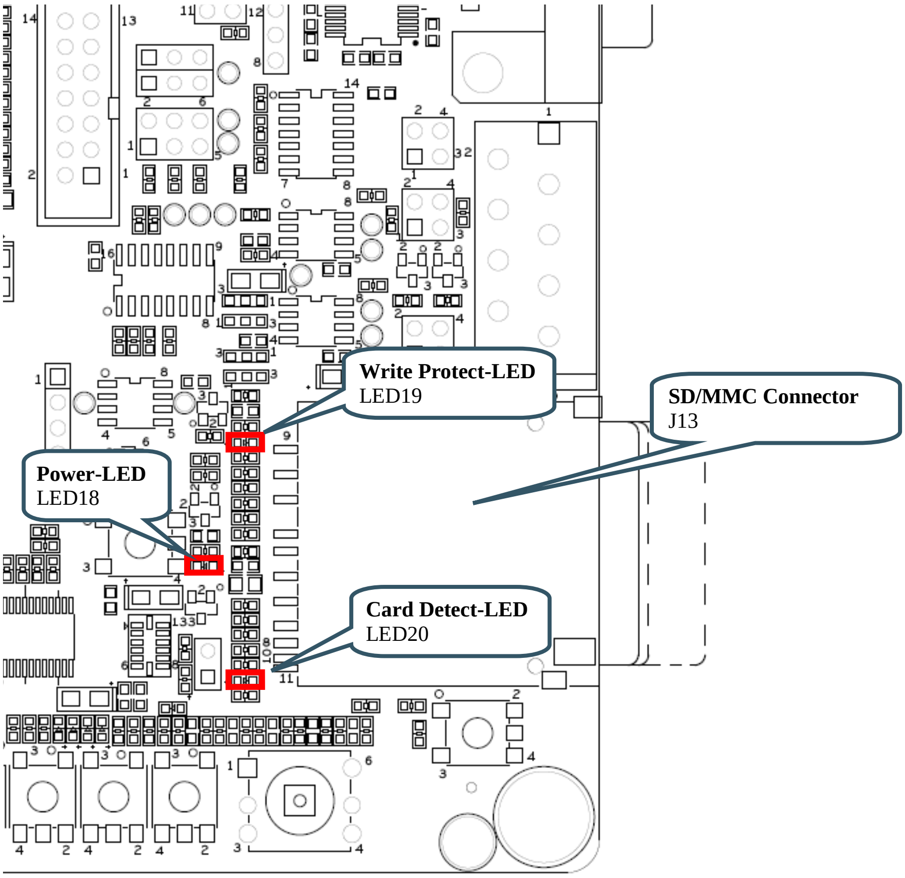


Figure 7 – SD/MMC Memory Card Interface

4.9 SP6: VBAT/ALARM Handling

The board can power the VBAT input supply (to the OEM board) from two different sources:

- The +3.3V power supply, via D1 (when board is normally powered).
- A 3V Lithium CR1025 size coin battery, via D2. Note that battery is not included.

See the LPC3250 datasheet for details about VBAT voltage range.

The ALARM signal control LED21. It is the LPC3250 signal ONSW that is connected to this signal. Check the LPC3250 errata for details about some limitations on this signal.

Also note that LED21 will consume a lot of current from the battery and/or super-capacitor. Restrict on/high time to conserve energy.

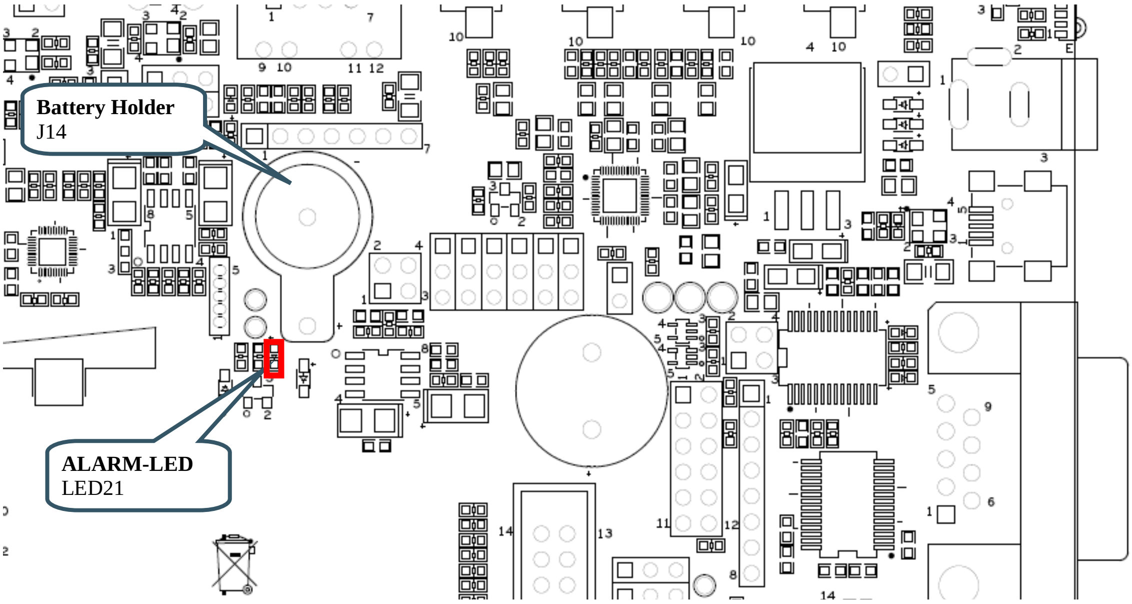


Figure 8 – VBAT and ALARM circuit

4.10 SP7: I2C Peripherals

There are several I2C peripherals on the board. See picture below for locating the different components on the board. The I2C addresses for the individual components are given in the schematic.

- Configuration E2PROM, 64kbit. This chip contains version information about the OEM Base Board.
- LM75 Temperature sensor.
- 3-axis Accelerometer (MMA7455). Note that the two interrupt outputs are not connected but available on JP5. **Note** that because the accelerometer is obsolete, it is not mounted on boards after March 2017.
- Port expander (PCA9532) with 8 LEDs and 4 pushbuttons. The Card Detect and Write Protect signals from the SD/MMC memory card connector are also connected to this chip. LED22/23 are positioned above SW2, LED24/25 are positioned above SW3, LED26/27 are positioned above SW4 and LED28/29 are positioned above SW5.

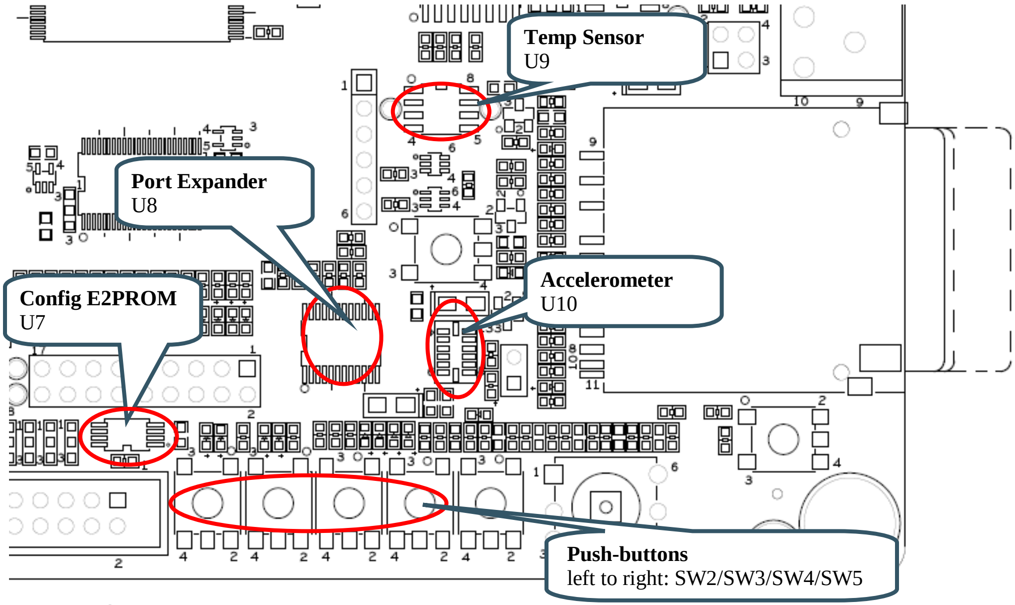


Figure 9 – I2C Peripherals

Signals	LPC3250
I2C-SDA	I2C1_SDA (channel#1)
I2C-SCL	I2C1_SCL (channel#1)

4.11 SP8: Analog Input

The board contains a trimming potentiometer (R94) for manually generating an adjustable voltage (between GND and VREF). See picture below where to locate the trimming potentiometer on the board. The table list which pin the adjustable voltage is connected to.

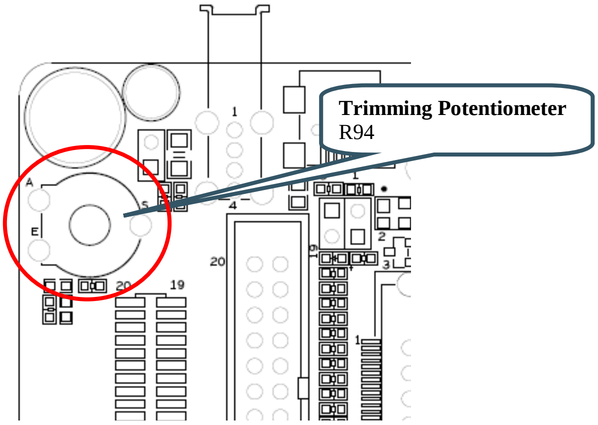


Figure 10 – Analog Input

Signals	LPC3250
GPIO39_AIN2	ADIN0

4.12 SP8: Digital IO

There is a push-button (SW6) that is connected to a signal that enable the service mode after reset on the OEM Board. For the LPC3250 this is pin GPI_01. If this pin is sampled low after reset, the uart service mode for the LPC3250 is entered (allowing program code download to internal SRAM via UART #5). LED30 is positioned above SW6 and will light when SW6 is pressed. It is also possible to control LED30 as an output from the OEM Board without damaging the output driver when SW6 is pressed. R110 limits the current.

There is also a 5-key joystick that directly connects to five general purpose input/output pins. See picture below for locating SW6 and SW7.

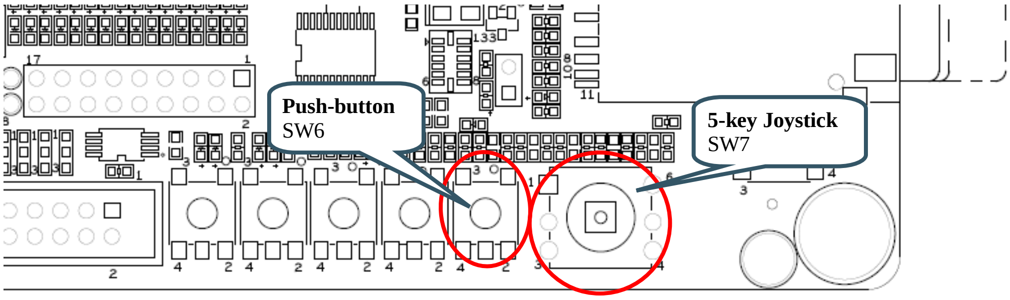


Figure 11 – Digital IO

Signals	LPC3250
GPIO_10	GPI_01
GPIO_73	P2.0
GPIO_74	P2.1
GPIO_75	P2.2
GPIO_76	P2.3
GPIO_77	P2.4

4.13 SP8: Serial Expansion Connector

The *Serial Expansion Connector* is a standardized serial interface connector that is included on many boards from Embedded Artists, including the OEM Base Board. The purpose is to provide a simple expansion connector for smaller expansion modules. Such modules are typically sensors of different kinds and communication modules, but can also be smaller displays.

The connector contains 14 pins that support SPI, UART and I2C communication. Four additional pins exist for specific functionality, like module reset, interrupt pins, analog signals and pwm signals. Power (3.3V) is also provided. Maximum current consumption of the external module is 250mA. All signals are protected with 470 ohm series resistors to minimize current in case of shorts to ground, +3.3V, or similar.

Usage of the different signals is specific for each module connected. All signals can be configured as either main function or alternatively as a general purpose input/output signal (GPIO). The picture below show where the connector can be found and the table lists which pins are connected. The application program has the responsibility to program the individual pins to correct state/function.

Note that the UART channel is not directly connected to a UART channel on the LPC3250. It is multiplexed with other UART channels, see section 4.15 for details. Also note that the *Serial Expansion Connector* and the RF-module (see section 4.14) share the same UART channel. Once the RF module is soldered, the *Serial Expansion Connector* cannot be used for UART communication.

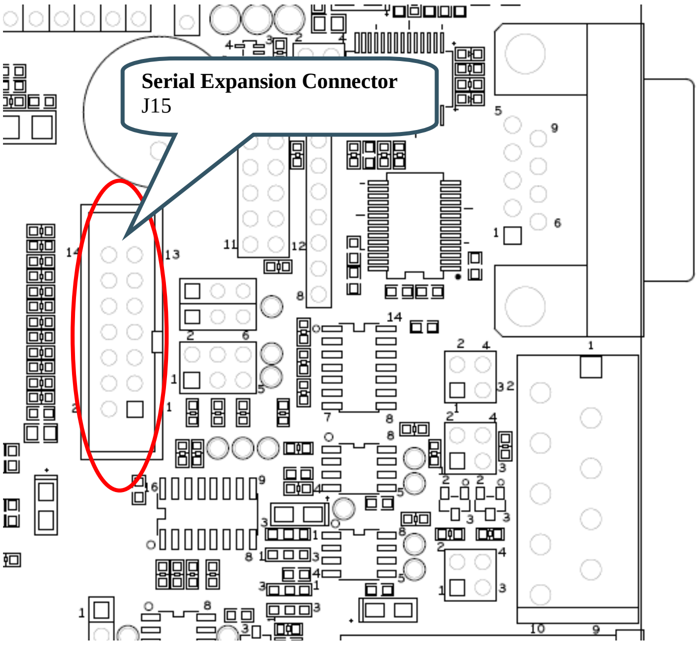


Figure 12 – Serial Expansion Connector

Signals	LPC3250
GPIO29_SPI-CLK	SPI1_CLK
GPIO31_SPI-MISO	SPI1_DATIN
GPIO32_SPI-MOSI	SPI1_DATIO
GPIO69	P2.7
SIE_UART_RXD	U1_RX, UART#1 is used
SIE_UART_TXD	U1_TX, UART#1 is used
I2C-SCL	I2C1_SCL
I2C-SDA	I2C1_SDA
GPIO42	GPO_14
GPIO72	GPI_07
GPIO37_AIN0	AIN0
GPIO38_AIN1	AIN1

4.14 SP8: NXP/Jennic RF Module Interface

There are pads on the backside of the OEM Base Board for soldering a JN5148 NXP (former Jennic) RF module. These pads are opposite of the SD/MMC memory card interface connector. There is also provision on the board for programming the RF module. This is done via 6-pos pin list (J31) where a FTDI programming cable is connected (TTL to USB Serial Converter cable, TTL-232R-3V3, see for example Digikey: 768-1015-ND). During programming, multiplexor U29 and U30 connects the UART channel directly to the programming cable, via J31. The RF-module can be reset via SW1. Program mode is entered by pressing SW9 during (and shortly after) reset.

Note that the UART channel is not directly connected to a UART channel on the LPC3250. It is multiplexed with other UART channels, see section 4.15 for details. Also note that the *Serial Expansion Connector* (see section 4.13) and the RF-module share the same UART channel. Once the RF module is soldered, the *Serial Expansion Connector* cannot be used for UART communication.

See picture below for locating the relevant components on the board. See also the table for signal connection.

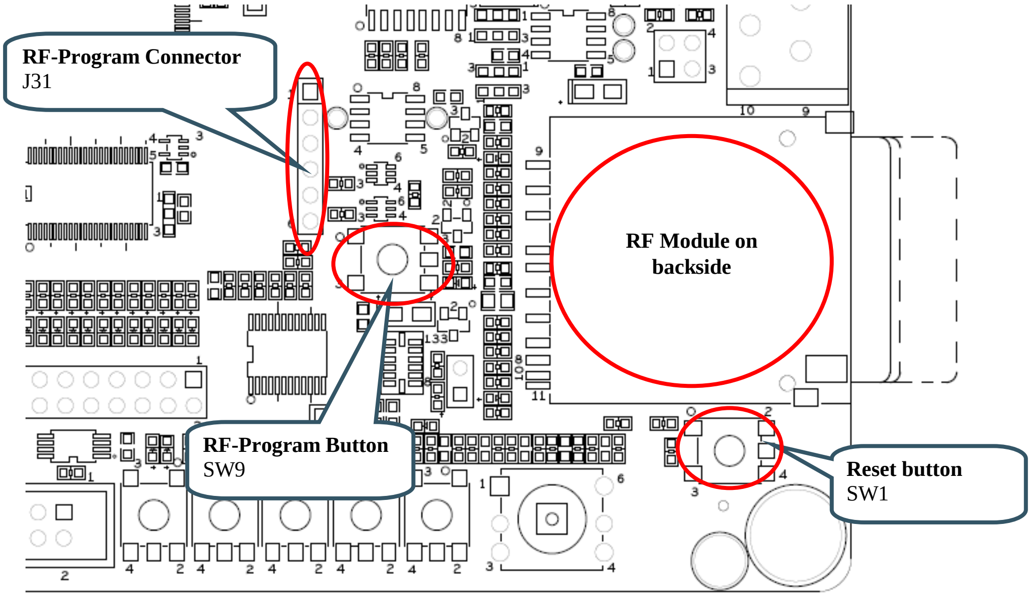


Figure 13 – RF Module

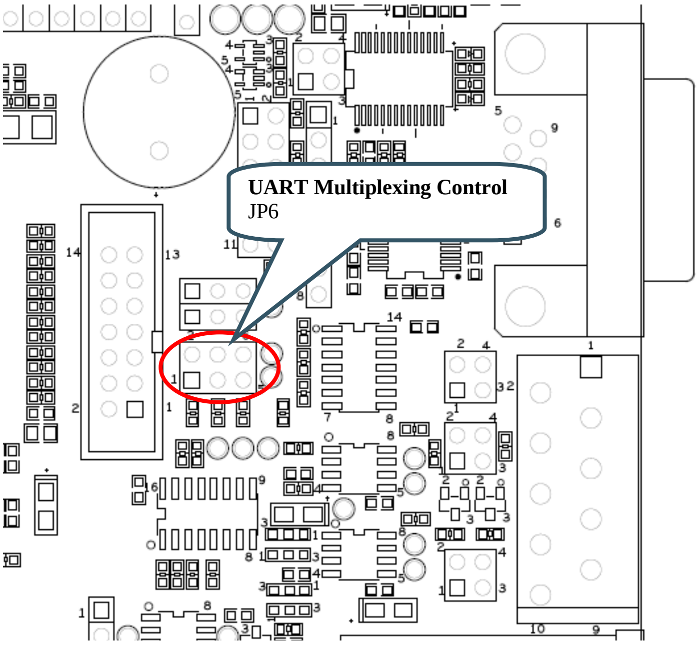
Signals	LPC3250
SIE_UART_RXD	U1_RX, UART#1 is used
SIE_UART_TXD	U1_TX, UART#1 is used

4.15 SP9: UART Multiplexing

UART channels from four sources are multiplexed into one UART channel, which is connected to the OEM Board:

- Serial Expansion Connector / RF module. Note that if the RF module is soldered to the board, then UART channel on the Serial Expansion Connector is occupied.
- LCD Expansion connector
- RS422/485 Interface
- RS232 Interface

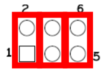
The multiplexing is static, in the sense that a change in the multiplexor setting requires jumpers to be moved. It is not controlled from the OEM Board. The picture below shows where to find the pin list to control multiplexing and the different settings. The table lists signal connection.



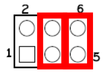
UART Multiplexing Control
JP6

Jumper settings for JP6

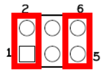
UART to Serial Interface Expansion Connector / RF-module (default)



UART to LCD Expansion Connector



UART to RS422/485 transceiver



UART to RS232 Interface

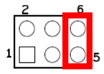


Figure 14 – UART Multiplexing

Signals	LPC3250
GPIO25_RXD	U1_RX, UART#1 is used
GPIO24_TXD	U1_TX, UART#1 is used

4.16 SP9: RS232 Interface

There is a RS232 interface with all modem signals on the board. The RS232 interface is available on a 9-pos male DSUB. The OEM Base Board is a DTE (Data Terminal Equipment). An external device can be a DCE (Data Communications Equipment). For connecting DTE-to-DCE a straight-thru cable shall be used. An external device can also be a DTE. For connecting DTE-to-DTE a null-modem cable shall be used (also called a crossover cable). A PC is normally also a DTE so a null-modem cable shall be used for communicating with a PC.

See picture below for locating relevant components on the board. The table below lists signal connections. On the LPC3250 it is possible to connect UART #1 (just rx and tx) to the RS232 transceiver or UART#3 with all modem signals. Insert jumpers in all JP11 positions and place jumpers in left position in JP12/JP13 to connect UART #3 (in full modem configuration) to the RS232 interface. Remove all jumpers in JP11 and place jumpers in right position in JP12/JP13 to connect UART #1 (just rx and tx) to the RS232 interface.

The interface component used (SP3243E) has built-in ESD protection.

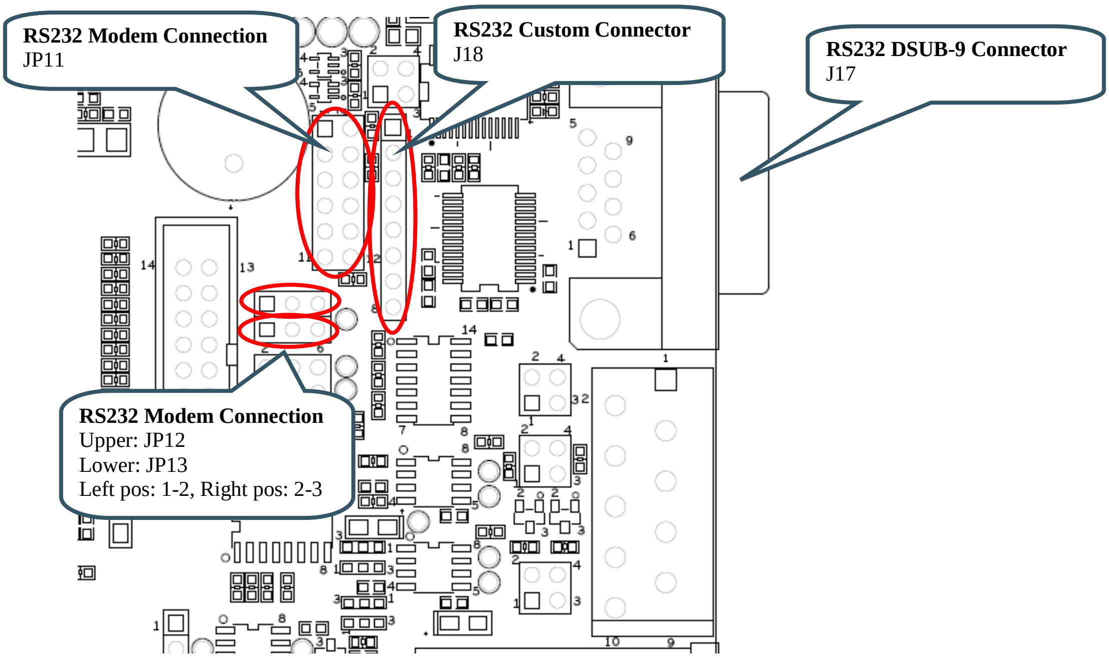


Figure 15 – RS232 Interface

Signals	LPC3250
GPIO25_RXD	U1_RX, UART#1 is used
GPIO24_TXD	U1_TX, UART#1 is used
GPIO83_RTS_BD30	U3_HRTS / U2_RTS / GPO_23
GPIO92_DTR_BD21	U3_DTR / U2_TX
GPIO94_DCD_BD19	U3_DCD / GPI_05
GPIO93_DSR_BD20	U3_DSR / U2_RX
GPIO95_CTS_BD18	U3_CTS / U2_CTS / GPI_16
GPIO91_RI_BD22	U3_RI / GPI_28
GPIO97_TXD_BD16	U3_TX
GPIO96_RXD_BD17	U3_RX / P3.17

4.17 SP9: RS422/485 Interface

There is a RS422/485 interface on the board. RS422 has individual transmit and receive pairs, while RS485 share the same signal pair (and is hence half-duplex). By inserting both jumpers in JP9, the two signal pairs are connected and a RS485 interface is created. By removing the jumpers in JP9 a RS422 interface is created. Via JP10 it is also possible to add termination resistors, if needed.

Direction is controlled via signal GPIO43 and is by default half-duplex (i.e., no simultaneous transmit and receive). A high level enables the transmitter while a low level enables the receiver. It is possible to remove R135 and R136 and control the transmitter and receiver individually via JP7 and JP8.

Note that on LPC3250, signal GPI_03 is connected to GPIO43 in the base board. This is an input signal in the LPC3250, whereas an output signal is what is needed to control GPIO43. This is not supported by the OEM base board, due to trade-offs when making different OEM boards and the base board compatible. **The RS422/485 interface is not directly supported for the LPC3250.** Another free GPO/GPIO pin has to be used for controlling GPIO43 and this requires re-wiring on the OEM base board.

The interface component used (SN65HVD35) has built-in ESD protection.

See picture below for locating relevant components. The table below lists signal connections.

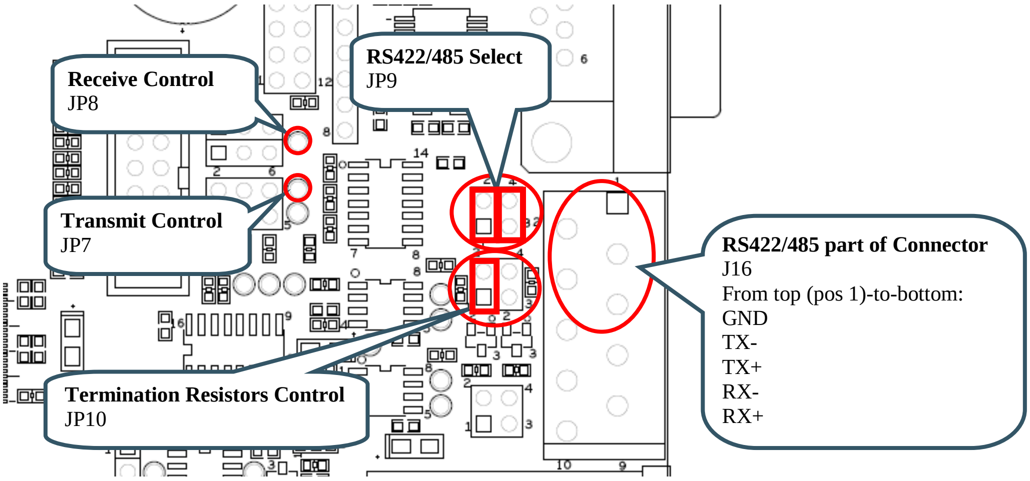

Receive Control
JP8
RS422/485 Select
JP9
Transmit Control
JP7
RS422/485 part of Connector
J16
From top (pos 1)-to-bottom:
GND
TX-
TX+
RX-
RX+
Termination Resistors Control
JP10

Figure 16 – RS422/484 Interface

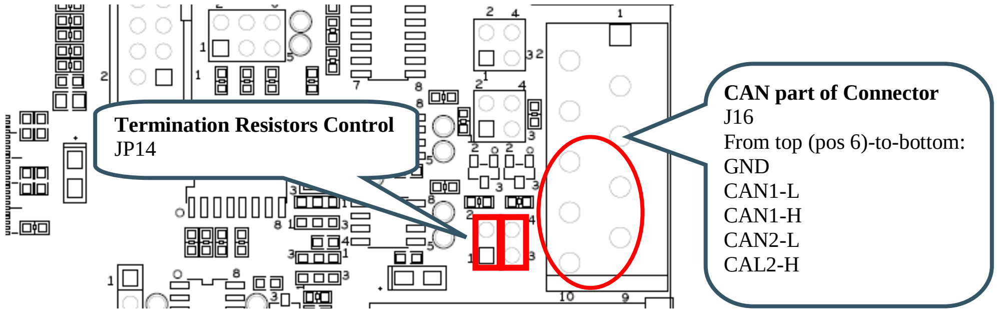
Signals	LPC3250
GPIO25_RXD	U1_RX, UART#1 is used
GPIO24_TXD	U1_TX, UART#1 is used
GPIO43	GPI_03

4.18 SP10: CAN

The CAN interface is not supported since there is no on-chip CAN peripheral on the LPC3250. For reference, the information about the interface is described below.

There is one CAN interface mounted on the board. The board is also prepared for a second CAN interface, if needed. Via JP16 it is also possible to add termination resistors, if needed. The interface has on-board ESD protection.

See picture below for locating relevant components. The table below lists signal connections.



Termination Resistors Control
JP14

CAN part of Connector
J16
From top (pos 6)-to-bottom:
GND
CAN1-L
CAN1-H
CAN2-L
CAL2-H

Figure 17 – CAN Interface

Signals	LPC3250
GPIO14_CANRD	U6_IRTX. Note: not a CAN interface
GPIO15_CANTD	U6_IRRX. Note: not a CAN interface
GPIO18_CANRD	GPO_02. Note: not a CAN interface
GPIO19_CANTD	GPO_03. Note: not a CAN interface

4.19 SP10: IrDA

The board is prepared for an IrDA interface, but the IrDA transceiver (TFBS4652) is not mounted. To enable the IrDA interface, mount U18 (TFBS4652) and adjust SJ2 and SJ3 (connect 2-3 pads). See picture below for locating relevant components. The table below lists signal connections.

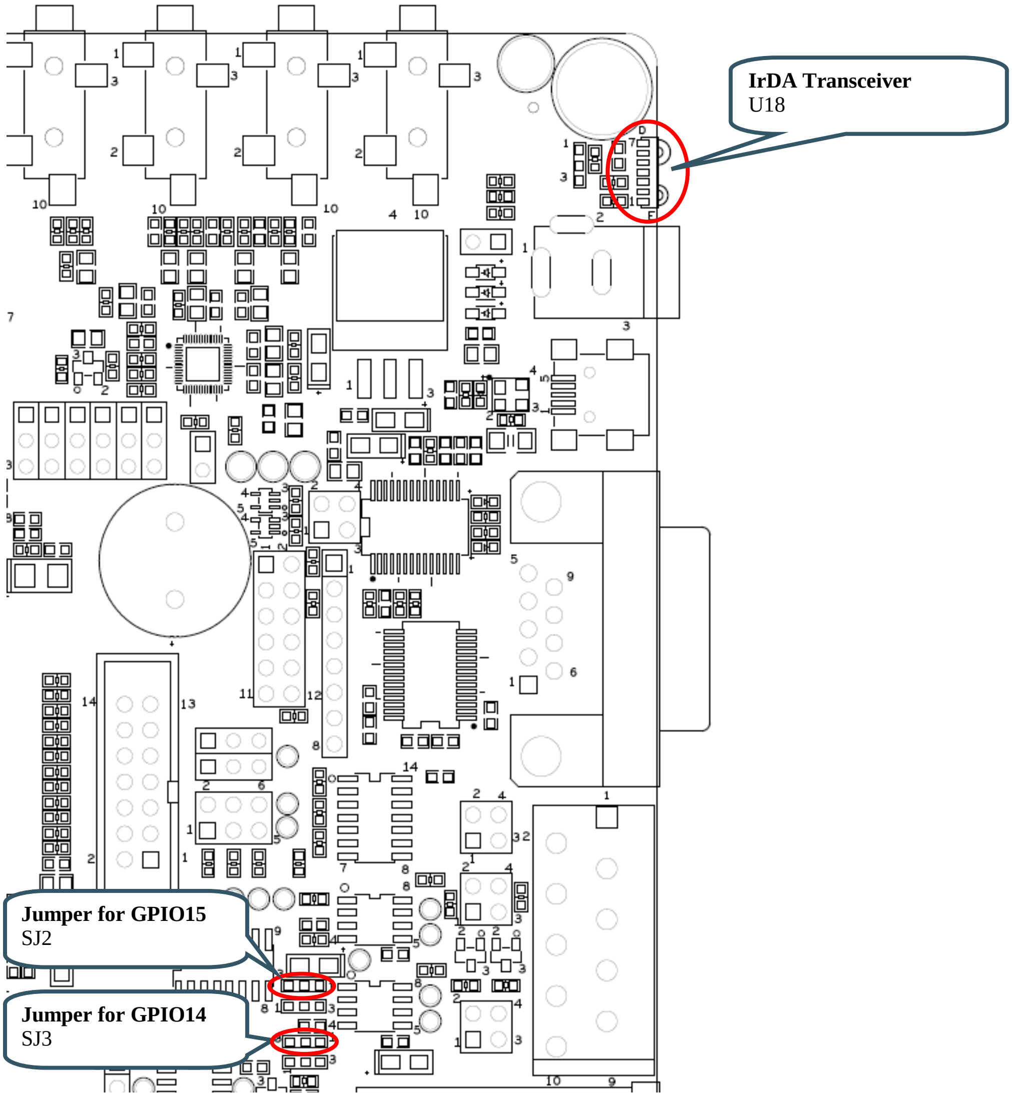


Figure 18 – IrDA Interface

Signals	LPC3250
GPIO14_CANRD	U6_IRTX
GPIO15_CANTD	U6_IRRX
GPIO69	P2.7

4.20 SP11: USB Channel 1

The USB1 interface is not supported since there is one on-chip USB interface on the LPC3250 and this is connected to the USB2 interface on the OEM base board. For reference, the information about the interface is described below. The interface cannot be used.

The USB1 interface offers a USB Host or USB OTG interface. Only one of these interfaces can be used at a time, i.e., both cannot be used simultaneously. There is an external USB OTG transceiver, U31. This transceiver is used for OTG specific signaling but also for controlling the 1.5Kohm pull-up and 15Kohm pull-down resistor control that is needed for USB Host/Device. If just USB Device is needed, it is possible to insert a jumper in JP15 to statically attach a 1.5Kohm pull-up resistor to the USB-DP signal. LED31 can be used to signal USB-up state.

USB Host power is controlled via distribution switch U20. Pads 1-2 on SJ5 are by default connected to always enable +5V on VBUS channel 1. LED34 light whenever +5V is present on VBUS1. VBUS feedback via GPIO46 is by default disconnected as well as status feedback from the distribution switch via GPIO51.

In case VBUS1 shall be actively controlled from the OEM Board, R190 and R198 must be soldered and pad 2-3 on SJ5 must be connected. Further, GPIO43/46/51 must not be used for other tasks in the design.

It is possible to force USB Host detection when using the USB OTG interface by inserting a jumper on JP16.

See picture below for locating relevant components. The table below lists signal connections.

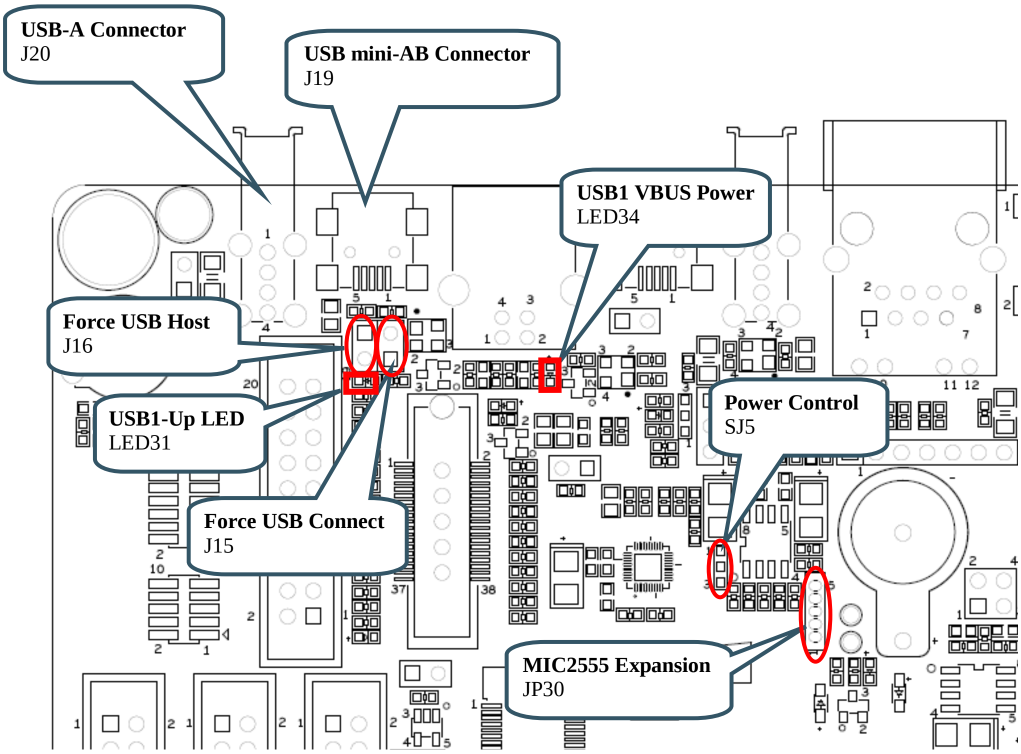


Figure 19 – USB Channel#1

Signals	LPC3250
USB1-DP	Not connected
USB1-DM	Not connected

GPIO42	GPO_14. Not used for USB1 interface
GPIO51	SPI2_DATIN. Not used for USB1 interface
GPIO52	GPIO_04. Not used for USB1 interface
GPIO53	SPI2_CLK. Not used for USB1 interface
GPIO46	P0.6 normally not connected. Not used for USB1 interface
GPIO43	GPI_03 normally not connected. Not used for USB1 interface

4.20.1 USB Channel 1 as USB Host

When USB channel #1 is used as USB Host, 15Kohm pull-down resistors are needed on the USB data signals. These are activated inside the USB OTG chip (U31), and this has to be done via the I2C interface of GPIO52/GPIO53.

J20 is the connector to use when USB Host is used. In order to provide +5V to the external USB device connected to this connector (J20), channel A of U20 must be enabled. It is enabled by default since SJ5 is normally connected between pin 1-2. LED34 lights green when +5V is available on J20.

JP15 shall not be inserted. JP16 has no effect.

4.20.2 USB Channel 1 as USB Device

When USB channel #1 is used as USB Device, a 1.5Kohm pull-up resistor is needed on the USB DP data signal. There are two methods to create this. JP15 is inserted and the pull-up resistor is always enabled. Alternatively, the pull-up resistor is activated inside the USB OTG chip (U31), and this has to be done via the I2C interface of GPIO52/GPIO53. In the latter case, JP15 shall not be inserted.

J19 is the connector to use when USB Device is used. Normally it should be a USB-B connector for creating a USB Device interface, but the mini-AB connector can also be used in this case. The status of VBUS can be read via U31.

JP16 shall not be inserted.

LED34 is has no meaning.

4.20.3 USB Channel 1 as USB OTG

When USB channel #1 is used as USB OTG, pull-up and pull-down resistors are controlled via the USB OTG chip (U31), and this has to be done via the I2C interface of GPIO52/GPIO53. Note that some LCD signals interfere with the I2C signals to the USB OTG chip. In particular note that R163 is typically not mounted on the *OEM Base Board*.

J19 is the connector to use when USB OTG is used. VBUS is controlled via U31.

JP15 shall not be inserted.

LED34 is has no meaning.

JP16 can be inserted in order to force USB Host behavior on the interface (if the external device connected to J19 cannot pull the ID signal low).

4.21 SP12: USB Channel 2

This section describes the second USB interface on the board. **This is the interface that the USB interface on the LPC3250 is connected to.**

This interface has a USB Host connector (USB-A), a USB OTG (USB mini-B) and a USB Device connector (USB-B). One of these interfaces can be used at a time, i.e., both cannot be used simultaneously. The software on the LPC3250 OEM Board is also different between the interfaces.

For USB Device operation; insert jumpers in position 1-2 in JP17/JP18/JP19. GPIO28 controls USB connect functionality and LED32 lights when the USB Device is connected. SJ4 has pads 1-2 shorted by default. LED33 is controlled by GPIO27 and signals USB-up state. GPIO54 is used for VBUS sensing. It is also possible to control the "connect functionality" via the USB OTG transceiver that is located on the LPC3250 OEM board.

For USB Host operation; insert jumpers in position 2-3 in JP17/JP18/JP19. USB Host power is controlled via distribution switch U20 (found in schematic page 11). Signal GPIO26 is active low and enables +5V on VBUS2. LED35 light whenever +5V is present on VBUS2. GPIO55 is connected to status feedback from the distribution switch. GPIO54 is used for VBUS sensing. 15Kohm pull-down resistors are always active.

See picture below for locating relevant components. The table below lists signal connections.

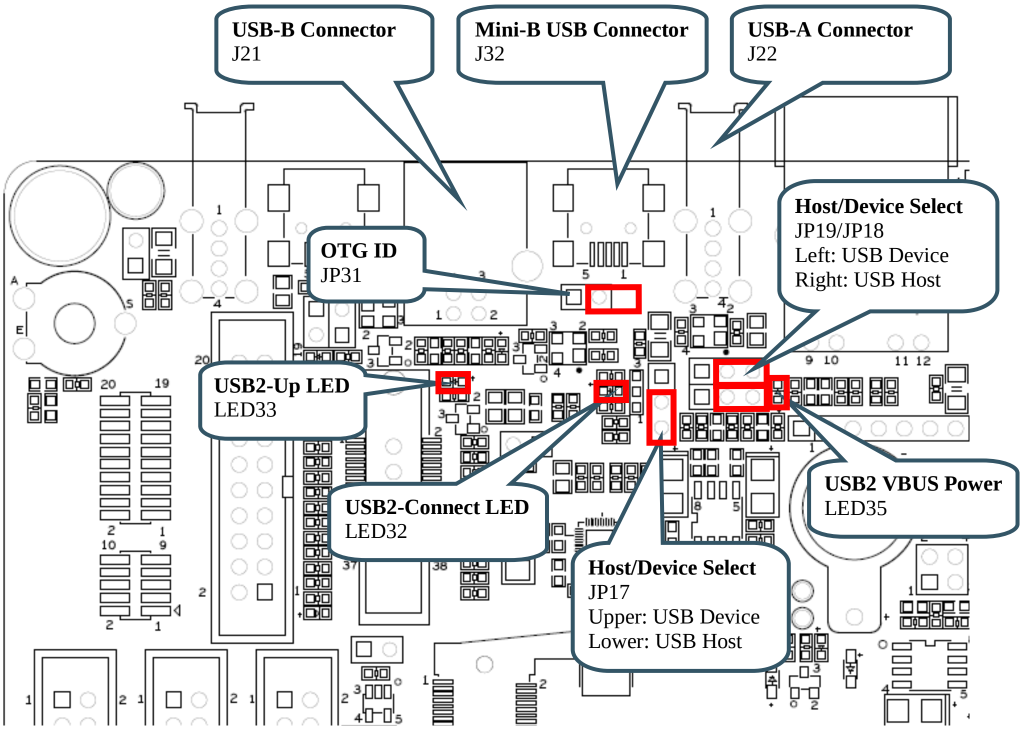


Figure 20 – USB Channel#2

Signals	LPC3250
USB2-DP	USB-DP from USB OTG transceiver on LPC3250 OEM board
USB2-DM	USB-DM from USB OTG transceiver on LPC3250 OEM board
GPIO27	GPO_17
GPIO28	GPO_20
GPIO54	USB-VBUS from USB OTG transceiver on LPC3250 OEM board

GPIO26	USB-VBUS-CTRL from USB OTG transceiver on LPC3250 OEM board
GPIO55	GPI_02
GPIO66	USB-ID to USB OTG transceiver on LPC3250 OEM board

4.21.1 USB Channel 2 as USB Host

When USB channel #2 is used as USB Host, JP17/JP18/JP19 shall all be set in position 2-3.

J22 is the connector to use when USB Host is used. In order to provide +5V to the external USB device connected to this connector (J20), channel B of U20 must be enabled. VBUS2 is enabled by GPIO26 (active low). Over-current status back is provided via GPIO55. LED35 lights green when +5V is available on J22. The VBUS2 signal is also feed back to GPIO54.

JP31 has no effect.

LED33 can signal that an external USB device has been connected.

4.21.2 USB Channel 2 as USB Device

When USB channel #2 is used as USB Device, JP17/JP18/JP19 shall all be set in position 1-2. Also a 1.5Kohm pull-up resistor is needed on the USB DP data signal. GPIO28 controls this. Alternatively, SJ4 pin 2-3 can be connected to always enable this pull-up resistor. LED32 lights when the pull-up resistor is enabled.

J21 is the connector to use when USB Device is used. This is a USB-B connector. The VBUS signal is also feed back to GPIO54.

LED35 is has no meaning.

JP31 has no effect.

4.21.3 USB Channel 2 as USB OTG

When USB channel #2 is used as USB OTG, pull-up and pull-down resistors are controlled via the USB OTG chip on the LPC3250 OEM board. Note however that 15Kohm pull-down resistors are always connected (R187/R188). If this is a problem, remove these resistors.

J32 is the connector to use when USB OTG is used.

JP17/JP18/JP19 shall all be set in position 2-3.

JP31 shall be inserted.

LED32/33 have no meaning.

4.22 SP13: UART-to-USB Bridge

There is a UART-to-USB bridge on the board. This is to simplify connection to a PC because serial ports are not so common any more, especially not on laptops. The USB port also offers the possibility to power the board. It is UART#5 that is connected to the USB channel. This UART is commonly used as the console channel for applications. Printf() output is for example typically directed to this UART channel.

The UART-to-USB bridge is based on the chip FT232RL from FTDI. A driver is typically needed to be installed on the PC side. The driver creates a virtual COM port on the PC that represents the UART channel. Any program on the PC can connect to this COM port for communication with the LPC3250 UART channel. There are two LEDs (Transmit – LED38 and Receive – LED37) that signal communication activity.

It is possible to automatically enable service mode (for program download via UART). By inserting two jumpers in JP20, the RTS signal can control signal GPI_01 and DTR can control reset. By pulling GPI_01 low during a reset cycle, service mode is entered. Note that some terminal programs controls the DTR/RTS signals so that the board is always in reset and/or always enter service mode. It this is the case, just remove both JP20 jumpers.

See picture below for locating relevant components. The table below lists signal connections.

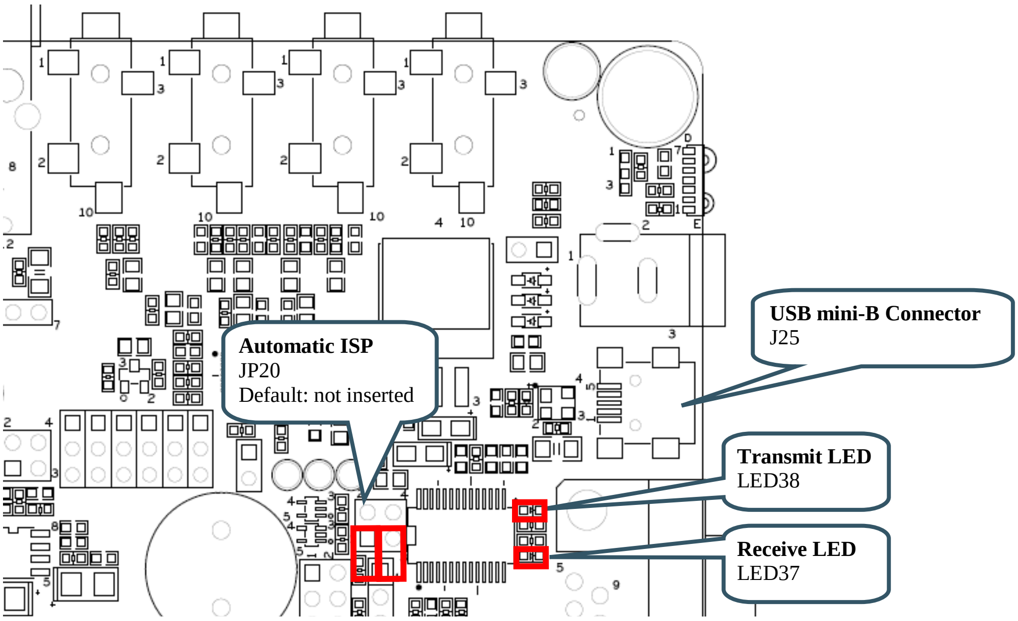


Figure 21 – UART-to-USB Bridge

Signals	LPC3250
GPIO16_TXD	U5_TX
GPIO17_RXD	U5_RX
GPIO10	GPI_01

4.23 SP13: Power Supply

The power supply on the board is straight forward. There are three powering sources:

1. +5V DC via 2.1mm power jack (J24). Center pin positive. There is also an alternative connector (J23) for this powering option.
2. USB powering via the USB-to-UART Bridge connector (J25).
3. +5V DC power form the PoE (Power-over-Ethernet) connector (J12).

In all cases, a stable +5V DC voltage powers the board. Current consumption depends on USB Host usage. If USB Host is not used (or used with very low external current consumption), a 1 Amp power supply (5W) is all what is needed. If USB Host is used with high external current consumption a 2-2.5 Amp power supply (10-12.5W) is needed.

The second powering option, via the USB-to-UART connector will work in most cases. Note however that not all PC:s and laptops can provide this amount of current. An external USB Hub can be used in that case or an external power supply.

U21 is a 1.5A +3.3V linear regulator. LED36 signals presence of regulated +3.3V. Three voltage measurement pads exist for verifying correct voltage levels on the board. See picture below for locating relevant components. The table below lists signal connections.

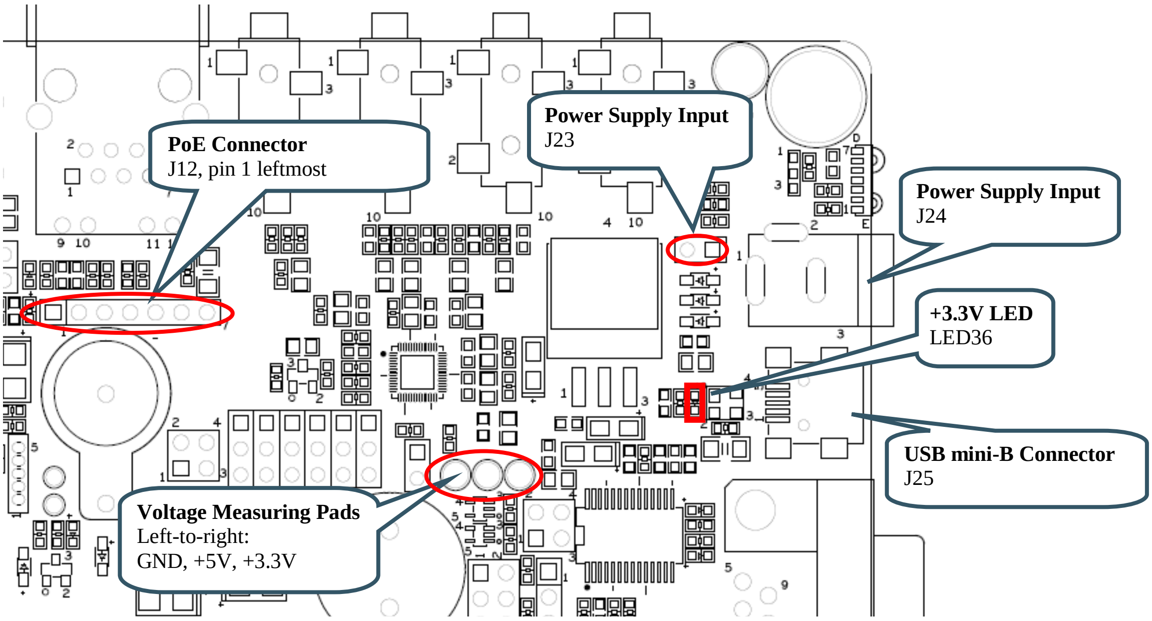


Figure 22 – Power Supply

4.24 SP14: LCD Expansion Interface

The board contains a buffered LCD expansion interface via a 50 pos IDC connector. The expansion interface also includes UART (see UART multiplexing), I2C and SPI interfaces. These additional interfaces are for identifying external displays (via configuration I2C-E2PROM) as well as touch screen controllers. An external LCD pixel clock can also be supplied via the expansion connector.

The LCD expansion connector carries 18 data bits per pixel by default. (6 per RGB color). The LPC3250 LCD controller can produce 24 data bits per pixel and it is possible to output all these signals on the LCD expansion connector. The trade-off is that the UART and I2C serial interfaces have to be removed. Via SJ6-SJ11 it is possible to select what signals to make available on the LCD expansion connector. By default pad 1-2 are connected on SJ6-SJ11.

On rev B of the OEM base board, JP34 has been added to support the LPC3250 OEM board. JP34 shall be in position 2-3 to route signal LCD_VD18 properly. On the generic schematic this signal is called LCD_VD3 but for the LPC3250 it is LCD_VD18. The reason for this is for backward compatibility with previous OEM board designs. **Note that the default setting for JP34 is 1-2 and not 2-3, which it should be for LPC3250.**

For performance reasons (on the LPC3250), a system with 16-bit color information per pixel is typically what is implemented.

See picture below for locating relevant components. The table below lists signal connections.

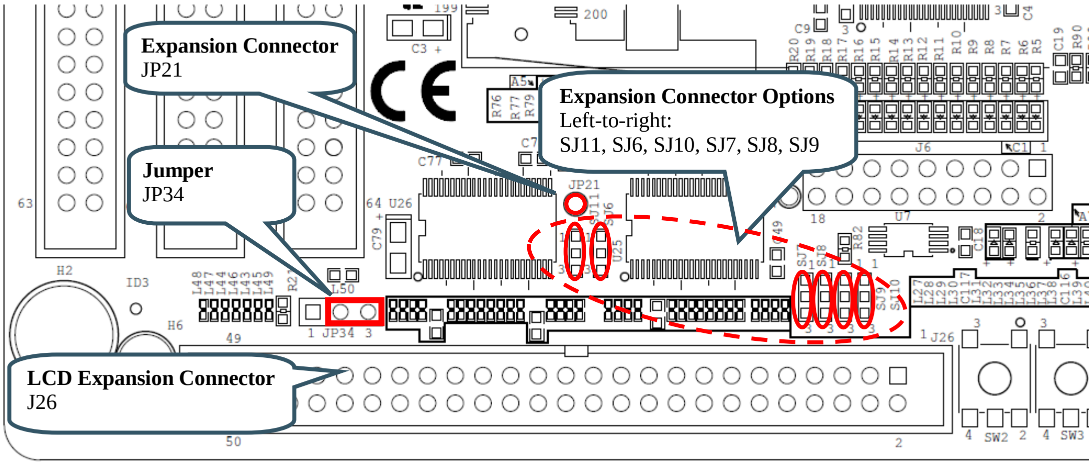


Figure 23 – LCD Expansion Interface

Signals	LPC3250
I2C-SDA	I2C1-SDA
I2C-SCL	I2C1-SCL
LCD_UART_RXD	U1_RX via multiplexor, UART#1 is used
LCD_UART_TXD	U1_TX via multiplexor, UART#1 is used
BSPI_SCK	SPI1_CLK
BSPI_MOSI	SPI1_DATIO
GPIO31	SPI1_DATIN
GPIO30	GPO_04
GPIO33	GPO_05
GPIO34	GPO_11

GPIO70	GPIO_00
GPIO42	GPO_14

4.25 SP15: I2S Audio Codec

The board contains an I2S audio codec based on UDA1380 from NXP. The codec is controlled via I2C and audio data is transferred over the I2S bus.

The OEM Base Board supports different OEM Boards and the I2S signals can be located at two different groups of signals from the OEM Board. Jumper JP22-JP27 selects which group of signals to connect to the I2S interface of the UDA1380. For LPC3250 OEM Board, set all jumpers in lower position (see picture below). **Note that this is not the default setting of the jumpers when the board is delivered.**

The audio codec provides a mono microphone input, a stereo line input, a stereo line output and a stereo headphone output. The board also has an on-board speaker. The right line output can be used to drive the speaker. Signal GPIO40 (GPO_00 from the LPC3250) is also connected to the speaker via JP29 (pos 1-2). By PWM-modulating this signal and audio signal can also be created.

See picture below for locating relevant components. The table below lists signal connections.

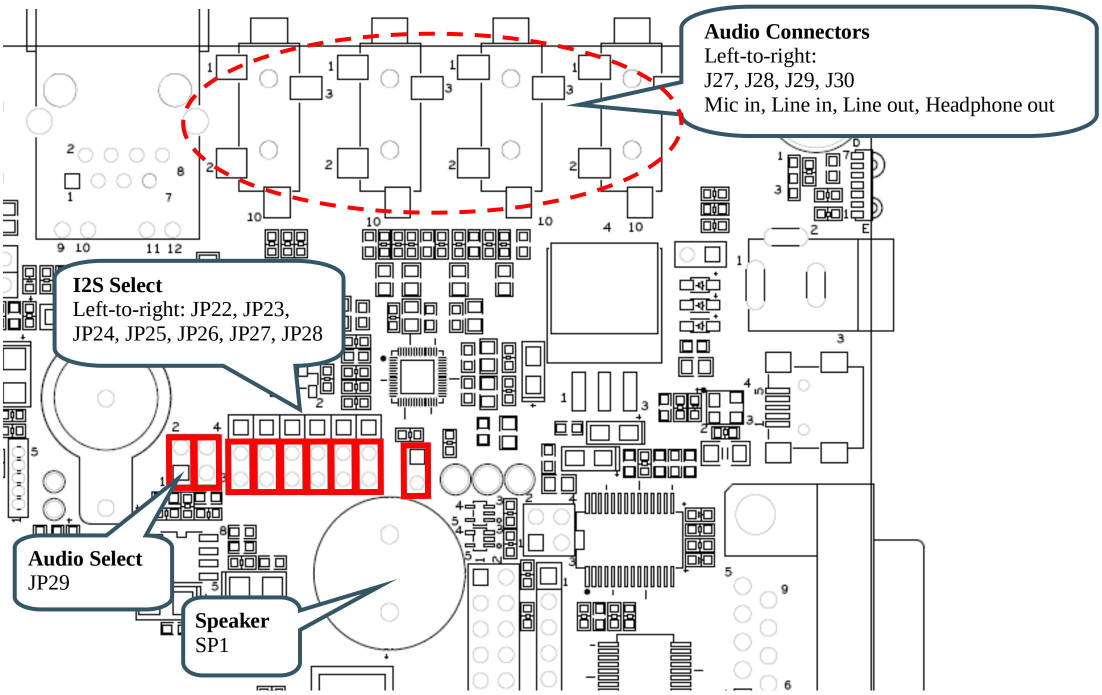


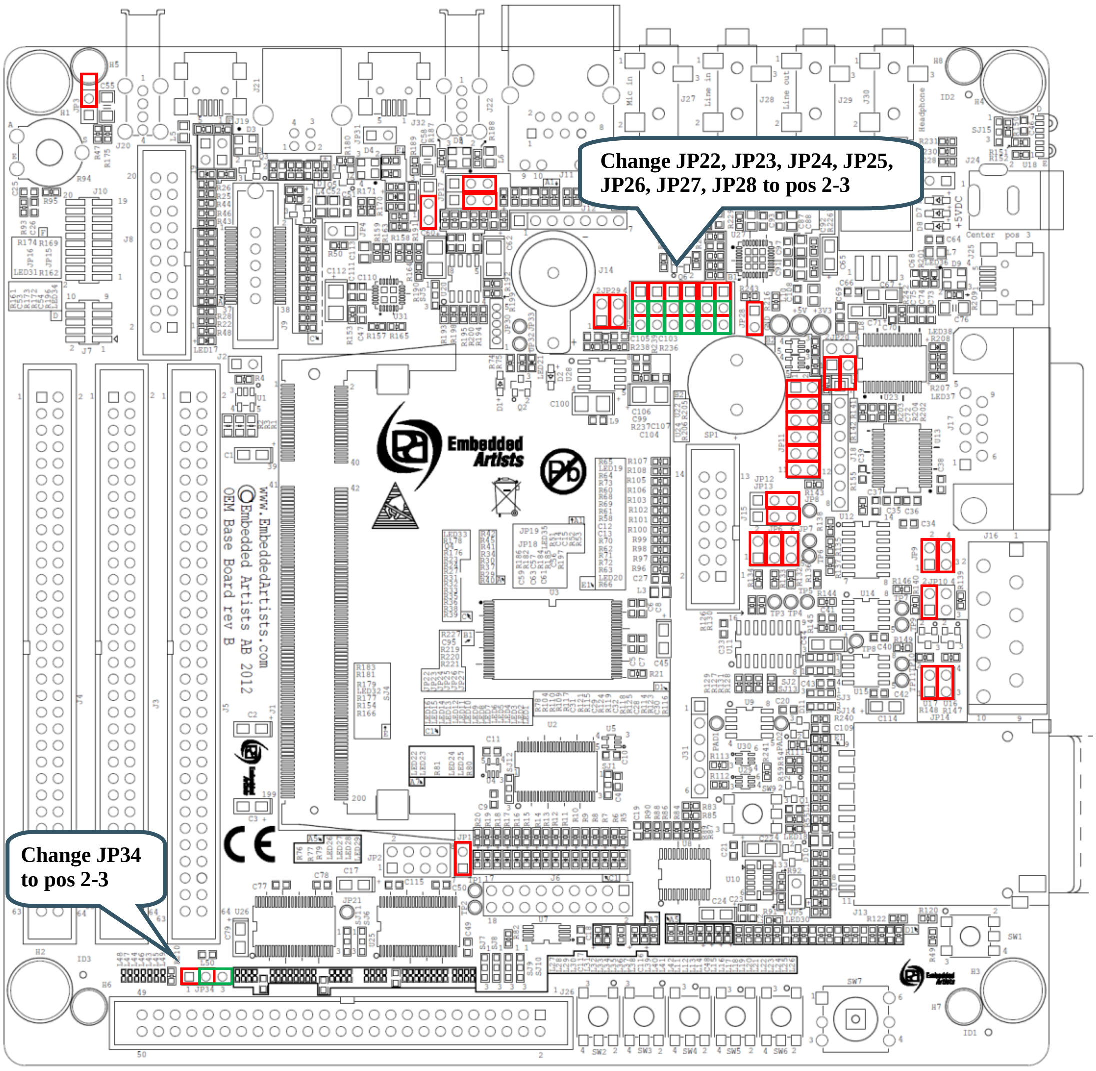
Figure 24 – I2S Audio Codec Interface

Signals	LPC3250
I2C-SDA	I2C1-SDA
I2C-SCL	I2C1-SCL
GPIO40	GPO_00
GPIO56	I2S1TX_CLK
GPIO57	I2S1TX_SDA
GPIO58	I2S1TX_WS

GPIO59	P0.0 / I2S1RX_CLK
GPIO60	P0.1 / I2S1RX_WS
GPIO61	GPI_00 / I2S1RX_SCA
GPIO68	TST_CLK2

4.26 Default Jumpers Positions

Figure 25 illustrates the default jumper positions as mounted when the board is delivered from Embedded Artists. Default jumpers are marked with red color.


Change JP22, JP23, JP24, JP25, JP26, JP27, JP28 to pos 2-3

Change JP34 to pos 2-3

Figure 25 – OEM Base Board Default Jumper Positions

Note that the default position for the six I2S jumpers is not what is needed for the LPC3250 OEM board. Also, JP34 is in wrong position. Before working with I2S audio and the LCD interface change the jumper settings. Jumpers needed to change are marked with green – this is the position they should have.

Some signals are used on different interfaces that cannot be active simultaneous. See table in next section for information about pin usage.

4.27 Usage of CPU Pins

Almost all pins of the LPC3250 are directly available on the expansion connectors. Only in a few cases are pins used for dedicated functionality like Ethernet interface and chip select signals. Such pins are not available on the expansion connector. The table below lists all pins and their possible restrictions.

OEM Base Board signal name	***LPC3250 Pin (OEM Board signal name)***	***Usage***
GPIO0	GPO_10	LCDPWR signal to LCD expansion connector.
GPIO1	GPO_12	LCDLE signal to LCD expansion connector.
GPIO2	GPO_13	LCDDCLK signal to LCD expansion connector.
GPIO3	GPO_15	LCDFP signal to LCD expansion connector.
GPIO4	GPO_16	LCDENAB signal to LCD expansion connector.
GPIO5	GPO_18	LCDLP signal to LCD expansion connector.
GPIO6	P0.2	LCD databit 4.
GPIO7	P0.3	LCD databit 5.
GPIO8	P0.4	LCD databit 6.
GPIO9	P0.5	LCD databit 7.
GPIO10	GPI_01	Connected to push-button (for enabling service mode). Also connects to LED (active low). Connects to USB-to-serial bridge (for automatic service mode functionality)
GPIO11	GPI_22	LCDCLKIN, an external clock signal can be feed to this pin.
GPIO12	GPO_06	LCD databit 18
GPIO13	LCDVD19 / PWMOUT2	LCD databit 19
GPIO14	U6_IRTX	Connects to RD1 for CAN channel #1, can also connect to IrDA transceiver. Note that the LPC3250 does not have a CAN peripheral.
GPIO15	U6_IRRX	Connects to TD1 for CAN channel #1, can also connect to IrDA transceiver. Note that the LPC3250 does not have a CAN peripheral.
GPIO16	U5_TX	Connects to USB-to-serial bridge (TxD on UART #5).
GPIO17	U5_RX	Connects to USB-to-serial bridge (RxD on UART #5).
GPIO18	GPO_02	LCD databit 0
GPIO19	GPO_03	LCD databit 1
GPIO20	GPO_08	LCD databit 8
GPIO21	GPO_09	LCD databit 9
GPIO22	LCDVD16 / PWMOUT1	LCD databit 16

GPIO23	HICORE / LCDVD17	LCD databit 17
GPIO24	U1_TX	Connects to UART multiplexing (TxD); further to LCD expansion connector, serial expansion connector, RF module, RS232 and RS422/485.
GPIO25	U1_RX	Connects to UART multiplexing (RxD); further to LCD expansion connector, serial expansion connector, RF module, RS232 and RS422/485.
GPIO26	USB_VBUS_CTRL	Connects to USB2 VBUS power enable.
GPIO27	GPO_17	Connects to USB2-up LED (active low).
GPIO28	GPO_20	Connects to USB2-connect functionality (for USB device operation).
GPIO29	SPI1_CLK	SPI-SCK. Connects to LCD expansion connector via buffer. Connects to serial expansion connector.
GPIO30	GPO_04	SPI-SSEL. Connects to LCD expansion connector.
GPIO31	SPI1_DATIN	SPI-MISO. Connects to LCD expansion connector. Connects to serial expansion connector.
GPIO32	SPI1_DATIO	SPI-MOSI. Connects to LCD expansion connector via buffer. Connects to serial expansion connector.
GPIO33	GPO_05	Connects to LCD expansion connector (GPIO function).
GPIO34	GPO_11	Connects to LCD expansion connector (GPIO function, touch controller spi-ssel).
GPIO35	TS_XP	No special usage on *OEM Base Board*.
GPIO36	TS_YP	No special usage on *OEM Base Board*.
GPIO37	AIN0	Connects to serial expansion connector.
GPIO38	AIN1	Connects to serial expansion connector.
GPIO39	AIN2	Connects to trimming potentiometer.
GPIO40	GPO_00	Connects to speaker output on AOUT signal. A PWM modulated signal can generate audio signals.
I2C-SDA	I2C1_SDA	I2C-SDA0, connects to PCA9532, E2PROM, LM74, 3-axis Accelerometer, audio codec, serial expansion connector, lcd expansion connector.
I2C-SCL	I2C1_SCL	I2C-SCL0, connects to PCA9532, E2PROM, LM74, 3-axis Accelerometer, audio codec, serial expansion connector, lcd expansion connector.
USB1-DP/DM	Not connected	Connects to USB Host/OTG interface (USB #1)
USB2-DP/DM	USB_CONN_DP/DM	Connects to USB Host/Device interface (USB #2)
MCICLK	MCICLK	Connects to MCICLK on SD/MMC connector
MCICMD	MCICMD	Connects to MCICMD on SD/MMC connector
MCIPWR	GPO_01	Connects to MCIPWR on SD/MMC connector
MCIDAT0	MCIDAT0	Connects to MCIDAT0 on SD/MMC connector

MCIDAT1	MCIDAT1	Connects to MCIDAT1 on SD/MMC connector
MCIDAT2	MCIDAT2	Connects to MCIDAT2 on SD/MMC connector
MCIDAT3	MCIDAT3	Connects to MCIDAT3 on SD/MMC connector
GPIO41	GPIO_05	No special usage on *OEM Base Board*, but is used for SPI_SSEL on the boot serial flash on the LPC3250 OEM board
GPIO42	GPO_14	Connects to LCD expansion connector (backlight control). Connects to USB1-up LED (active low). Connects to serial expansion connector.
GPIO43	GPI_03	Connects to RS422/485 transmit/receive control. Can optionally be connected to USB1 VBUS power enable. Note that the LPC3250 cannot control these functionalities on the OEM base board since GPIO43 is connected to an input (GPI_03) on the LPC3250.
GPIO44	U7_RX / LCDVD10	LCD databit 10
GPIO45	U7_TX / LCDVD11	LCD databit 11
GPIO46	P0.6 / LCDVD12	LCD databit 12 Can optionally be connected to USB1 VBUS feedback.
GPIO47	P0.7 / LCDVD13	LCD databit 13
GPIO48	GPO_22 / LCDVD14	LCD databit 14
GPIO49	SYSCLKEN / LCDVD15	LCD databit 15
GPIO50	SPI2_DATIO / LCDVD20	LCD databit 20
GPIO51	SPI2_DATIN / LCDVD21	LCD databit 21. Also connected to USB OTG transceiver, interrupt signal. Can optionally be connected to USB1 VBUS power switch over-current flag (not used on the LPC3250).
GPIO52	GPIO_04 / LCDVD22	LCD databit 22. Can also be connected to USB OTG transceiver(not used on the LPC3250)
GPIO53	SPI2_CLK / LCDVD23	LCD databit 23. Can also be connected to USB OTG transceiver (not used on the LPC3250)
GPIO54	USB_CONN_VBUS	Connects to USB2 VBUS signal.
GPIO55	GPI_02	Connects to USB2 VBUS power switch over-current flag.
GPIO56	I2S1TX_CLK	Can be connected to I2S audio codec BCKI (I2STX-BCK)
GPIO57	I2S1TX_SDA	Can be connected to I2S audio codec BCKI (I2STX-DATA)
GPIO58	I2STX_WS	Can be connected to I2S audio codec BCKI (I2STX-WS)
GPIO59	P0.0 / I2SRX_CLK	Can be connected to I2S audio codec BCKI (I2SRX-BCK)

GPIO60	P0.1 / I2SRX_WS	Can be connected to I2S audio codec BCKI (I2SRX-WS)
GPIO61	GPI_00 / I2SRX_SDA	Can be connected to I2S audio codec BCKI (I2SRX-DATA)
GPIO62	I2C2_SDA	No special usage on *OEM Base Board*.
GPIO63	I2C2_SCL	No special usage on *OEM Base Board*.
GPIO64	GPI_04	No special usage on *OEM Base Board*.
GPIO65	GPI_06	No special usage on *OEM Base Board*.
GPIO66	USB_CONN_ID	Connects to ID-pin of mini-AB USB connector on USB#2 interface.
GPIO67	POWER_ON	No special usage on *OEM Base Board*.
GPIO68	TST_CLK2	Can be connected to I2S audio codec SYSCLK.
GPIO69	P2.7	Can be used as enable signal for IrDA transceiver (active low).
GPIO70	GPIO_00	Connects to LCD expansion connector (interrupt signal from touch screen controller).
GPIO71	GPIO_01	Can be connected to Ethernet PHY interrupt output
GPIO72	GPI_07	Connects to serial expansion connector.
GPIO73	P2.0	Connects to joystick switch
GPIO74	P2.1	Connects to joystick switch
GPIO75	P2.2	Connects to joystick switch
GPIO76	P2.3	Connects to joystick switch
GPIO77	P2.4	Connects to joystick switch
GPIO78	P2.5	No special usage on *OEM Base Board*
GPIO79	P2.6	No special usage on *OEM Base Board*
GPIO80	GPO_07 / LCDVD2	LCD databit 2
GPIO81	GPO_21 / LCDVD3	LCD databit 3
GPIO82	GPI_08	No special usage on QVGA Base Board
GPIO83	GPO_23 / U3_RTS	Can be connected to RS232 interface
GPIO84	GPI_09	No special usage on QVGA Base Board
GPIO85	GPI_19 / U4_RX	No special usage on QVGA Base Board
GPIO86	P2.8	No special usage on QVGA Base Board
GPIO87	P2.9	No special usage on QVGA Base Board
GPIO88	P2.10	No special usage on QVGA Base Board
GPIO89	P2.11	No special usage on QVGA Base Board
GPIO90	P2.12	No special usage on QVGA Base Board
GPIO91	GPI_28 / U3_RI	Can be connected to RS232 interface
GPIO92	U3_DTR / U2_TX	Can be connected to RS232 interface

GPIO93	U3_DSR / U2_RX	Can be connected to RS232 interface
GPIO94	GPI_05 / U3_DCD	Can be connected to RS232 interface
GPIO95	U3_CTS / U2_CTS	Can be connected to RS232 interface
GPIO96	U3_RX	Can be connected to RS232 interface
GPIO97	U3_TX	Can be connected to RS232 interface
BD0-BD15	EMC_D0-EMC_D15	Connects to parallel NOR flash and 16-bit register.
BA0-BA23	EMC_A0-EMC_A23	The address bus, buffered signals from OEM Board. Connects to parallel NOR flash and 16-bit register.
BOE	EMC_OE	OE, buffered signal from cpu board . Connects to parallel NOR flash.
BWE	EMC_WE	WE, buffered signal from cpu board. Connects to parallel NOR flash and 16-bit register.
BBLS0	EMC_BLS0	BLS0 controls lower 8 bits of external memory databus (16-bit parallel register). Connects to 16-bit register.
BBLS1	EMC_BLS1	BLS1 controls upper 8 bits of external memory databus (16-bit parallel register). Connects to 16-bit register.
BCS0	EMC_CS0	No special usage on *OEM Base Board*.
BCS1	EMC_CS1	No special usage on *OEM Base Board*.
BCS2	EMC_CS2	CS2, chip select for parallel NOR flash.
BCS3	EMC_CS3	CS3, chip select for 16-bit regsiter.
VBAT	VBAT	Super-capacitor on vbat signal
ALARM	ONSW	Connected to alarm-LED (active high)
WP_NAND	GPO_19	Connected to NAND FLASH WriteProtect input internally on LPC3250 OEM board
VREF	Not connected	
VDDA, VSSA	VDDA, VSSA	Used to generate reference voltage for trimpot (analog inputs)
JTAG signals	JTAG signals	Connected to JTAG connectors
JTAG_DBGEN	DBGEN	
RESET	RESET_IN	Connects to RESET push-button and USB-to-serial bridge (for automatic service mode functionality)
RSTOUT	RESOUT	Connects to RESET LED indicator. Used to reset PCA9532, UDA1380, FT232RL, MIC2555, RF module and LCD expansion connector.

5 Getting Started

This chapter contains information about how to get acquainted with the *LPC3250 Developer's Kit*. ***Please read this section first before you start using the board - it will be well spent time!***

5.1 Powering

The board can normally be powered from a PC/laptop via the included USB cable (mini-B to A cable). This is done via the USB-to-UART connector and will work in most cases. Note however that not all PC/laptops can provide the needed current. An external USB hub can be used in this case or an external power supply.

A separate power supply is always needed in stand-alone situations or when running USB Host application (when powering external USB devices). If USB Host is not used (or used with very low external current consumption), a 5VDC/1 Amp power supply (5W) is all what is needed. If USB Host is used with high external current consumption or a display with high current consumption, a 5VDC/2-2.5 Amp power supply (10-12.5W) is needed. A 2.1mm standard power plug is used to interface the board. Center pin is positive.

See *Figure 26* below for locating the USB connector and/or the 2.1mm power connector.

It is possible to have both the USB cable and external powering connected to the board at the same time.

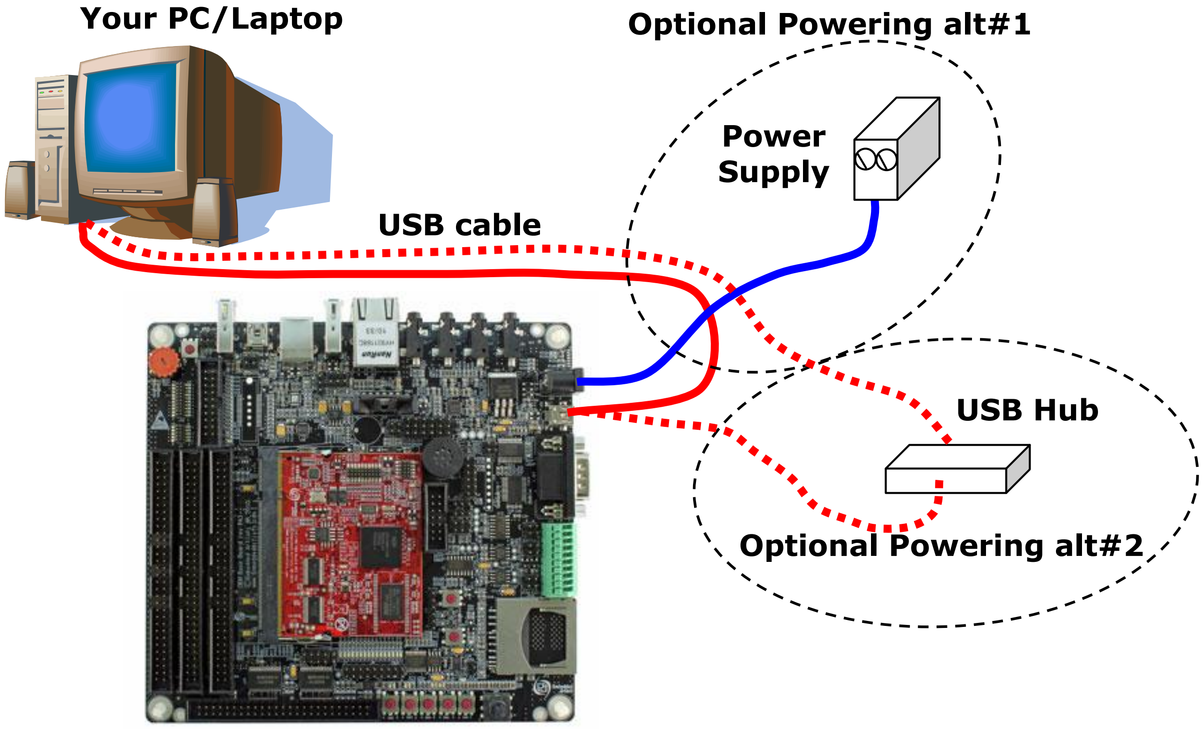


Figure 26 – Powering of OEM Base Board

5.2 Installing USB Driver

The *OEM Base Board* contains an USB-to-UART bridge chip (FT232R from FTDI) that connects UART channel #5 on the LPC3250 to a virtual COM port on the PC/laptop (via USB). This UART channel is typically used as the console channel for applications. Printf() output can for example be directed to this UART channel.

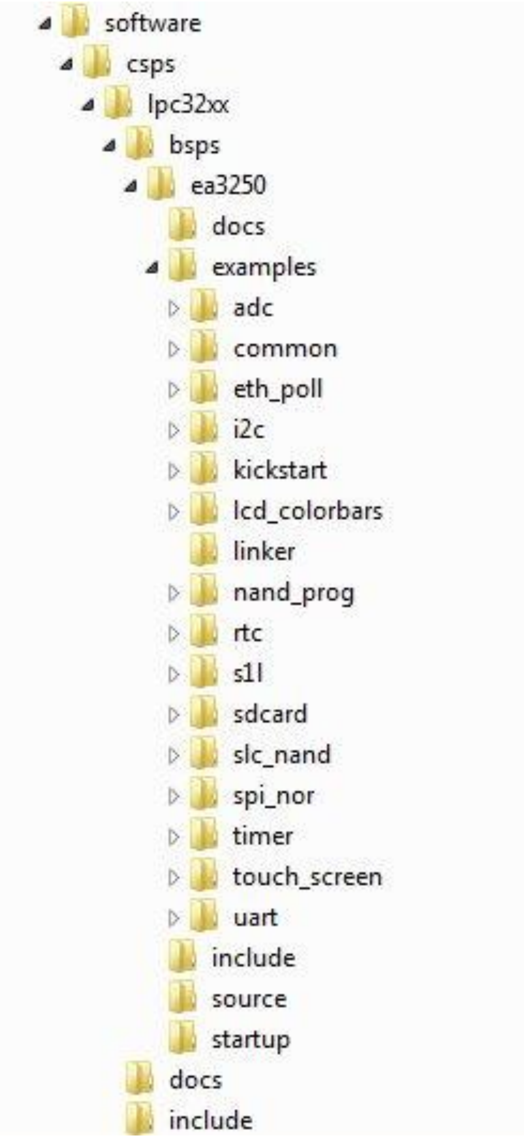
A USB driver must be installed on the PC/laptop in order for the virtual COM port to be created. See FTDI's installation guides for details how to install the driver for different operating systems:

http://www.ftdichip.com/Support/Documents/InstallGuides.htm

5.3 Building Sample Applications

A bundle with sample applications is available at Embedded Artists' support site. These applications illustrate the use of different peripherals on both the LPC3250 OEM Board and the OEM Base Board. The sample applications have been developed using the Common Driver Library (CDL) framework developed by NXP.

The CDL framework contains start-up code, initialization of clocks and drivers for the peripherals on the microcontroller. The file structure, see figure below, is organized in a way where chip specific code is separated from board specific code and the structure also supports using different build environment such as CodeSourcery's GNU tools or Keil's uVision.



```
software
  csps
    lpc32xx
      bsps
        ea3250
          docs
          examples
            adc
            common
            eth_poll
            i2c
            kickstart
            lcd_colorbars
            linker
            nand_prog
            rtc
            s1l
            sdcard
            slc_nand
            spi_nor
            timer
            touch_screen
            uart
          include
          source
          startup
      docs
      include
```

Figure 27 – Common Driver Library Structure

5.3.1 Compile Using CodeSourcery

This section describes how you compile an application using the CodeSourcery GNU tools. Please **note** that there is no guarantee that all applications provided by Embedded Artists will be able to compile using CodeSourcery GNU since the main tool is Keil's uVision.

1. Download the CodeSourcey Tools from www.codesourcery.com. The tool is called Sourcery CodeBench and there is a *Lite Edition* available for free after a registration. Select the version for ARM processors and with the EABI target OS.
2. Download the sample applications from Embedded Artists support site and unzip the file.
3. Start a command prompt (Start button → run → cmd)
4. Change to the software directory in the recently unzipped file.
5. Run setenv ea3250 gnu.
6. Change directory to the sample application you would like to compile, for example, software/csps/lpc32xx/bsps/ea3250/examples/timer
7. Run make to compile the sample application.
8. A .bin file as well as a .srec file will be generated that can be used for booting.

5.3.2 Compile Using Keil's uVision

This section describes how you can compile an application using Keil's uVision.

1. You must have Keil's uVision installed on your computer. An evaluation version can be downloaded from www.keil.com.
2. Download the sample applications from Embedded Artists support site and unzip the file.
3. The sample applications contain Keil project files. Double click on the file with extension uvproj and the Keil uVision tool will be started.
4. Press the F7 key or go to the *Project* menu and select *Build Target*.
5. If you have a USB-JTAG adapter connected between the *LPC3250 OEM Board* and your computer you can directly download the application to internal RAM of the LPC3250. You do this by pressing CTRL+F5 keys or by going to the Debug menu and selecting the Start/Stop Debug Session alternative.
6. If you don't have a USB-JTAG adapter you can program the board with the generated bin file.

5.4 Booting

The processor will start its code execution from an internal ROM, containing the boot code. This boot loader can load code from one of four external sources to internal RAM (IRAM) at address 0x0.

- UART5
- SSP0 (in SPI mode)
- EMC Static CS0 memory
- NAND FLASH

The boot loader loads code as a single contiguous block at a maximum size of **56 kByte**. Programs larger than this size must be loaded in more steps, for example, by a secondary boot loader.

The LPC3250 User's Manual from NXP (chapter 35) contains all details about booting. The rest of this section gives an overview of what is supported by the *LPC3250 OEM Board* and *OEM Base Board*, and how to quickly get started with the download process.

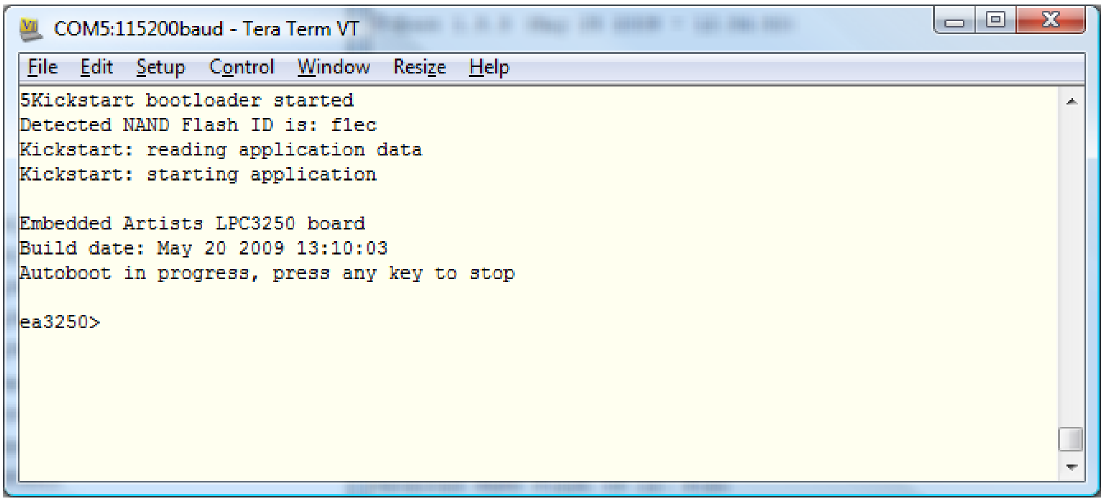
5.4.1 Kickstart Loader

By default the Embedded Artists *LPC3250 OEM Board* is programmed with the kickstart loader in block 0 of the NAND flash. The responsibility of this loader is to load an application stored in block 1 and onwards of the NAND flash. The kickstart loader will load the application into internal RAM (IRAM) at address 0x0.

The kickstart loader is available in the bundle with sample applications provided by Embedded Artists (examples/kickstart). The application nand_prog (examples/nand_prog) is used to program the NAND flash with the kickstart loader as well as a second application, for example, the stage 1 loader.

5.4.2 Stage 1 Loader

By default the Embedded Artists LPC3250 OEM Board is programmed with the stage 1 loader (s1l) in block 1 of the NAND flash. This application will be loaded by the kickstart loader when the *LPC3250 OEM Board* powers up, see figure below.


COM5:115200baud - Tera Term VT
File Edit Setup Control Window Resize Help

```
5Kickstart bootloader started
Detected NAND Flash ID is: flec
Kickstart: reading application data
Kickstart: starting application

Embedded Artists LPC3250 board
Build date: May 20 2009 13:10:03
Autoboot in progress, press any key to stop

ea3250>
```

Figure 28 – S1L loader up and running

The S1L loader will initialize the board, such as clocks and external memory and then start a console where you can give input commands to the loader. S1L offers the following booting options.

- MMC/SD card
- UART5
- NAND Flash

Load application from MMC/SD card

First make sure you have stored the srec file on the FAT formatted MMC/SD card. Please note that the S1L loader doesn't support long file names.

1. Load srec file to memory

```
ea3250> load blk timer~1.sre srec
```

2. Start the application using the exec command

```
ea3250> exec
```

If the application has been built and linked to execute from external memory you could also use the binary file when loading the application. This is, for example, the case for the u-boot bootloader.

1. Load the binary file with type set to raw.

```
ea3250> load blk u-boot.bin raw 0x83fc0000
```

2. Start the application using the exec command and the load address.

```
ea3250> exec 0x83fc0000
```

Instead of directly executing the binary file you could store it in NAND flash by issuing the nsave command after it has been loaded to the external memory.

Load application from UART

Loading from UART is similar to loading from MMC/SD card with the difference that source is set to term.

1. Start the download process.

```
ea3250> load term srec
```

2. Start to transfer the srec file using your terminal application
3. When the application has been transferred, the prompt will return and you can execute the application

```
ea3250> exec
```

More commands are available in the S1L loader and you can use the help command to find out more about how to use them.

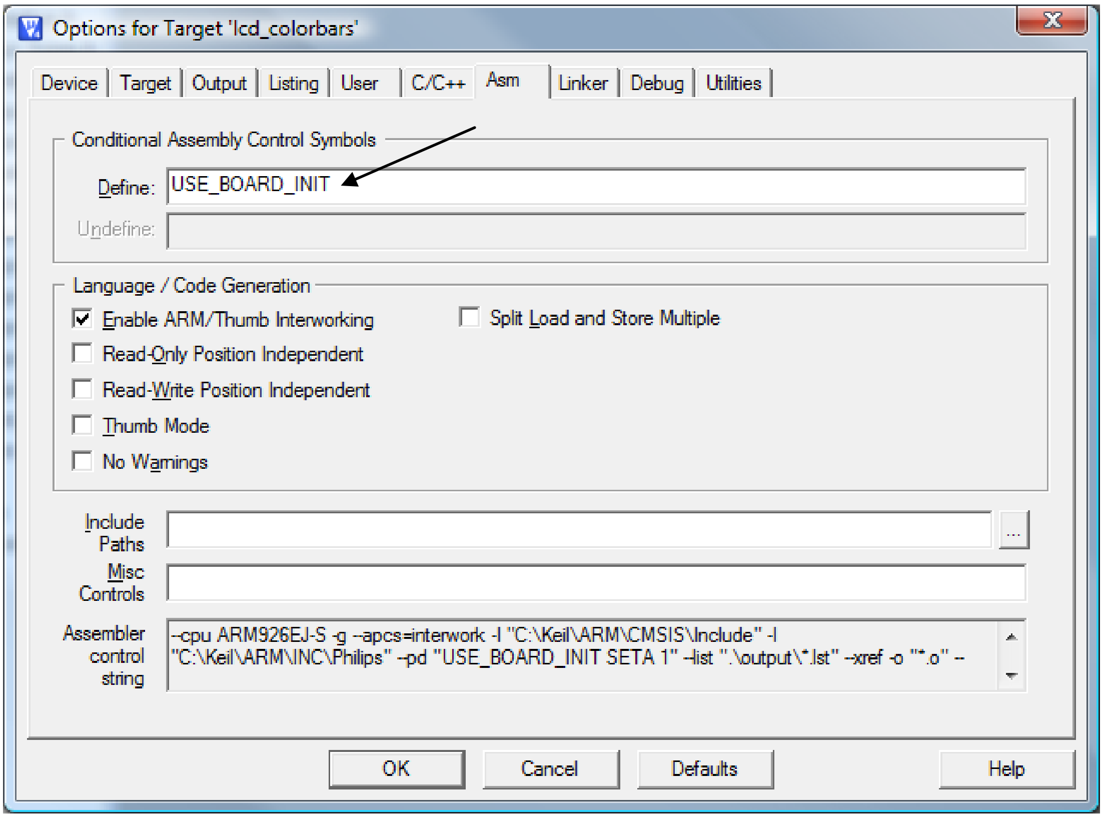
Store Application in NAND Flash and Execute From External Memory

Note: These instructions are only valid for the sample applications based on the CDL2 package.

By default the sample applications found in the zip file on Embedded Artists support site have been setup to run from internal memory which is especially suitable when testing and developing an application. The application is downloaded directly to the internal memory through, for example, a JTAG adapter.

When an application should be stored persistently on the LPC3250 OEM Board the NAND flash is a suitable place to store it since the LPC3250 microcontroller doesn't have any on-chip flash. It can also be suitable to run the application from external memory especially if it is too large to fit into internal memory. These instructions show how to build the applications (using Keil uVision) so that they can be run from external memory and also how to store that application in NAND flash.

1. Open the project in Keil uVision
2. Make sure board initialization isn't executed when the application is built for external memory. The S1L boot loader will take care of initialization and it shouldn't be executed twice. Go to Project → Options → Asm. **Remove** the USE_BOARD_INIT define.


Options for Target 'lcd_colorbars'
Device
Target
Output
Listing
User
C/C++
Asm
Linker
Debug
Utilities
Conditional Assembly Control Symbols
Language / Code Generation
Include Paths
...
OK
Cancel
Defaults
Help

Define: USE_BOARD_INIT

Undefine:

☑ Enable ARM/Thumb Interworking

□ Read-Only Position Independent

□ Read-Write Position Independent

□ Thumb Mode

□ No Warnings

□ Split Load and Store Multiple

Misc Controls:

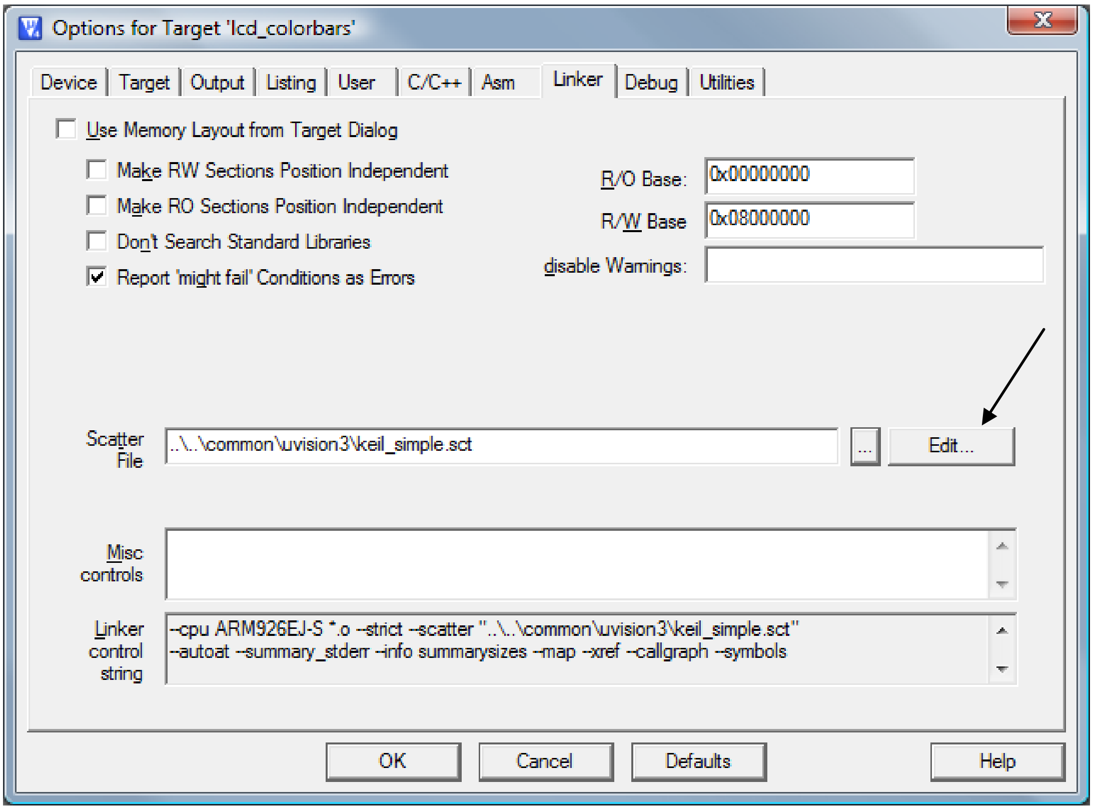
Assembler control string:

```
--cpu ARM926EJ-S -g --apcs=interwork -I "C:\Keil\ARM\CMSIS\Include" -I
"C:\Keil\ARM\INC\Philips" --pd "USE_BOARD_INIT SETA 1" --list ".\output\*.lst" --xref -o "*.o" --
```

Figure 29 – Asm settings in Keil uVision

3. Change linker script so that the application runs from external memory instead of internal memory. Go to Project → Options → Linker. Click Edit button to the right on the Scatter File. The scatter file will now be opened. **Please note** that this file is common for all sample applications. If only one application should be changed copy this file to the local project instead of changing the common file.

   Change the LR_RAM1 row. The first argument is the execute address which could be changed to 0x80000000 if the application should be run from the beginning of the external memory. The second argument is the size of the LR_RAM1 region. In the example below this is set to 16 MB (0x01000000).



Options for Target 'lcd_colorbars'

□ Use Memory Layout from Target Dialog

□ Make RW Sections Position Independent

□ Make RO Sections Position Independent

□ Don't Search Standard Libraries

☑ Report 'might fail' Conditions as Errors

R/O Base: 0x00000000

R/W Base 0x08000000

disable Warnings:

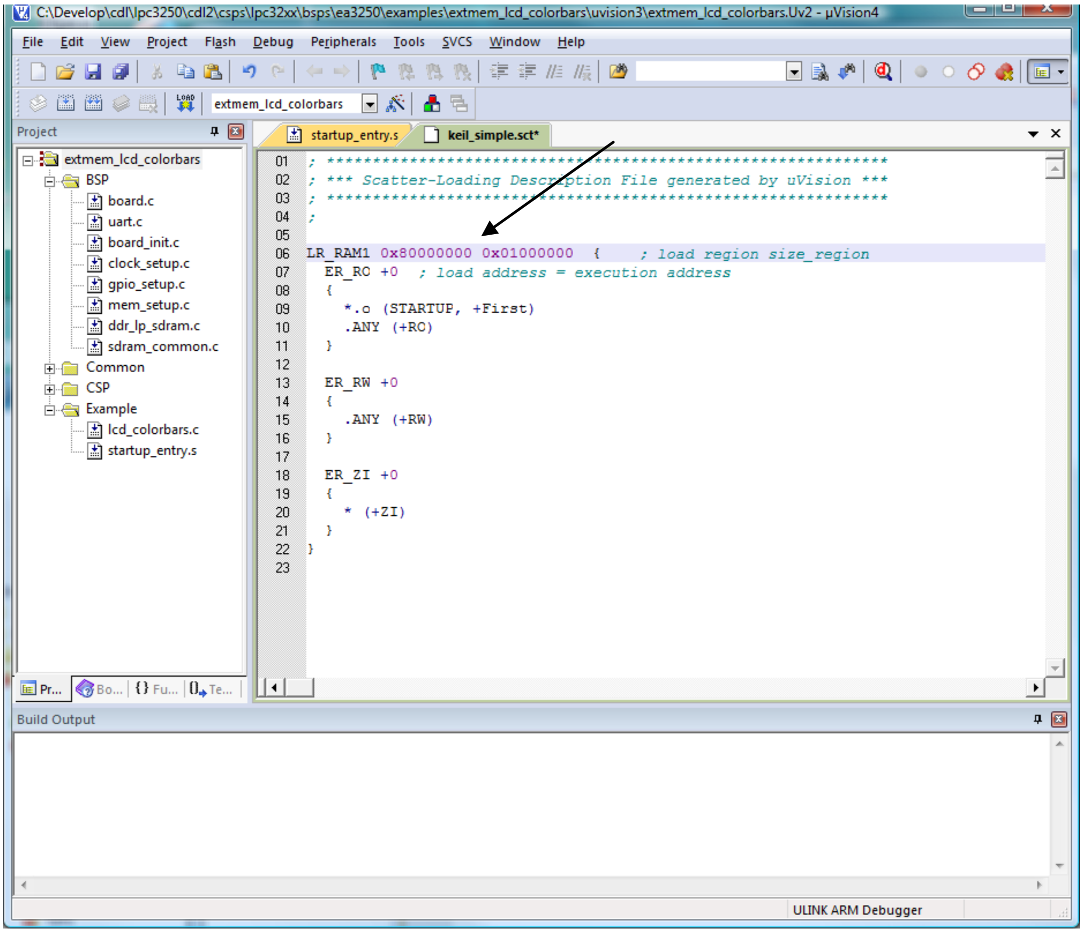
Scatter File: ..\..\common\uvision3\keil_simple.sct

Misc controls:

Linker control string:

```
--cpu ARM926EJ-S *.o --strict --scatter "..\..\common\uvision3\keil_simple.sct"
--autoat --summary_stderr --info summarysizes --map --xref --callgraph --symbols
```

Figure 30 – Linker settings in Keil uVision



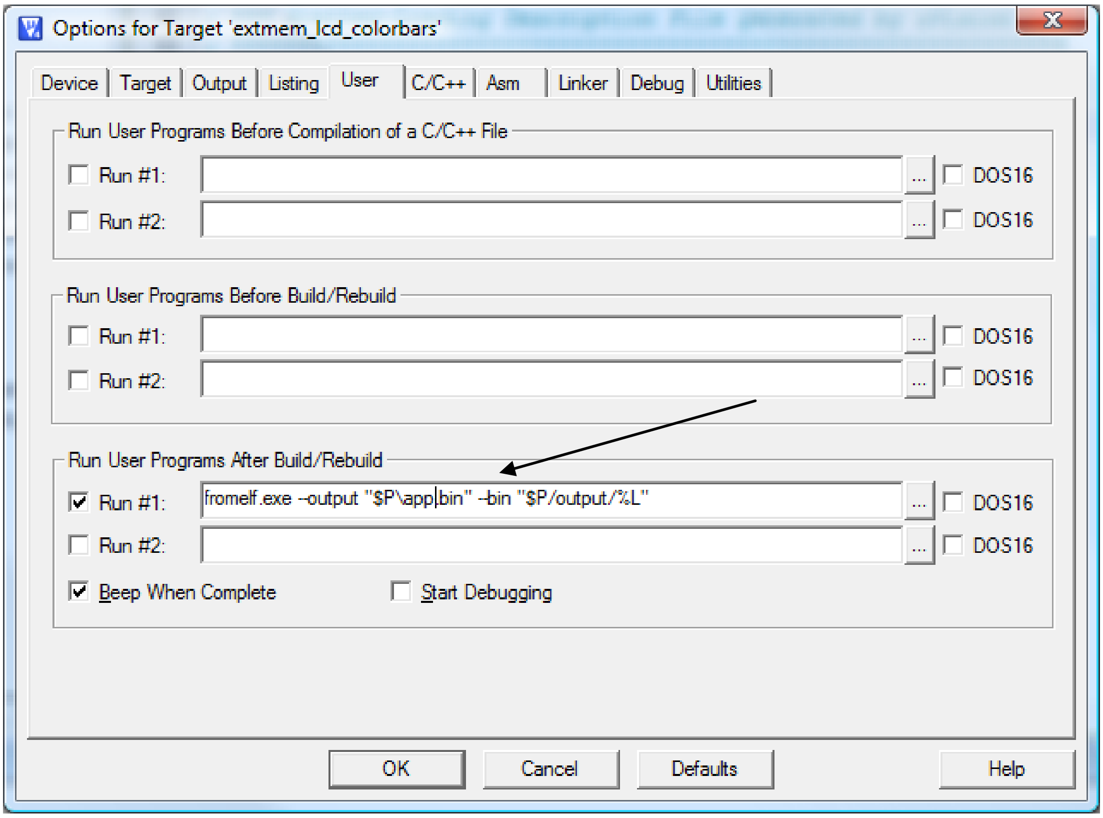
C:\Develop\cdl\lpc3250\cdl2\csps\lpc32xx\bsps\ea3250\examples\extmem_lcd_colorbars\uvision3\extmem_lcd_colorbars.Uv2 - µVision4

```
; *************************************************************
; *** Scatter-Loading Description File generated by uVision ***
; *************************************************************
;

LR_RAM1 0x80000000 0x01000000  {    ; load region size_region
  ER_RO +0  ; load address = execution address
  {
    *.o (STARTUP, +First)
    .ANY (+RO)
  }

  ER_RW +0
  {
    .ANY (+RW)
  }

  ER_ZI +0
  {
    * (+ZI)
  }
}

```

Figure 31 – Scatter file for the sample applications

4. Create a binary file to be stored in NAND flash. This can be done by using the fromelf.exe utility. Go to Project → Options → User. Add the command below to the "Run User Programs After Build/Rebuild"

```
fromelf.exe --output "$P\app.bin" --bin "$P/output/%L"
```



Options for Target 'extmem_lcd_colorbars'

Run User Programs Before Compilation of a C/C++ File

□ Run #1: □ DOS16

□ Run #2: □ DOS16

Run User Programs Before Build/Rebuild

□ Run #1: □ DOS16

□ Run #2: □ DOS16

Run User Programs After Build/Rebuild

☑ Run #1: fromelf.exe --output "$P\app.bin" --bin "$P/output/%L" □ DOS16

□ Run #2: □ DOS16

☑ Beep When Complete □ Start Debugging

5. Put the binary file (app.bin) on a memory card.

6. Boot into the S1L boot loader and load the application from the memory card, save it to NAND flash and set S1L to automatically load the application at boot.

```
ea3250> load blk app.bin raw 0x80000000
ea3250> nsave
ea3250> aboot flash raw 0x80000000
ea3250> prompt ea3250> 2
```

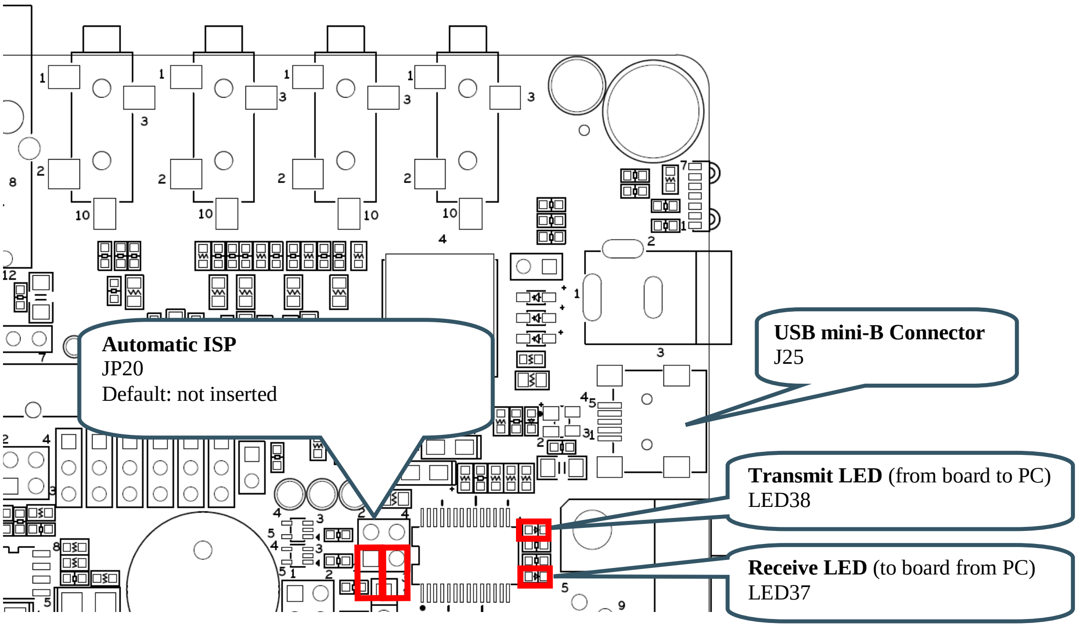
5.4.3 Service Boot

The LPC3250 bootloader starts by reading the SERVICE_N (GPI_01) input when deciding from which source to boot. If SERVICE_N is low a service boot is performed, which means that it will load code from UART5.

On the *OEM Base Board*, GPI_01 is connected to the SW6 button. Holding down the SW6 button while resetting the board will force it into service boot.

There are two jumpers on the *OEM Base Board* related to the USB-to-UART serial channel (UART #5) control signals and automatic service mode functionality. See Figure 32 for details about where the jumpers are located. Make sure the two jumpers in JP20 are not inserted, i.e., open.

Other OEM boards can enable this mode automatically via the UART modem control signals, but this is not possible for the LPC3250. Therefore JP20 jumpers shall be open.



Automatic ISP
JP20
Default: not inserted

USB mini-B Connector
J25

Transmit LED (from board to PC)
LED38

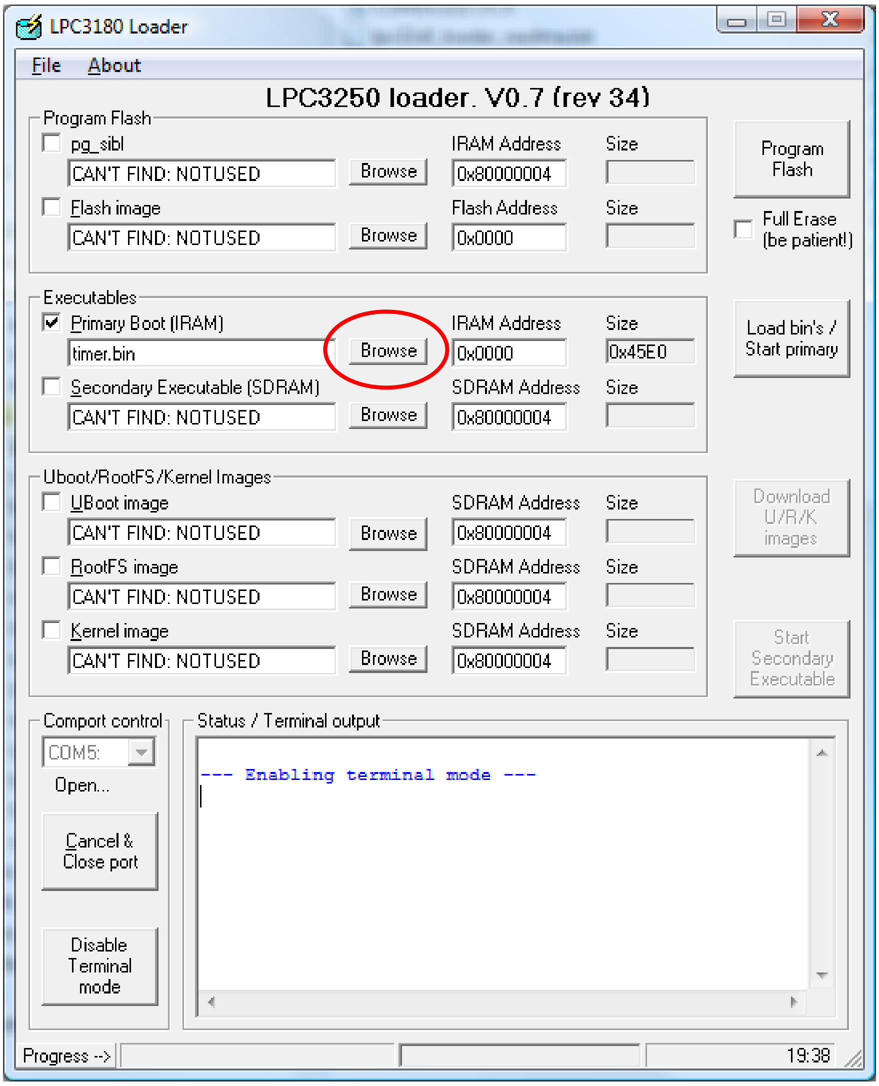
Receive LED (to board from PC)
LED37

Figure 32 – UART-to-USB Bridge

If JP20 jumpers are inserted it is possible that a terminal program resets the board and/or enable service boot mode by accident.

In the software bundle with sample applications there is also a Windows application which can be used to transfer a file to the board while it is in service boot mode. The application is available in the `software/cspc/lpc2xx/tools/LPC3250 loader` directory.

1. Start the LPC3250 loader application, see Figure 33.
2. Click the "Browse" button in the Executables - Primary Boot section and select the binary file you would like to transfer to the board.
3. Select COM port and then click the "Enable Terminal Mode" button.
4. Click the "Load bin's / start primary" button.
5. Hold down the SW6 button on the *OEM Base Board* and then press the Reset button on the *OEM Base Board*. When you release the reset button the download will begin.



LPC3180 Loader

LPC3250 loader. V0.7 (rev 34)

Program Flash

□ pg_sibl: CAN'T FIND: NOTUSED — IRAM Address: 0x80000004 — Size:

□ Flash image: CAN'T FIND: NOTUSED — Flash Address: 0x0000 — Size:

□ Full Erase (be patient!)

Executables

☑ Primary Boot (IRAM): timer.bin — IRAM Address: 0x0000 — Size: 0x45E0

□ Secondary Executable (SDRAM): CAN'T FIND: NOTUSED — SDRAM Address: 0x80000004 — Size:

Uboot/RootFS/Kernel Images

□ UBoot image: CAN'T FIND: NOTUSED — SDRAM Address: 0x80000004 — Size:

□ RootFS image: CAN'T FIND: NOTUSED — SDRAM Address: 0x80000004 — Size:

□ Kernel image: CAN'T FIND: NOTUSED — SDRAM Address: 0x80000004 — Size:

Comport control: COM5: — Open...

Status / Terminal output

```
--- Enabling terminal mode ---
```

Progress --> 19:38

Figure 33 – LPC3250 loader application

5.5 Handling SO-DIMM Boards

See picture below for instructions about how to mount/remove the *LPC3250 OEM Board*.

To install the *OEM Board*, align it to the socket (1). Push the board gently, and with even force between the board edges, fully into the socket (2). Then push the board down in a rotating move (3) until it snaps into place (4). The *OEM Board* shall lie flat and parallel to the base board.

To remove the *OEM Board*, spread the two arms of the SO-DIMM socket apart slightly. The board will pop up (5). Gently rise the board in a rotating move (6) and then extract the board from the socket (7). Apply even force between board edges when removing so that the board is removed parallel to the locking arms.

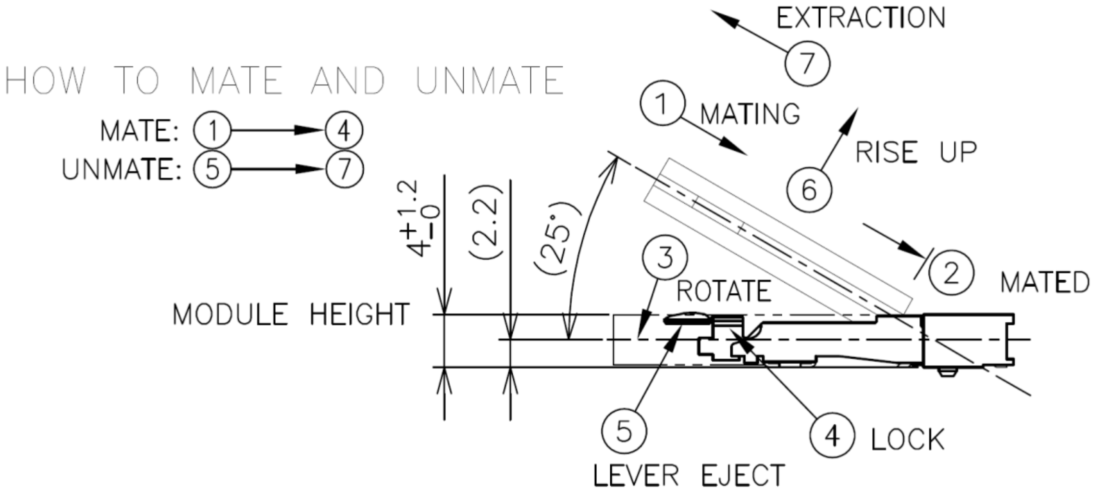


Figure 34 – Instructions how to Mount/Remove the LPC4088 OEM Board

Do not forget to follow standard ESD precaution routines when mounting/removing the *OEM Board*. Most signals exposed on the 200 edge contact fingers on the SO-DIMM board are unprotected. Maintain the same electrical potential of the *OEM Board* (to be mounted) and the base board. Do not touch the *OEM Board* edge connectors. Handle the *OEM Board* only by the three other edges. Also, do not touch the components on the board.

5.6 Things to Note

This section lists things to note when using the *LPC3250 OEM Board* and the *OEM Base Board*.

5.6.1 LPC3250 Adjustable Core Voltage

The core voltage for the LPC3250 is adjustable via I2C commands sent to the LTC3447 voltage converter. The core voltage should always be increased to 1.35 volt in order for external mobile DDR SDRAM to function properly. Initialization code that increase the adjustable core voltage to 1.35V can be downloaded from Embedded Artists support pages.

Note that it is the user's responsibility not to program the LTC3447 to generate too high core voltage, which is possible. The LTC3447 can generate voltages up to 2V, which by far exceed the limits for the LPC3250.

5.6.2 NAND FLASH Bad Block

The NAND Flash is the K9F1G08 from Samsung and has 1 GBit capacity. The chip may include invalid blocks when shipped from factory. A maximum of 20 invalid blocks may exist initially, i.e., 1004-1024 valid blocks. Additional invalid blocks may develop while being used. Invalid blocks are defined as blocks that contain one or more bad bits. Do not erase or program factory-marked bad blocks. More information about appropriate management of invalid blocks can be found in technical notes and datasheet from Samsung.

5.6.3 Brand of Memory Chips

Note that there is no guarantee for a certain brand or version of memory chips; SPI-NOR flash, parallel NAND flash and mobile DDR SDRAM. The lifetime of memory chips is limited and availability can also be limited from time to time. Embedded Artists make every effort to mount the original design chip on the board. In case that is impossible a compatible chip will instead be mounted without any prior notice. There can be small programming differences between mounted brands. The application program shall always read the chip id of flash devices to make certain which chip is actually mounted on the board.

The support page contains datasheets to the different memory devices and information about mounted devices on different board versions.

5.6.4 LPC3250 Peripherals

The *key scan interface* peripheral cannot be used with the *LPC3250 OEM Board* because the Ethernet interface is active.

5.6.5 Current Consumption and Limits of USB Ports

The current consumption of *LPC3250 OEM Board*, *OEM Base Board*, and the *4.3 inch LCD Board* is in the region of 90-150mA. This is below what a normal USB-A port can supply. However, if the boards are powered from a USB-A port (for example a PC or laptop) and there are problems, like spontaneous resets or other strange things, it is likely that an external power supply is needed. Powering via a USB hub is also a possibility.

When using the *7 inch LCD Board* an external power supply is always needed. A normal USB-A port cannot supply the needed current when the 7 inch LCD is used.

5.6.6 LCD flickering

When using the LCD expansion interface, R163 must be removed on the *OEM Base Board*. If not, the LCD can start flickering and the blue component of the pixel color can disappear for short time periods (= the flickering). If the picture is mainly blue it will look like a black row on the display. If other colors are dominating on the picture shown, it can also look like yellow lines (white minus block color component result in a yellow color).

The reason for this problem is that the I2C interface of the MIC2555 USB OTG transceiver is connected to the two MSB signals of the blue color component. Even though the pixel data is very much out of spec for the I2C interface, the MIC2555 reacts on some sequences and start driving the MSB signal low, which is the flickering that is seen. The solution is to remove resistor R163.

The LCD interface and the USB OTG transceiver cannot be used at the same time on the *OEM Base Board*.

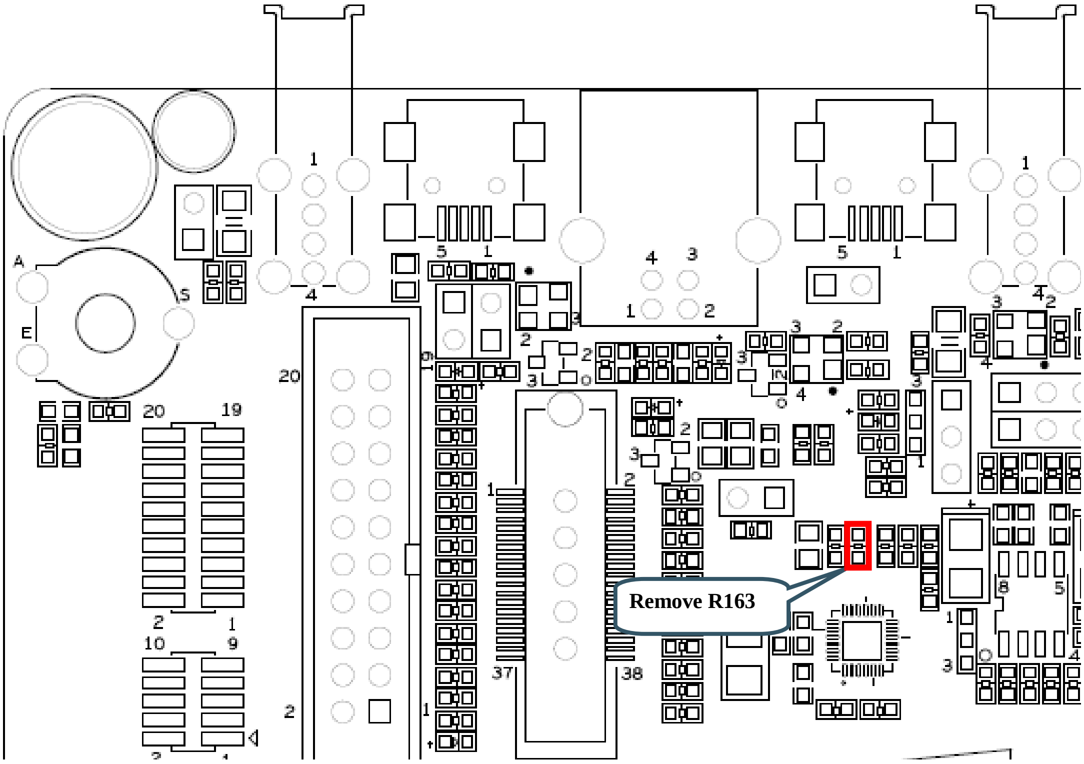


Figure 35 – Identifying R163

Note that R163 has been removed on all boards shipped out from Embedded Artists after May 15, 2012. Since there is a transition period, and stock at distributors, there is no guarantee that the resistor has been removed on boards delivered after this date. Please always check if the board has R163 removed. If not, remove R163.

5.6.7 Initialization of External Memory Bus

The databus buffer (U15) to the external memory bus is controlled by the OE and BLS0/BLS1 signals. Even though the external memory bus is not used, it is recommended to initialize the EMC peripheral and associated pins fully (for both SDRAM and static memory regions). This will give OE/BLS0/BLS1 defined states.

5.6.8 USB OTG Transceiver

The USB OTG transceiver (U31, MIC2555) exists in two versions, -0YML and -1YML. Either one can be mounted on the *OEM Base Board*. The difference is the I2C address that the chip answers to. Any software using the MIC2555 must check which I2C address the chip responds to.

The OTG transceiver is connected to the USB channel that is not used by the LPC3250, so this is no problem for LPC3250 OEM board users.

5.6.9 OEM Base Board revision

The LPC3250 is fully supported by the OEM Base board revision B. Earlier revisions are rev A and PB1. The main difference between these earlier board revisions is jumper JP34 (LCD pixel data mapping). This jumper was added on revision B.

6 LCD Expansion Connector

Embedded Artists supplies many different display options as add-on boards. These display boards are great to use for getting started quickly and test a specific display resolution in an application. All LCD boards supplied from Embedded Artists use the LCD Expansion Connector. There is also an option to connect custom displays via the LCD Expansion Connector, J26.

The *OEM Base Board* contains a buffered LCD expansion interface via a 50 pos IDC connector, J26. The expansion interface also includes UART (see UART multiplexing), I2C and SPI interfaces. These additional interfaces are for identifying external displays (via configuration I2C-E2PROM) as well as touch screen controllers. An external LCD pixel clock can also be supplied via the expansion connector.

The LCD expansion connector carries 18 data bits per pixel by default (6 per RGB color). The LPC3250 LCD controller can produce 24 data bits per pixel and it is possible to output all these signals on the LCD expansion connector. The trade-off is that the UART and I2C serial interfaces have to be removed. Via SJ6-SJ11 it is possible to select what signals to make available on the LCD expansion connector. By default pad 1-2 are connected on SJ6-SJ11.

For performance reasons (on the LPC3250), a system with 16-bit color information per pixel is typically what is implemented. A 565-system is most commonly used. This means 5 bits for red, 6 for green and 5 for blue - 16 bits in total. 24-bit systems will double to load on the external memory bus since each pixel is stored as 32-bits (instead of 16-bits, or fewer).

Note that the LCD Expansion Connector only supports connecting to a display via the RGB interface. Some lower resolution displays have the option to connect via a memory bus-like interface, typically 8 or 16-bit. If a memory bus-like interface shall be used, use the external memory bus available via expansion connector J4 instead.

Also note that JP34 shall have a jumper in position 2-3 when working with the LPC3250.

There are many steps to consider when connecting a custom LCD to the LCD Expansion Connector.

- As a first step, the pixel color signals must be matched between the LPC3250 and display. The LPC3250 has many options for routing the pixel data to different pins and there are differences depending on the display mode selected. The *OEM Base Board* supports the default settings when using TFT 4:4:4 or TFT 5:6:5-mode. For 24-bit systems, the extra serial interfaces on the LCD Expansion Connector must be traded-off.
    - Many displays have 24 or 18 bit color interfaces, i.e., 8 or 6 bits per color. This means that some color information is missing when using TFT 4:4:4 or TT5:6:5-mode. Always connect the available color bits from the LPC3250 to the MSB bits. A common solution is to connect the most significant color bit to the missing (LSB) bits. Alternatively, just ground the LSB bits.
    - The table below lists where the pixel color information can be found when working in TFT 4:4:4 or TFT 5:6:6 mode.

LCD-VD bit in LCD Expansion Connector, J26	***J26 pin***	***TFT 4:4:4 mode***	***TFT 5:6:5 mode***
LCD_VD4	7	RED0 (LSB)	RED1
LCD_VD5	8	RED1	RED2
LCD_VD6	9	RED2	RED3
LCD_VD7	10	RED3 (MSB)	RED4 (MSB)

LCD_VD10	15		GREEN0 (LSB)
LCD_VD11	16		GREEN1
LCD_VD12	17	GREEN0 (LSB)	GREEN2
LCD_VD13	18	GREEN1	GREEN3
LCD_VD14	19	GREEN2	GREEN4
LCD_VD15	20	GREEN3 (MSB)	GREEN5 (MSB)
LCD_VD18	25		RED0 (LSB)
LCD_VD19	26		BLUE0 (LSB)
LCD_VD20	27	BLUE0 (LSB)	BLUE1
LCD_VD21	28	BLUE1	BLUE2
LCD_VD22	29	BLUE2	BLUE3
LCD_VD23	30	BLUE3 (MSB)	BLUE4 (MSB)

- The HSYNC, VSYNC, DEN control signals and the DOTCLK pixel clock signals must be connected to the display. All displays typically require the DOTCLK signal but there are variations on the control signals.
  - Some displays require all three control signals (HSYNC, VSYNC, DEN).
  - Some displays require only HSYNC and VSYNC.
  - Some displays require only DEN control signals.
  - Some displays require that HSYNC is delayed compared to VSYNC, i.e., VSYNC must have a falling edge before HSYNC (assuming that HSYNC/VSYNC are active negative) . If this is needed, HSYNC can be delayed one DOTCLK cycle with two D-type flip-flops.
- In most cases the LPC3250 can generate the appropriate DOTCLK frequency. The higher the frequency needed, the fewer available frequencies can be selected when dividing the core clock. If a specific frequency is needed, the LCDCLKIN signal can be used. It is an input to the LPC3250.
  - Most displays can accept a quite wide range of frequencies on the DOTCLK signal.
  - Embedded Artists has successfully used LCDCLKIN signals up to 36MHz. The upper limit is not known exactly. It is typically related to the core clock frequency.
If not used, leave this signal unconnected (i.e., LCDCLK is generated by the LPC3250).
- Some displays has a serial interface (typically SPI-like) for initialization of the controller chip inside the display. This is relatively common for smaller QVGA-sized displays but larger (resolution above QVGA – 320x240) displays typically do not require this initialization.
  - The LCD Expansion connector has an SPI interface that supports both 3-wire transfer and 4-wire transfers. A 3-wire SPI interface typically means that 9 data bits are transferred in every time, 8 data bits and one bit indicating if it is a command or data byte. SPI-CLK, SPI-MOSI and SPI-SSEL are needed, i.e., 3 signals.
For a 4-wire interface the transfers are 8 bits and a separate signal (the fourth wire) is used to signal if it is a command or data transfer. Signal SPI_LCD_DC is typically used for this.

Few displays allow for read back of internal control registers via SPI. Only write transfers are typically supported.

- Many displays have strict requirements of power sequencing during power up and down. Embedded Artists display boards has an I2C-GPIO expansion chip (PCA9532) and voltage control of +3.3V and +5V to allow power sequencing.
    - Embedded Artists recommend that the power sequences outlined in the display datasheets are followed for best long term performance of the display.
- For simplicity, Embedded Artists has added a configuration i2c-e2prom in the display add-on boards that contains information about the connected display. This simplified development of portable software.
    - A standard 24LC64 chip is used with i2c-address 0x56 (1.0.1.0.1.1.0.rw). A0 on the chip is connected to GND and A1/A2 to VCC to get address 0x56.
- Some displays also require a reset signal after VCC/VDD has stabilized. In that case the I2C-GPIO expansion chip can be used to create this signal. Alternatively a free GPIO signal in the LCD Expansion Connector can be used.
- All TFT LCDs must have a control for backlight. A constant current source is needed. Some displays have one LED string while others have 2-4 strings making control slightly more complicated.
    - OLEDs do not require separate backlight since the pixels transmit light themselves.
- As a final step a touch screen interface is typically needed.
    - Embedded Artists display boards often use the TSC2046 from TI. It has a simple SPI interface.
    - Capacitive touch screen controllers typically have I2C or SPI interfaces.
- Do not forget that some displays have configurations pins for selecting interface type functionality.

7 Troubleshooting

This chapter contains information about how to troubleshoot boards that does not seem to operate properly. It is strongly advised to read through the list of tests and actions that can be done before contacting Embedded Artists. The different tests can help determine if there is a problem with the board, or not. For return policy, please read Embedded Artists' General Terms and Conditions document (can be found at http://www.embeddedartists.com/shop/).

7.1 Powering

The first step is to make sure that powering works properly. The input power interface is described in section 4.23 .

1. Disconnect the powering and removed the *OEM Board*, i.e., remove it from the socket (J1). All other connections to the *OEM Base Board* shall also be removed.

2. Connect a stable DC supply (+5V, positive center) to J24. Minimum current capability shall be 1A for this test. The noise level should be minimal, less than 50mV.

3. Measure the +5V and +3.3V voltages with the help of the voltage measuring pads on the *OEM Base Board*.
   Acceptable range for the +5V voltage is 4.5V to 5.5V.
   Acceptable range for the +3.3V voltage is +3.2V to +3.4V.
   If the reset-LED is dimming instead of being clearly on or off, it is also a sign that the supply voltages are not stable and outside of valid ranges.
   If problem; since the *OEM board* is disconnected and there is normally no big current consumption on passive *OEM Base Board*, it is likely that some part of the *OEM Base Board* consumes a lot of current. It might be possible to locate the faulty section/component by checking component temperatures with the finger tip. Be careful because faulty components can get very hot.

4. Verify that there are no voltage dips on the supply. A typical indication of this is that the reset-LED flash from time to time. An oscilloscope can detect voltage dips; measure the supply voltage in AC mode. Set horizontal resolution to 50mV per square. Set the trigger to normal mode and adjust the trigger to 50mV below the normal level. That way, the oscilloscope will trigger every time a voltage dips occurs. By watching the dips it is possible to judge if there is a problem, or not.

5. Verify that the reset-LED flash when pressing the reset push-button. The LED should also light shortly after releasing the push button. The extra "LED on time" is very short but still noticeable (typically a 1/5 of a second, i.e., 200mS).

6. Disconnect the powering and mount the *OEM Board* again (in J1).
   Now, measure the +5V and +3.3V voltages again (see step 2 above). If the +5V voltage is below acceptable range it is a sign of too weak power supply to the board. If the reset-LED starts dimming instead of being clearly on or off, it is also a sign of too weak power supply to the board.
   If the reset-LED always light now when the *OEM Board* is mounted it is a sign of too weak power supply or a serious error on the *OEM Board*. When doing this test, be sure to not have any SWD/JTAG interface connected or any external source driving the reset signal.
   Verify that the reset-LED flash when pressing the reset push-button.

7. In section 4.3 it is described how to measure the current consumption of the *OEM Board*. Measure the voltage over J2 while pressing the reset push-button constantly. This will give the passive current consumption of the *OEM Board*, i.e., when the board is in reset. Valid range is 100-350mV (which corresponds to 20-70 mA). If current consumption is outside of this range, it is an indication that something can be wrong with the *OEM board*. Note that the valid range is quite wide. Being outside this does not necessarily mean that something is

wrong. More symptoms than this is needed to judge if a board is wrong, or not. Do not forget to press the reset push-button all the time during this measurement. Redo the test in case of uncertainty.

8. Now, measure the current consumption of the *OEM Board* when it is running an application. It is a problem to not have a specified application running because the current consumption can vary widely between different applications. The current consumption shall increase compared to the previous test (in case an application is running). By default the LPC3250 OEM Board is pre-programmed with the S1L bootloader. The consumption for the bootloader is in the range 450-550mV (which corresponds to 90-110mA). The current consumption varies between different sample applications depending on which peripherals that are active.
9. In most cases it is possible to power the board via the USB interface, J25 (the UART-to-USB bridge mini-B USB connector). Redo tests 3) to 8) above again with USB powering. Connect J25 directly to a PC with the USB cable included in the Developer's Kit.
   Note that not all PC:s and/or laptops can power the board. In such cases, a USB hub with independent powering can work. In this case, connect J25 to a USB Hub, which in turn is connected to a PC.

7.2 Contact with OEM Board MCU

The second step is to make sure that the LPC3250 MCU on the *OEM Board* is working.

Test that it is possible to download one of the precompiled test programs that exist on Embedded Artists support site. Sections 5.4.2 and 5.4.3 describe different ways of downloading applications to the board.

Optionally test that it is possible to download via the JTAG interface. A JTAG debug interface is then needed. The low-cost LPCXpresso target board series from NXP contains a JTAG debug interface called LPC-LINK that can be used for this. Other brands will also work.

7.3 Using Applications to Verify Correct Operation

The third step is to utilize the just downloaded (pre)compiled sample applications, one after one, to verify correct operation of individual parts of the *OEM Board* as well as the *OEM Base Board*. While doing these tests, be sure to not have anything connected to the system.

8 Further Information

The LPC3250 microcontroller is a complex circuit and there are a number of other documents with more information. The following documents are recommended as a complement to this document.

[1] NXP LPC3250 Datasheet
http://www.nxp.com/documents/data_sheet/LPC3220_30_40_50.pdf

[2] NXP LPC3250 User's Manual
http://www.nxp.com/documents/user_manual/UM10326.pdf

[3] NXP LPC3250 Errata
http://www.nxp.com/documents/errata_sheet/ES_LPC3250.pdf

[4] ARM926EJ-S Technical Reference Manual. Document identity: DDI0198E
http://infocenter.arm.com/help/topic/com.arm.doc.ddi0198e/DDI0198E_arm926ejs_r0p5_trm.pdf

[5] ARM Architecture Reference Manual. Document identity: DDI0100E
Book, Second Edition, edited by David Seal, Addison-Wesley: ISBN 0-201-73719-1
Also available in PDF form on the ARM Technical Publications CD

[6] ARM System Developer's Guide – Designing and Optimizing System Software, by A.N. Sloss, D Symes, C. Wright. Elsevier: ISBN 1-55860-874-5

[7] Embedded System Design on a Shoestring, by Lewin Edwards.
Newnes: ISBN 0750676094.

[8] GNU Manuals
http://www.gnu.org/manual/

[9] An Introduction to the GNU Compiler and Linker, by Bill Gatliff
http://www.billgatliff.com/

[10] LPCXpresso IDE: NXP's low-cost development platform for LPC families, which is an Eclipse-based IDE.
http://nxp.com/lpcxpresso

Note that there can be newer versions of the documents than the ones linked to here. Always check for the latest information/version.