EFM32®

... the world's most energy friendly microcontrollers

USER MANUAL

Development Kit EFM32-G8XX-DK

Feature rich development platform for evaluation, prototyping and application development for the EFM32 Gecko MCU family with the ARMCortex-M3 CPU core.

Main features;

- **Advanced Energy Monitoring provides real-time visibility into the energy consumption of an application or prototype design.**
- **Exchangeable prototyping board for custom application development**
- **On-board emulator with debug out functionality**

ENERGY® micro

www.energymicro.com

1 Introduction

1.1 Features

- Advanced Energy Monitoring system for precise current tracking.
- Special hardware configuration for isolation of the MCU power domain.
- Replaceable prototyping board for quick custom application development.
- Full feature USB debugger / emulator with debug out functionality.
- 3.5-inch TFT-LCD 320x240 pixel RGB color display.
- Board Controller for board configuration / signal routing.
- Single ended and differential ADC inputs.
- Line-in stereo audio input amplifier.
- Line-out stereo audio output amplifier.
- 2 RS232 connectors.
- 3-axis accelerometer.
- SPI Flash and microSD card reader (SPI mode).
- EEPROM.
- Temperature sensor.
- IrDA tranceiver.
- 256Kx16bit / 512KB parallel bus SRAM.
- 2Mx16 / 4MB parallel bus NOR Flash.
- Ambient light sensor and potmeter.
- 5 way joystick.
- 4 User buttons, 8-bit DIP switch and 16 user LEDs.

1.2 Board Configuration

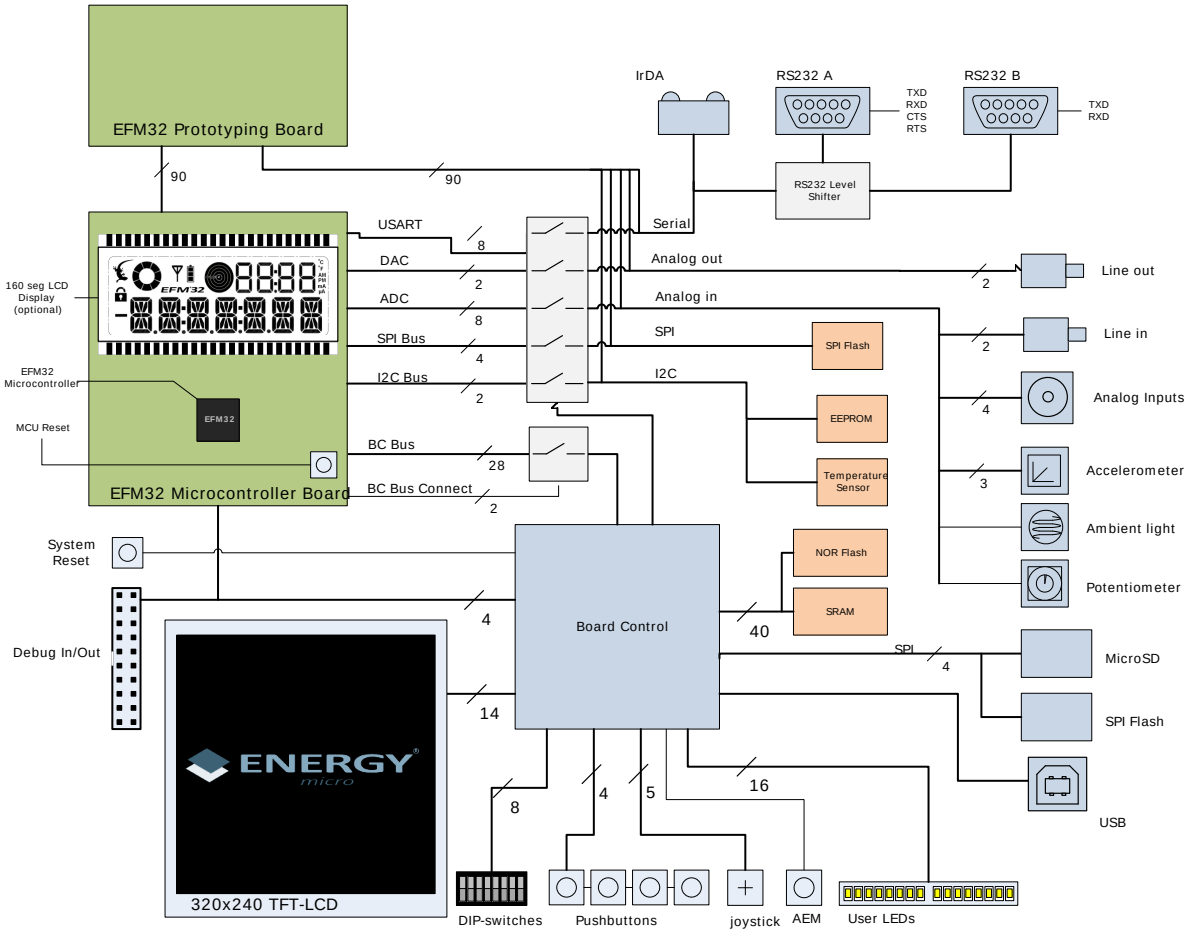
The EFM32-G8XX-DK is a highly flexible development kit. It offers many features and peripherals to the EFM32 through jumperless configuration. The different features on the kit are available as configured in the motherboard's Board Controller. Configuration is easily done by a simple API in the kit Board Support Package.

If none of the motherboard features are needed, configuration of the Board Controller is not necessary. All EFM32 GPIO pins are available on the prototyping board.

2 Kit Block Diagram

An overview of the Kit is shown in the block diagram below.

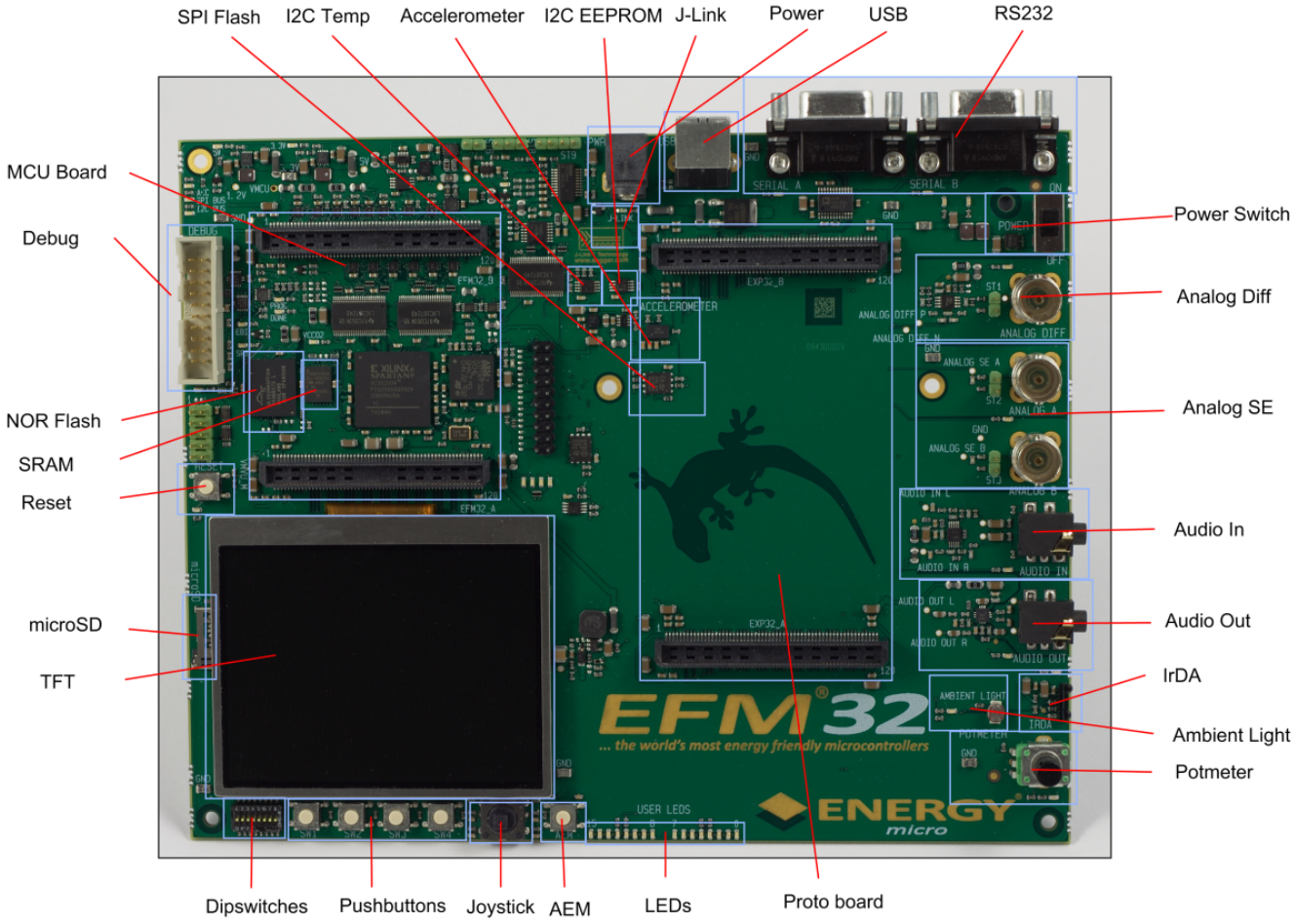
Figure 2.1. EFM32-G8XX-DK Block Diagram



3 Mainboard hardware layout

The layout of the EFM32-G8XX-DK mainboard is shown below.

Figure 3.1. EFM32-G8XX-DK hardware layout



4 Power supply

4.1 USB

The EFM32-G8XX-DK can get it's power from the standard USB 2.0 Type B port located on the motherboard. The USB hub the kit is connected to needs to be able to deliver 500 mA (5 unit loads).

4.2 External power supply

By using the DC jack plug located on the motherboard, the EFM32-G8XX-DK can be powered by an external power supply. The voltage must be 5V and the supply must be able to deliver 500mA.

The power jack dimensions should be a standard 5.5 mm outer diameter and 2.1 mm inner diameter. The tip is 5V and the sleeve is GND.

5 Reset infrastructure

5.1 MCU

The primary user reset for the MCU is the reset button on the MCU board. This will only reset the MCU. It can also be reset using the board controller, by writing to the RESET_MCU bit in the RESET register. Finally, it can be reset by debuggers.

5.2 Board controller

The board controller can be reset by pushing the reset button on the main board.

6 Peripherals

The development kit has a rich set of user programmable peripherals that allows most of the EFM32G on-chip peripherals to be evaluated and tested.

The registers referred to in this chapter are accessible using the kit Board Support Package. Refer to the BSP chapter in this manual to learn how to enable the motherboard peripherals. A reference to all the registers and their function is in the Board Controller chapter.

6.1 Pushbuttons

The state of the pushbuttons marked SW1 to SW4 can be read from the board controller, using the PUSHBUTTON register. The buttons are debounced by RC filters with a time constant of 1ms.

6.2 DIP switches

The dipswitch positions can be read from the board controller, using the DIPSWITCH register. The switches are not debounced.

6.3 Joystick

The joystick position can be read from the board controller, using the JOYSTICK register. The positions are debounced by RC filters with a time constant of 1ms.

6.4 LEDs

The user LEDs can be set by the board controller by writing to the LED register. The state of the LEDs can also be read back.

6.5 Differential analog input

This BNC input signal is converted to a differential signal by a differential operational amplifier using ground as reference.The op amp output common mode voltage is 1.65V, and also implements a low-pass active filter with a 3dB cutoff frequency of 4MHz.

The common mode voltage can be changed by adjusting the R5 and R12 resistors. It can also be controlled by the VCM pin on the EFM if a shunt resistor is soldered in place of R252 and R5 and R12 are removed.

The peripheral is connected directly to the EFM when the ANALOG_DIFF bit in the PERCTRL register in the board controller has been set.

6.6 Single ended analog inputs

This peripheral connects the two BNCs to the ADC on the EFM, and can be used as a single ended analog interface. It can also be used for digital I/O.

The peripheral is connected directly to the EFM when the ANALOG_SE bit in the PERCTRL register in the board controller has been set.

6.7 Line in / Audio in

This is an audio input amplifier with filter, and the output connects to the ADC of the EFM. The gain of the amplifier is 0 dB and the bias point is 1.65 V. The filter is a 3-pole linear phase MFB filter with a cutoff frequency of 20 kHz. In addition to the input amplifier and filter, the line in is equipped with a voltage divider resulting in 6dB attenuation.

The peripheral is connected directly to the EFM when the AUDIO_IN bit in the PERCTRL register in the board controller has been set.

6.8 Line Out / Audio out

This is an audio output amplifier with filter, and the input connects to the DAC of the EFM. The gain of the amplifier is 6 dB and is referenced to ground. The filter is a 3-pole linear phase MFB filter with a cutoff frequency (at -3 dB) is at 27 kHz.

The peripheral is connected directly to the EFM when the AUDIO_OUT bit in the PERCTRL register in the board controller has been set.

6.9 RS232

There are two RS232 connectors on the board which connects to the USART and LEUART of the EFM. The RS232 driver runs at 3.3 V and it is recommended that the MCU voltage is 3.3 V as well. Unpredicted behavior can occur if the MCU voltage is much lower than 3.3V and the RS232 driver is enabled.

The two RS232 drivers can be connected to the EFM individually by setting the RS232_A and RS232_B bits in the PERCTRL register in the board controller. The RS232_SHUTDOWN bit must also be cleared.

Note

When none of the RS232 drivers are in use, it is highly recommended that the driver is shut down by setting the RS232_SHUTDOWN bit in the PERCTRL register.

6.10 Accelerometer

This is a 3-axis accelerometer that connects to the ADC of the EFM. It outputs voltages proportional to the g-forces for each axis. There are two settings for the range. If ACCEL_GSEL in the PERCTRL register is cleared, the range is from 0 to 1.5 g, and when the bit is set the range is from 0 to 6 g.

The peripheral is connected directly to the EFM when the ACCEL bit in the PERCTRL register in the board controller has been set.

6.11 IrDA

This is a 115.2 kBit/s (SIR) IrDA transceiver with a range of up to 70 cm, and connects to the USART of the EFM.

The peripheral is connected directly to the EFM when the IRDA bit in the PERCTRL register in the board controller has been set.

6.12 Potmeter

This is a potmeter pulled to 3.3 V by a 10k resistor, and it is connected to the ADC of the EFM. Using the potmeter the output of this peripheral can be adjusted from 0 V to 3 V.

The peripheral is connected directly to the EFM when the POTMETER bit in the PERCTRL register in the board controller has been set.

6.13 Ambient light sensor

This is an LDR in series with a 10k resistor, and it is connected to the ADC of the EFM. The output voltage of the sensor ranges from 0.1 V to 2 V, increasing with the amount of light.

The peripheral is connected directly to the EFM when the AMBIENT bit in the PERCTRL register in the board controller has been set.

6.14 I2C EEPROM

The 2 KB I2C EEPROM is connected to the I2C module of the EFM. The maximum bus speed is 400 kHz and the address is 0xA0.

The peripheral is connected directly to the EFM when the I2C bit in the PERCTRL register in the board controller has been set.

6.15 I2C Temperature sensor

The I2C temperature sensor is connected to the I2C module of the EFM. Temperature range of the sensor is -55 °C to +125 °C. The maximum bus speed is 400 kHz and the address is 0x90.

The peripheral is connected directly to the EFM when the I2C bit in the PERCTRL register in the board controller has been set.

6.16 SPI Flash

A 16 MBit SPI flash is connected to the SPI module of the EFM.

The peripheral is connected directly to the EFM when the SPI bit in the PERCTRL register in the board controller has been set. To route the chip select correctly, the FLASH bit in the BC_SPI_CFG register in the board controller must also be set.

6.17 microSD

A microSD slot is connected to the SPI module of the EFM.

The peripheral is connected directly to the EFM when the SPI bit in the PERCTRL register in the board controller has been set. To route the chip select correctly, the MICROSD bit in the BC_SPI_CFG register in the board controller must also be set.

6.18 TFT LCD

The TFT LCD can be accessed from the EFM through the board controller. The interface can be configured to be either 9 bit or 16 bit. This selection is done by setting the 16BIT bit in the DISPLAY_CTRL register in the board controller. It is also possible to use the SPI interface, but then the R112 resistor must be moved over to the position of R112.

Note

16 bit or SPI interface options are currently not supported by the board controller.

6.19 SRAM

The 512 KB SRAM can be accessed from the EFM through the board controller. The data width is either 8 or 16 bit, depending on the access method.

6.20 NOR Flash

The 4 MB NOR Flash can be accessed from the EFM through the board controller. The data width is either 8 or 16 bit, depending on the access method.

7 Board Support Package

The Board Support Package (BSP) is a set of C source and header files that enables easy access to, and control over board specific features and peripherals.

The package defines an API for direct access to the board controller registers, as well as regular function calls for the most frequently used features.

7.1 Installation location

When installing the complete software package for the kit, the BSP will be installed under the main installation directory, typically in a location such as

```
C:\Program Files\Energy Micro\boards\EFM32_Gxxx_DK\bsp\
```

or something similar. All files in the board support package is prefixed by dvk.

7.2 Resource usage

The BSP can be configured to use 1 of 2 access methods

- SPI - USART2 Serial Peripheral Interface
- EBI - External Bus Interface

SPI and EBI have different requirements regarding pin usage, see table below:

Table 7.1. GPIO Usage

GPIO Port	SPI Pins	EBI Pins
A		0-6, 15
B		
C	2-5,13	12
D		
E		8-15
F		2-5

The advantage of EBI over SPI, is that EBI access is a fast, directly memory mapped register access, while SPI will add a synchronous two way transfer over a slower SPI interface. The disadvantage of EBI is that it will consume a lot more I/O pins than SPI. The DVK will by default be configured to use SPI, to enable EBI a small SPI initialization routine needs to be called once (per restart of the entire kit, not per restart of the EFM32).

For DK part number EFM32_G8xx_DK with LCD controller, SPI is the only option as EBI and LCD cannot be combined.

You must take care in not using these pins for other purposes after initialization of the BSP, as conflicts and unpredictable behavior will result. You can disable the DVK interfaces after you have set your configuration.

7.3 Application Programming Interface

To use the BSP, include the Development Kit header file, like this:

```
#include "dvk.h"
```

Depending on the part number defined in your project as a build option, the DVK will default to the most common access method for your MCU module according to part number. If you need to override the default board control access method, you can define the access method by overriding the default with DVK_SPI_CONTROL or DVK_EBI_CONTROL defines, such as

```
#define DVK_SPI_CONTROL
#include "dvk.h"
```

All functions in the BSP are prefixed with DVK_. The main initialization routine is defined as

```
void DVK_init(void);
```

and must be called before any access to the DVK-functions. To disable the BSP, call

```
void DVK_disable(void);
```

You can access all registers with the generic functions

```
void DVK_writeRegister(volatile uint16_t *addr, uint16_t data);
uint16_t DVK_readRegister(volatile uint16_t *addr);
```

Usable addresses for these functions, including bit fields are defined in the header file

```
dvk_bcregisters.h
```

The functions

```
void DVK_enablePeripheral( DVKPeripheral peri );
void DVK_disablePeripheral( DVKPeripheral peri );
```

can be used to toggle access/peripheral switches to all peripherals on the DVK. See the "peripheral" example application for usage.

In addition to these main functions, full documentation of the complete API is included in the Doxygen/ HTML documentation of the installed package.

7.4 Example Applications

There are a number of example applications to illustrate the usage of the DVK API. You will find these with their corresponding IAR Embedded Workbench and Keil MDK-ARM project files under

```
C:\Program Files\Energy Micro\boards\EFM32_Gxxx_DK\examples\
```

The examples include, among others

- blink - Simple application using the DVK and it's LED control API
- peripherals - Toggles peripherals on and off, indicated by LEDs on the board
- joystick - Use DVK LED, joystick and interrupt APIs for indicating DVK joystick movement

The example files above have been configured to be built for both the EFM32_G2xx_DK and EFM32_G8xx_DK kits, with the EFM32G290F128 and EFM32G890F128 part numbers. Select the project that matches your setup to ensure correct operation.

7.5 How to include in your own applications

The easiest way to include the BSP in your application is to base your work on one of the example applications, for instance the easy "blink" demonstration. The following items are recommended for correct configuration:

1. Make sure you define the correct part number (e.g. EFM32G290F128) as a preprocessor defined symbol
2. Make sure you define the correct part number (e.g. EFM32G290F128) for your IAR EWARM / Keil MDK-ARM project
3. Add and include the EFM32_CMSIS-files (startup_efm32.s, system_efm32.c, core_cm3.c) to your project
4. Add and include _all_ BSP package .c-files, with the dvk-prefix to your project
5. Configure include paths to point at the CMSIS/CM3/CoreSupport and CMSIS/CM3/DeviceSupport/EnergyMicro/EFM32 directories
6. Configure include paths to point to the dvk/bsp directory

Make sure you call "DVK_init()" early at startup, and you should be all set.

7.6 Chip errata

Early versions of the development kit are shipped with EFM32 Engineering Samples on the MCU modules. There has been updates to configuration and reset values that needs to be configured correctly on these early parts. We recommend always starting your application with a call to

```
#include "efm32_chip.h"

CHIP_Init();
```

to ensure correct and stable behavior. This function is found in EFM32_CMSIS (version 1.3.0 or later). See the BSP examples for details. We recommend also to download and read the latest errata from the Energy Micro website for your part number.

8 Configuration

Some parameters can be configured using the GUI. The other parameters, such as peripheral control, can be controlled by software. See the Board Controller chapter for details.

8.1 MCU voltage

The MCU voltage can be set by entering the CFG page from the main page. Use the joystick to navigate to VMCU and set your desired voltage by moving the joystick sideways. The measured VMCU can be read at the bottom of this screen. Push Save to store your settings.

8.2 Debug settings

The debug routing can be set by entering the CFG page from the main page. Use the joystick to navigate to Debug Control and set your desired mode by moving the joystick sideways. Push Save to store your settings.

See the debug chapter to read more about the different modes.

8.3 Peripheral configuration

The peripheral configuration can be set by entering the CFG page from the main page and then entering the Peri page. All peripherals connected to the EFM can be en- or disabled individually using the list displayed in the GUI.

8.4 Program MCU

To program the MCU with files uploaded to the flash, enter the Flash page from the main page. The list of available binaries are shown, and one of them can be selected by using the joystick. When the desired binary has been selected, push Flash to program the MCU. While programming, a new page shows with a progress bar. A status message appears when the programming is finished.

Note

The debug mode has to be set to MCU for this to work.

8.5 Upload files

To upload files, Gecko Commander must be used. This is an executable that can be found in the install location, typically:

```
C:\Program Files\Energy Micro\EFM32 Kit Package\GeckoCmd\Gecko.exe
```

After launching the program, execute this command in Gecko Commander:

```
put your_binary_file.bin \flash\your_binary_file.bin
```

To see the commands available, execute this command for help:

```
h
```

9 Advanced Energy Monitor

9.1 AEM Display

To enter AEM from the main page, press the pushbutton under the display labeled AEM. If the EFM is using the display, press the AEM button at the right side of the display, and the board controller will take control of the display and show the AEM. To return to EFM control, simply press the AEM button once more.

When the AEM is entered, you will get a real-time graphical display of the current consumption of the EFM and other circuits powered by the VMCU power rail. The AEM display mainly features a plot of current the consumption over time. It also displays the latest sampled current consumption and voltage.

9.2 AEM configuration

There are several parameters that can be configured on the AEM. To configure AEM, first enter the AEM page. Then push the button labeled CFG. In the CFG page, you can adjust the scale of the time axis of the current plot.

9.3 AEM theory of operation

In order to be able to measure currents ranging from 0.1uA to 50mA (114dB dynamic range), two current sense amplifiers are utilized. The amplifiers measure voltage drop over a small series resistor and translates this into a current. Each amplifier is adjusted for current measurement in a specific range. The ranges for the amplifiers overlap and a change between the two occurs when the current is 200uA. To reduce noise, averaging of the samples is performed before the current measurement is presented in the AEM GUI.

During startup of the kit and when VMCU is changed, a calibration of the AEM is performed. This calibration compensates for the offset error in the sense amplifiers. In order for the calibration to be correct, no load should be connected between the pins of ST6 during calibration.

9.4 AEM accuracy and performance

The Advanced Energy Monitor is capable of measuring currents in the range of 0.1uA to 50mA. For currents above 200uA, the AEM is accurate within 0.1mA. When measuring currents below 200uA, the accuracy increases to 1uA. Even though the absolute accuracy is 1uA in the sub 200uA range, the AEM is able to detect changes in the current consumption as small as 100nA. The measurement bandwidth of the AEM is 60Hz when measuring currents below 200uA and 120Hz when measuring currents above 200uA. The table below summarizes accuracy of the two current sense amplifiers in different ranges.

Table 9.1. AEM accuracy

Current range	Low gain amplifier accuracy	High gain amplifier accuracy
50mA	0.1mA	-
1mA	0.1mA	-
200uA	0.01mA	1uA
10uA	-	0.1uA
1uA	-	0.1uA

Note

In order for the AEM to work correctly, VMCU should be 3.0V or higher.

10 Board controller

The board controller (BC) consists of the control MCU and an FPGA. The FPGA is essentially a programmable multiplexer that allows the resources on the board to be shared between the EFM and the control MCU. It also enables jumperless peripheral configuration. The control MCU implements the built-in debugger, the AEM and performs housekeeping tasks.

To use the board controller for your application, the Board Support Package (BSP) must be installed. See the BSP chapter to find out how.

10.1 Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	BC_BC_CFG	RW	Board Controller Config register
0x002	BC_EM	RW	Energy Mode register
0x004	BC_MAGIC	R	Magic number
0x006	BC_LED	RW	User LEDs register
0x008	BC_PUSHBUTTON	R	User pushbutton status register
0x00A	BC_DIPSWITCH	R	User dipswitch status register
0x00C	BC_JOYSTICK	R	Joystick state register
0x00E	BC_AEM	R	AEM button status register
0x010	BC_DISPLAY_CTRL	RW	Display control register
0x012	BC_EBI_CFG	RW	EBI configuration register
0x014	BC_BUS_CFG	RW	BUS configuration register
0x018	BC_PERCTRL	RW	Peripheral control register
0x01A	BC_AEMSTATE	R	AEM button status register
0x01C	BC_SPI_CFG	RW	SPI configuration register
0x01E	BC_RESET	RW	Reset register
0x020	BC_ADC_START	RW	ADC start byte register
0x022	BC_ADC_STATUS	R	ADC status register
0x024	BC_ADC_DATA	R	ADC data register
0x028	BC_HW_VERSION	R	Hardware version register
0x02A	BC_FW_BUILDNO	R	Firmware build number
0x02C	BC_FW_VERSION	R	Firmware version register
0x02E	BC_SCRATCH_COMMON	RW	Common scratch register
0x030	BC_SCRATCH_EFM0	RW	EFM scratch register 0
0x032	BC_SCRATCH_EFM1	RW	EFM scratch register 1
0x034	BC_SCRATCH_EFM2	RW	EFM scratch register 2
0x036	BC_SCRATCH_EFM3	RW	EFM scratch register 3
0x038	BC_SCRATCH_BC0	RW	BC scratch register 0
0x03A	BC_SCRATCH_BC1	RW	BC scratch register 1
0x03C	BC_SCRATCH_BC2	RW	BC scratch register 2
0x03E	BC_SCRATCH_BC3	RW	BC scratch register 3
0x040	BC_INTFLAG	RW	Interrupt flags
0x042	BC_INTEN	RW	Interrupt enables

10.2 Register Description

10.2.1 BC_BC_CFG - Board Controller Config register

Offset	Bit Position															
0x000	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0
Access																RW
Name																BC_CFG

Bit	Name	Reset	Access	Description
15:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
0	BC_CFG	0	RW	**Board controller configuration**

Use this bit to change between SPI and EBI interface on the board controller

Value	Mode	Description
0	SPI	The BC is configured to use the SPI interface.
1	EBI	The BC is configured to use the EBI interface.

10.2.2 BC_EM - Energy Mode register

Offset	Bit Position															
0x002	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	
Access															RW	
Name															EM	

Bit	Name	Reset	Access	Description
15:3	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
2:0	EM	0x0	RW	**Energy Mode register**

This register is used to store the Energy Mode the EFM is running in.

10.2.3 BC_MAGIC - Magic number

Offset	Bit Position															
0x004	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0xEF32								
Access								R								
Name								MAGIC								

Bit	Name	Reset	Access	Description
15:0	MAGIC	0xEF32	R	**Magic number**
	This register can be used to test the interface.			

10.2.4 BC_LED - User LEDs register

Offset	Bit Position															
0x006	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	RW															
Name	LED															

Bit	Name	Reset	Access	Description
15:0	LED	0x0000	RW	**User LED register**
	Write to this register to change the DVK user leds			

10.2.5 BC_PUSHBUTTON - User pushbutton status register

Offset	Bit Position															
0x008	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset														0x0		
Access														R		
Name														PUSHBUTTON		

Bit	Name	Reset	Access	Description
15:4	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
3:0	PUSHBUTTON	0x0	R	**User pushbutton status register**
	Read this register to determine the state of the pushbuttons			

10.2.6 BC_DIPSWITCH - User dipswitch status register

Offset	Bit Position															
0x00A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x00							
Access									R							
Name									DIPSWITCH							

Bit	Name	Reset	Access	Description
15:8	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
7:0	DIPSWITCH	0x00	R	**User dipswitch status register**
	Read this register to determine the state of the dipswitch			

10.2.7 BC_JOYSTICK - Joystick state register

Offset	Bit Position															
0x00C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0	0	0	0	0
Access												R	R	R	R	R
Name												CENTER	LEFT	UP	RIGHT	DOWN

Bit	Name	Reset	Access	Description
15:5	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
4	CENTER	0	R	**Joystick CENTER switch state register**
	Read this register to get the status of the center switch of the joystick.			
3	LEFT	0	R	**Joystick LEFT switch state register**
	Read this register to get the status of the left switch of the joystick.			
2	UP	0	R	**Joystick UP switch state register**
	Read this register to get the status of the up switch of the joystick.			
1	RIGHT	0	R	**Joystick RIGHT switch state register**
	Read this register to get the status of the right switch of the joystick.			
0	DOWN	0	R	**Joystick DOWN switch state register**
	Read this register to get the status of the down switch of the joystick.			

10.2.8 BC_AEM - AEM button status register

Offset	Bit Position															
0x00E	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0
Access																R
Name																AEM

Bit	Name	Reset	Access	Description
15:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
0	AEM	0	R	**AEM button status register**
	Read this register to determine the state of the AEM button			

10.2.9 BC_DISPLAY_CTRL - Display control register

Offset	Bit Position															
0x010	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0	0
Access															RW	RW
Name															POWER_ENABLE	RESET

Bit	Name	Reset	Access	Description
15:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
1	POWER_ENABLE	0	RW	**Display power enable**
	Set this bit to enable power to the TFT display			
0	RESET	0	RW	**Display reset**
	Set this bit to put the TFT display into reset mode			

10.2.10 BC_EBI_CFG - EBI configuration register

Offset	Bit Position															
0x012	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	
Access															RW	
Name															EBI_CFG	

Bit	Name	Reset	Access	Description
15:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
1:0	EBI_CFG	0x0	RW	**EBI configuration register**
	Set this register to configure the BC EBI interface to match the configuration in your application			

Value	Mode	Description
0	EBI_16_16	The BC EBI is in a 16 address bits and 16 data bits configuration
1	EBI_8_8	The BC EBI is in a 8 address bits and 8 data bits configuration
2	EBI_24_8	The BC EBI is in a 24 address bits and 8 data bits configuration

10.2.11 BC_BUS_CFG - BUS configuration register

Offset	Bit Position															
0x014	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	
Access															RW	
Name															BUS_CFG	

Bit	Name	Reset	Access	Description
15:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
1:0	BUS_CFG	0x0	RW	**BUS configuration register**

Set this register to configure which bus has access to the SRAM, Nor Flash and TFT display.

Value	Mode	Description
0	FSMC	The FSMC has access
1	EBI	The EBI interface has access
2	SPI	The SPI interface has access

10.2.12 BC_PERCTRL - Peripheral control register

Offset	Bit Position															
0x018	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	IRDA_SHUTDOWN	RS232_SHUTDOWN	ACCEL_SELFTEST	ACCEL_GSEL	AUDIO_IN	AUDIO_OUT	ANALOG_DIFF	ANALOG_SE	IRDA	I2C	SPI	RS232B	RS232A	POTMETER	AMBIENT	ACCEL

Bit	Name	Reset	Access	Description
15	IRDA_SHUTDOWN	0	RW	**Shut down IrDA transceiver**

Set this bit to shut down the IrDA transceiver

Bit	Name	Reset	Access	Description
14	RS232_SHUTDOWN	0	RW	**Shut down RS232 driver**

Set this bit to shut down the RS232 driver. It is strongly recommended that this is done when the application does not use RS232.

Bit	Name	Reset	Access	Description
13	ACCEL_SELFTEST	0	RW	**Accelerometer selftest mode**

Set this bit to put the accelerometer into selftest mode

Bit	Name	Reset	Access	Description
12	ACCEL_GSEL	0	RW	**Accelerometer g-select**

Use this bit to configure the g-range of the accelerometer

Value	Mode	Description
0	LOW	The g-range is up to 1.5g
1	HIGH	The g-range is up to 6g

Bit	Name	Reset	Access	Description
11	AUDIO_IN	0	RW	**Audio in connect**

Set this bit to connect the audio in amplifier to the EFM

Bit	Name	Reset	Access	Description
10	AUDIO_OUT	0	RW	**Audio out connect**

Set this bit to connect the audio out amplifier to the EFM

Bit	Name	Reset	Access	Description
9	ANALOG_DIFF	0	RW	**Analog differential input connect**

Set this bit to connect the analog differential inputs to the EFM

Bit	Name	Reset	Access	Description
8	ANALOG_SE	0	RW	**Analog single ended input connect**

Set this bit to connect the analog single ended inputs to the EFM

Bit	Name	Reset	Access	Description
7	IRDA	0	RW	**IrDA connect**

Set this bit to connect the IrDA transceiver to the EFM

Bit	Name	Reset	Access	Description
6	I2C	0	RW	**I^2C bus connect**

Set this bit to connect the I^2C devices to the EFM

Bit	Name	Reset	Access	Description
5	SPI	0	RW	**SPI bus connect**

Bit	Name	Reset	Access	Description
	Set this bit to connect the SPI devices to the EFM			
4	RS232B	0	RW	**RS232B connect**
	Set this bit to connect the RS232 B to the EFM			
3	RS232A	0	RW	**RS232A connect**
	Set this bit to connect the RS232 A to the EFM			
2	POTMETER	0	RW	**Potmeter connect**
	Set this bit to connect the potmeter to the EFM			
1	AMBIENT	0	RW	**Ambient light sensor connect**
	Set this bit to connect the ambient light sensor to the EFM			
0	ACCEL	0	RW	**Accelerometer connect**
	Set this bit to connect the accelerometer chip to the EFM			

10.2.13 BC_AEMSTATE - AEM button status register

Offset	Bit Position															
0x01A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0
Access																R
Name																AEM

Bit	Name	Reset	Access	Description
15:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
0	AEM	0	R	**AEM state register**
	When this bit is 0, the BC can take control of the TFT display, and when 1 the EFM can take control. This bit is toggled by the AEM button push			

10.2.14 BC_SPI_CFG - SPI configuration register

Offset	Bit Position															
0x01C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0
Access																RW
Name																SPI_CFG

Bit	Name	Reset	Access	Description
15:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
0	SPI_CFG	0	RW	**SPI configuration register**
	This register selects which SPI module will receive the CS signal			

Value	Mode	Description
0	FLASH	The SPI CS is routed to the SPI Flash

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	1	MICROSD		The SPI CS is routed to the microSD

10.2.15 BC_RESET - Reset register

Offset	Bit Position															
0x01E	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0	0
Access															RW	RW
Name															EFM	FLASH

Bit	Name	Reset	Access	Description
15:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
1	EFM	0	RW	**EFM reset signal**
	Set this bit to put the EFM into a reset state			
0	FLASH	0	RW	**Flash reset signal**
	Set this bit to put the Nor flash into a reset state			

10.2.16 BC_ADC_START - ADC start byte register

Offset	Bit Position															
0x020	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x00							
Access									RW							
Name									ADC_START							

Bit	Name	Reset	Access	Description
15:8	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
7:0	ADC_START	0x00	RW	**ADC start byte**
	Write this byte to start a conversion on the voltage monitor ADC. The content of the byte is equal to the start byte for the ADC itself.			

10.2.17 BC_ADC_STATUS - ADC status register

Offset	Bit Position															
0x022	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0
Access																R
Name																BUSY

Bit	Name	Reset	Access	Description
15:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
0	BUSY	0	R	**ADC status byte**
	Read this bit to determine the state of the ADC conversion			

Value	Mode	Description
0	DONE	The ADC is not doing a conversion
1	BUSY	The ADC is busy doing a conversion

10.2.18 BC_ADC_DATA - ADC data register

Offset	Bit Position															
0x024	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	R															
Name	ADC_START															

Bit	Name	Reset	Access	Description
15:0	ADC_START	0x0000	R	**ADC data register**
	This register contains the result of the latest conversion			

10.2.19 BC_HW_VERSION - Hardware version register

Offset	Bit Position															
0x028	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0							0x0		
Access							R							R		
Name							PCB							BOARD		

Bit	Name	Reset	Access	Description
15:11	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
10:8	PCB	0x0	R	**PCB revision**
	Read these bits to determine the PCB revision			
7:4	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
3:0	BOARD	0x0	R	**Board revision**
	Read these bits to determine the board revision			

10.2.20 BC_FW_BUILDNO - Firmware build number

Offset	Bit Position															
0x02A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	R															
Name	BUILDNO															

Bit	Name	Reset	Access	Description
15:0	BUILDNO	0x0000	R	**Board revision**
	Read this register to determine the firmware build number			

10.2.21 BC_FW_VERSION - Firmware version register

Offset	Bit Position															
0x02C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0				0x00							
Access	R				R				R							
Name	MAJOR				MINOR				PATCHLEVEL							

Bit	Name	Reset	Access	Description
15:12	MAJOR	0x0	R	**Firmware major revision**
	Read these bits to determine the major revision			
11:8	MINOR	0x0	R	**Firmware minor revision**
	Read these bits to determine the minor revision			
7:0	PATCHLEVEL	0x00	R	**Firmware patch level**
	Read these bits to determine the patch level			

10.2.22 BC_SCRATCH_COMMON - Common scratch register

Offset	Bit Position															
0x02E	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	RW															
Name	SCRATCH_COMMON															

Bit	Name	Reset	Access	Description
15:0	SCRATCH_COMMON	0x0000	RW	**Common scratch register**
	This register can be used as a scratch register for both the EFM and the board controller.			

10.2.23 BC_SCRATCH_EFM0 - EFM scratch register 0

Offset	Bit Position															
0x030	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	RW															
Name	SCRATCH_EFM0															

Bit	Name	Reset	Access	Description
15:0	SCRATCH_EFM0	0x0000	RW	**EFM scratch register 0**
	This register can be used as a scratch register for the EFM. The board controller has read only access.			

10.2.24 BC_SCRATCH_EFM1 - EFM scratch register 1

Offset	Bit Position															
0x032	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	RW															
Name	SCRATCH_EFM1															

Bit	Name	Reset	Access	Description
15:0	SCRATCH_EFM1	0x0000	RW	**EFM scratch register 1**
	This register can be used as a scratch register for the EFM. The board controller has read only access.			

10.2.25 BC_SCRATCH_EFM2 - EFM scratch register 2

Offset	Bit Position															
0x034	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	RW															
Name	SCRATCH_EFM2															

Bit	Name	Reset	Access	Description
15:0	SCRATCH_EFM2	0x0000	RW	**EFM scratch register 2**
	This register can be used as a scratch register for the EFM. The board controller has read only access.			

10.2.26 BC_SCRATCH_EFM3 - EFM scratch register 3

Offset	Bit Position															
0x036	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	RW															
Name	SCRATCH_EFM3															

Bit	Name	Reset	Access	Description
15:0	SCRATCH_EFM3	0x0000	RW	**EFM scratch register 3**
	This register can be used as a scratch register for the EFM. The board controller has read only access.			

10.2.27 BC_SCRATCH_BC0 - BC scratch register 0

Offset	Bit Position															
0x038	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	RW															
Name	SCRATCH_BC0															

Bit	Name	Reset	Access	Description
15:0	SCRATCH_BC0	0x0000	RW	**BC scratch register 0**
	This register can be used as a scratch register for the BC. The EFM has read only access.			

10.2.28 BC_SCRATCH_BC1 - BC scratch register 1

Offset	Bit Position															
0x03A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	RW															
Name	SCRATCH_BC1															

Bit	Name	Reset	Access	Description
15:0	SCRATCH_BC1	0x0000	RW	**BC scratch register 1**
	This register can be used as a scratch register for the BC. The EFM has read only access.			

10.2.29 BC_SCRATCH_BC2 - BC scratch register 2

Offset	Bit Position															
0x03C	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	RW															
Name	SCRATCH_BC2															

Bit	Name	Reset	Access	Description
15:0	SCRATCH_BC2	0x0000	RW	**BC scratch register 2**
	This register can be used as a scratch register for the BC. The EFM has read only access.			

10.2.30 BC_SCRATCH_BC3 - BC scratch register 3

Offset	Bit Position															
0x03E	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0000															
Access	RW															
Name	SCRATCH_BC3															

Bit	Name	Reset	Access	Description
15:0	SCRATCH_BC3	0x0000	RW	**BC scratch register 3**
	This register can be used as a scratch register for the BC. The EFM has read only access.			

10.2.31 BC_INTFLAG - Interrupt flags

Offset	Bit Position															
0x040	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0	0	0	0
Access													RW	RW	RW	RW
Name													AEM	JOYSTICK	DIP	PB

Bit	Name	Reset	Access	Description
15:4	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
3	AEM	0	RW	**AEM interrupt flag**
	This bit is set when the AEM button is pushed or released. It will assert an interrupt to the EFM if the interrupt has been enabled. This bit is cleared by writing a 1 to it.			
2	JOYSTICK	0	RW	**Joystick interrupt flag**
	This bit is set when the joystick changes position, or is pushed or released. It will assert an interrupt to the EFM if the interrupt has been enabled. This bit is cleared by writing a 1 to it.			
1	DIP	0	RW	**Dipswitch interrupt flag**
	This bit is set when any of the dipswitch positions are changed. It will assert an interrupt to the EFM if the interrupt has been enabled. This bit is cleared by writing a 1 to it.			
0	PB	0	RW	**Pushbuttons interrupt flag**
	This bit is set when any of the 4 pushbutton are pushed or released. It will assert an interrupt to the EFM if the interrupt has been enabled. This bit is cleared by writing a 1 to it.			

10.2.32 BC_INTEN - Interrupt enables

Offset	Bit Position															
0x042	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0	0	0	0
Access													RW	RW	RW	RW
Name													AEM	JOYSTICK	DIP	PB

Bit	Name	Reset	Access	Description
15:4	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information on ???*		
3	AEM	0	RW	**AEM interrupt enable**
	If this bit is set, an interrupt is asserted when the corresponding interrupt flag is set.			
2	JOYSTICK	0	RW	**Joystick interrupt enable**
	If this bit is set, an interrupt is asserted when the corresponding interrupt flag is set.			
1	DIP	0	RW	**Dipswitch interrupt enable**
	If this bit is set, an interrupt is asserted when the corresponding interrupt flag is set.			
0	PB	0	RW	**Pushbuttons interrupt enable**
	If this bit is set, an interrupt is asserted when the corresponding interrupt flag is set.			

11 Connectivity

11.1 Resource connections

In general, all ports are routed from the BRD3300A to the prototyping board. That means that Port A on the EFM is replicated on Port A on the prototyping board. In addition to that, a lot of EFM pins has been connected to other resources on the board, some using switches and some permanent.

The following table describes the connections between the EFM on the BRD3300A and the kit resources.

Table 11.1. Connections

MCU	MCU function	EFM32_I	Connection	EXP32_E	Proto port	Function
A0	LCD_SEG13					
A1	LCD_SEG14					
A2	LCD_SEG15					
A3	LCD_SEG16					
A4	LCD_SEG17					
A5	LCD_SEG18					
A6	LCD_SEG19					
A9	LCD_SEG37					
A15	LCD_SEG12					
B0	LCD_SEG32					
B1	LCD_SEG33					
B2	LCD_SEG34					
B3	LCD_SEG20					
B4	LCD_SEG21					
B5	LCD_SEG22					
B6	LCD_SEG23					
B7	US1_CLK #0	B53	RS232_A	B53	J4	RS232_A_#CTS
B8	US1_CS #0	B54	RS232_A	B54	J5	RS232_A_#RTS
B11	DAC0_OUT0	B48	AUDIO_OUT	B48	I7	AUDIO_OUT_RIGHT
B12	DAC0_OUT1	B49	AUDIO_OUT	B49	I8	AUDIO_OUT_LEFT
C0	US1_TX #0	B55	RS232_A	B55	J6	RS232_A_TX
C1	US1_RX #0	B52	RS232_A	B52	J3	RS232_A_RX
C2	US2_TX #0	B42	BC_BUS_CONNECT_SPI	B42	H13	BC_BUS26
C3	US2_RX #0	B43	BC_BUS_CONNECT_SPI	B43	H14	BC_BUS27
C4	US2_CLK #0	B40	BC_BUS_CONNECT_SPI	B40	H11	BC_BUS24
C5	US2_CS #0	B41	BC_BUS_CONNECT_SPI	B41	H12	BC_BUS25
C6	LEU1_TX #0	B62	RS232_B	B62	J9	RS232_B_TX

MCU	MCU function	EFM32_I	Connection	EXP32_E	Proto port	Function
C7	LEU1_RX #0	B63	RS232_B	B63	J10	RS232_B_RX
C8	US0_CS #2	B65	SPI	B65	J12	SPI_BUS_#CS
C9	US0_CLK #2	B64	SPI	B64	J11	SPI_BUS_SCLK
C10	US0_RX #2	B67	SPI	B67	J14	SPI_BUS_MISO
C11	US0_TX #2	B66	SPI	B66	J13	SPI_BUS_MOSI
C12		B72	Connected	B72	H15	BC_BUS_CONNECT_EBI
C13			Connected			FPGA_#INT
C15		B79	Connected	B79	J18	MCUDBG_TDO_SWO
D0	ADC_CH0	B50	AUDIO_IN	B50	I9	AUDIO_IN_RIGHT
D1	ADC_CH1	B51	AUDIO_IN	B51	I10	AUDIO_IN_LEFT
D2	ADC_CH2	B56	ACCEL	B56	I11	ACCEL_XOUT
D3	ADC_CH3	B57	ACCEL	B57	I12	ACCEL_YOUT
D4	ADC_CH4	B58	ACCEL	B58	I13	ACCEL_ZOUT
D5	ADC_CH5	B70	POTMETER	B70	I15	SENSOR_POTMETER
D5	ADC_CH5	B71	AMBIENT	B71	I16	SENSOR_LIGHT
D6	ADC_CH6	B46	ANALOG_DIFF	B46	I5	ANALOG_DIFF_N
D7	ADC_CH7	B47	ANALOG_DIFF	B47	I6	ANALOG_DIFF_P
D9	LCD_SEG28					
D10	LCD_SEG29					
D11	LCD_SEG30					
D12	LCD_SEG31					
D14	I2C0_SDA #3	B68	I2C	B68	J15	I2C_BUS_SDA
D15	I2C0_SCL #3	B69	I2C	B69	J16	I2C_BUS_SCL
E4	LCD_COM0					
E5	LCD_COM1					
E6	LCD_COM2					
E7	LCD_COM3					
E8	LCD_SEG4					
E9	LCD_SEG5					
E10	LCD_SEG6					
E11	LCD_SEG7					
E12	LCD_SEG8					
E13	LCD_SEG9					
E14	LCD_SEG10					

MCU	MCU function	EFM32_I	Connection	EXP32_E	Proto port	Function
E15	LCD_SEG11					
F0	DGB_SWCLK	B77	Connected	B77	I18	MCUDBG_TCK_SWCLK
F1	DGB_SWDIO	B78	Connected	B78	J17	MCUDBG_TCK_SWDIO
F2	LCD_SEG0					
F3	LCD_SEG1					
F4	LCD_SEG2					
F5	LCD_SEG3					
F6	LCD_SEG24					
F7	LCD_SEG25					
F8	LCD_SEG26					
F9	LCD_SEG72					

Table 11.2. Nomenclature

Name	Description
MCU	The pin name of the MCU
MCU function	The I/O function on that pin that is used for this resource
EFM32_B	The corresponding pin number on the EFM32_B connector. If this is empty, the signal is not routed out from the BRD3300A
Connection	Which API function is used to make the connection.
EXP32_B	The corresponding pin number on the EXP32_B connector. If this is empty, the signal is not routed out from the BRD3300A
Proto port	The corresponding pin on the proto board
Function	The name of the kit resource

12 Connectors

12.1 EFM32 and EXP32 connectors

The EFM32 connector is used to connect the MCU plugin board to the main board, and the EXP32 connector is used to connect the prototype card to the main board. The EFM32_A connector and the first 16 pins of the EFM32_B connector is directly connected to the corresponding pins on the EXP32_A and EXP32_B connectors. These signals duplicate the MCU ports to the port headers on the prototype board.

Some care must be taken when fitting and removing the plugin cards. Make sure that the plugin boards sits properly in place to ensure good connections. The connectors are rated for 100 plugin cycles.

For pinout check the schematics.

12.2 Debug connector

This connector is used for Debug In and Debug Out (see Debug chapter). The pinout is described in the table.

Table 12.1. Debug connector pinout

Pin number	Function	Note
1	VTARGET	Target voltage on the debugged application.
2	NC	
3	/TRST	JTAG tap reset
4	GND	
5	TDI	JTAG data in
6	GND	
7	TMS/SWDIO	JTAG TMS or Serial Wire data I/O
8	GND	
9	TCK	JTAG TCK or Serial Wire clock
10	GND	
11	RTCK	JTAG RTCK
12	GND	
13	TDO/SWO	JTAG TDO or Serial Wire Output
14	GND	
15	/RESET	Target MCU reset
16	GND	
17	PD	This pin has a 100k pulldown.
18	Cable detect	This signal must be pulled to ground by the external debugger or application for cable insertion detection.
19	PD	This pin has a 100k pulldown.
20	GND	

13 Debugging

The EFM32-G8XX-DK has an on-board debugger, and it can be used in different ways to debug the EFM, both on and off kit. Below are descriptions on the different modes. Check the configuration chapter to find out how to change the debug setting.

Table 13.1. Debug modes

Mode	Description
Debug MCU	In this mode the built-in debugger is connected to EFM on the BRD3300A.
Debug IN	In this mode the built-in debugger is disconnected, and an external debugger can be connected to debug the EFM on the BRD3300A.
Debug OUT	In this mode the built-in debugger can be used to debug an EFM mounted in your own application.

14 Integrated Development Environments

The Energy Micro software packages contains various examples in source form to use with the Starter Kit. The following IDEs are supported.

14.1 IAR Embedded Workbench for ARM

An evaluation version of IAR Embedded Workbench for ARM is included on a CD in the EFM32-G8XX-DK package. Check the quick start guide for where to find updates, and IAR's own documentation on how to use it. You will find the IAR project file in the

```
iar
```

subfolder of each project

14.2 Rowley Associates - CrossWorks for ARM

See the quick start guide for download details for CrossWorks for ARM. You will find CrossWorks project files in the

```
rowley
```

subfolder of each project.

14.3 CodeSourcery - Sourcery G++

See the quick start guide for download details for Sourcery G++. The

```
codesourcery
```

subfolder contains Makefiles for use with the Sourcery G++ development environment.

14.4 Keil - MDK-ARM

See the quick start guide for download details for evaluation versions of Keil MDK-ARM. The

```
arm
```

subfolder in each project contains project files for MDK-ARM. Please see the MDK-ARM documentation for usage details.

15 energyAware Commander and Upgrades

energyAware Commander is a program that comes with the Gecko DK Installer package. It can perform various kit and EFM32 specific tasks. The program has two modes of operation, either *command line* or *GUI mode*. The command line is not backwards compatible with the earlier "Gecko Commander" shell utility, but support all features of that package.

15.1 GUI Operation

The primary use of *energyAware Commander* is in the GUI mode. This utility gives the ability to program the EFM32, upgrade the kit, lock and unlock devices and more. Some of the features will only work with Energy Micro kits, while other will work with a J-Link debugger connected. Press the "F1" button, or select the "Help->Help" menu item for a full description.

15.2 Command Line operation

Using the command line involves runnig the binary in a command shell, with an appropriate argument, e.g.

```
eACommander.exe --help
```

All command line options will be parsed in the other they are given, so multiple commands can be run.

Table 15.1. energyAware Commander command line options

Option	Description
--version	Prints version and exits
--help (-h)	Prints this help information
--flash image.bin (-f)	Flash binary file to EFM32
--install package.emz (-i)	Install kit firmware
--mcuinfo	Print MCU information
--kitinfo	Print Kit information
--usb 0-3 (-u)	Specify USB address
--speed wanted speed (-s)	Set SWD speed (in kHz)
--verify (-v)	Verify flash after upload
--mode in|out|mcu|off (-m)	Set debug mode
--unlock	Unlock a locked chip
--lock (-l)	Lock a chip
--reset (-r)	Reset the EFM32
--unprotect	Unprotect pages
--protect start stop	Protect pages
--upload file	Upload files to DVK
The mode option argument	Sets the mode of the on-board debugger. Available modes: *mcu* - Debug the EFM32 using the on-board debugger *in* - Debug the EFM32 using an external debugger *out* - Use the DK as an external debugger *off* - Disable the debugger

15.3 Upgrades

Upgrading the kit can be done by using the "Upgrade Kit" script in the start menu. New versions can be downloaded from http://www.energymicro.com/downloads/. The script will use *energyAware Commander* to install the latest available Kit SW package. It is important to upgrade the kit when installing a new SW package, as new *energyAware Commander* functionality might require kit controller software upgrades.

You can also use the *energyAware Commander* for upgrades. Select the "Kit" icon, use the "Browse" button to select the correct file ending in ".emz", and press the "Install package button".

Finally, there is an option at the command line,

```
eACommander.exe --install dvkappl.emz
```

to explictly install new packages by hand (or within a script).

16 Version information

The current version information can be read from the EFM32-G8XX-DK by entering the About page from the main page in the GUI, and then pushing Info.

Table 16.1. Current versions

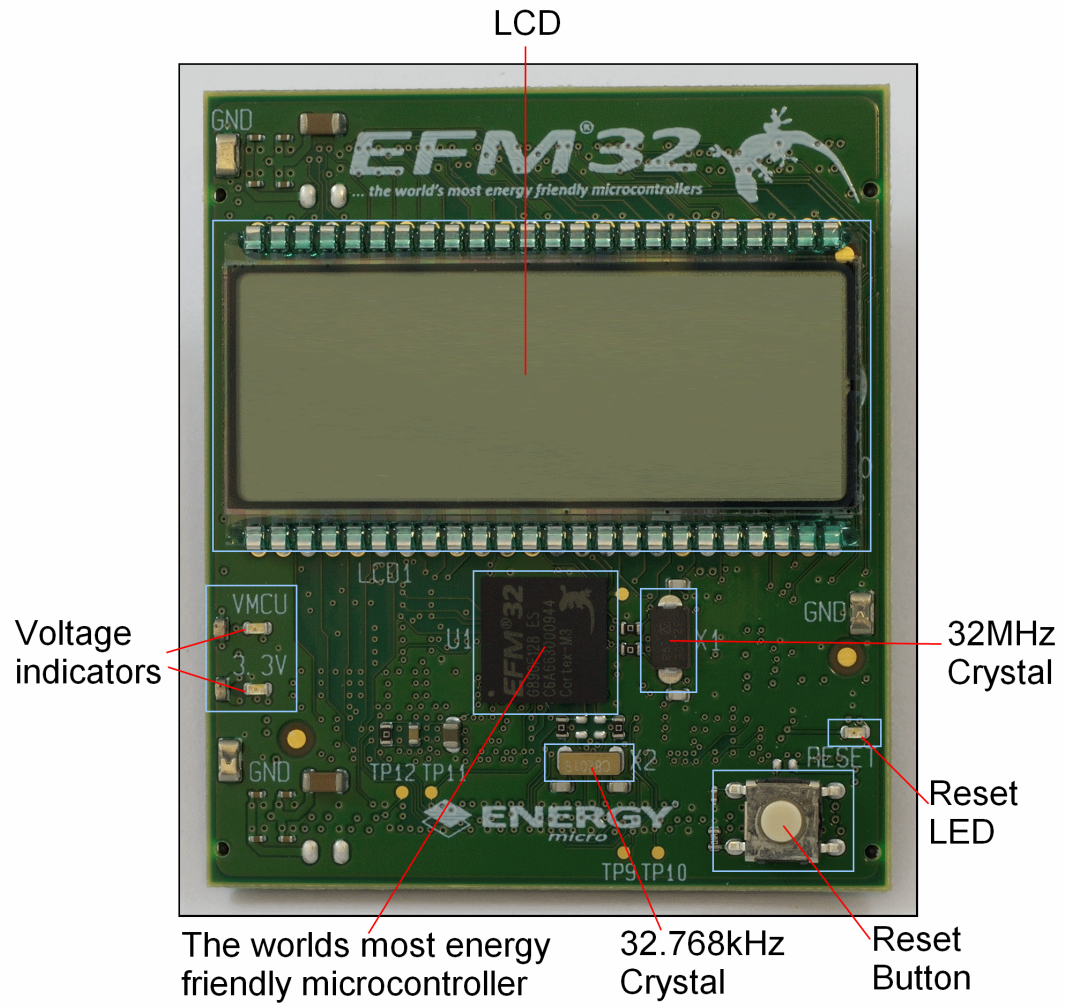
Type	Version	Released
Firmware revision	1v6p0	2010-11-15
FPGA version	1v1	2009-11-13
Motherboard	BRD3200C Rev.A01	2009-11-13

17 MCU board

The EFM32-G8XX-DK is equipped with the BRD3300A. The main features are listed here, but for a complete overview, check the BRD3300A user manual.

Features:

- The worlds most energy friendly microcontroller
- Compatible with the Advanced Energy Monitoring (AEM) system of the EFM32 Gecko Development Kit
- Leds indicating power and reset
- 32 MHz crystal
- 32.768 kHz crystal
- Reset button and ground-hooks for easy debugging
- Energy Micro LCD



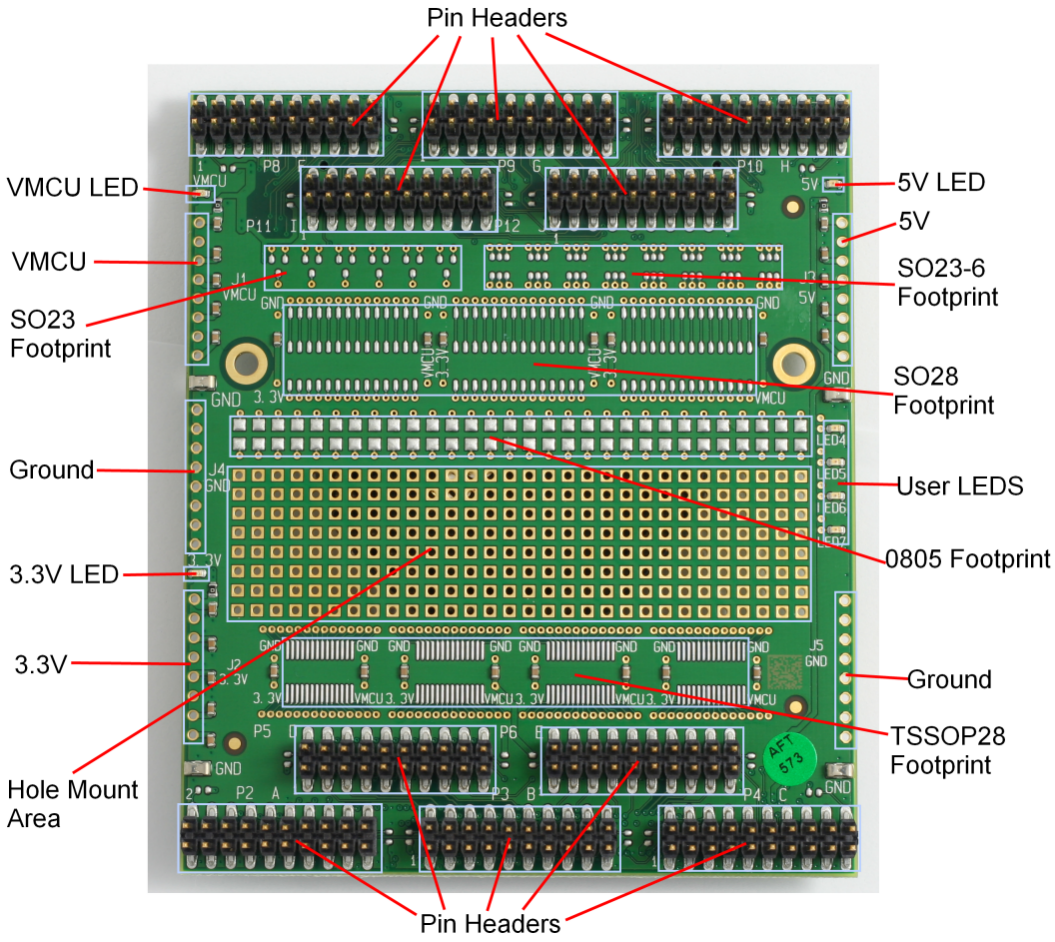
18 Prototyping Board

The EFM32-G8XX-DK is equipped with a prototyping board. The main features are listed here, but for a complete overview, check the prototyping board user manual.

18.1 Overview

Features:

- Ready-to-use prototyping area for hole-mount, TSSOP, SO, SOT23-6, SOT23 and 0805 SMD components.
- VMCU power domain tracked by the Advanced Energy Monitor (AEM).
- 3.3V and 5V power domains available.
- All EFM32 IO lines directly accessible through pin headers.
- User LEDS ready for use.
- LEDS indicating power.



19 Errata

19.1 DVK hardware errata

19.1.1 Port E0 on EFM32 MCU board disconnected

Kit Revisions

Rev A

Description

Revision A kits are shipped with engineering samples of the EFM32. On MCU boards containing these engineering samples, port E0 on the EFM32 is disconnected. The MCU boards in question are shipped with a sticker indicating that port E0 is disconnected.

19.2 DVK firmware errata

19.2.1 VMCU regulator

Firmware Revisions

1.0.1

Description

Firmware version 1.0.1 contains an I2C driver that on some occasions hangs. When this happens, VMCU freezes and can no longer be changed. The same issue may also result in wrong VMCU setting after startup. A power cycle of the DVK fixes the problem. This errata is fixed in firmware version 1.1.1 or newer.

19.2.2 Storing user configuration

Firmware Revisions

1.0.1

Description

Firmware version 1.0.1 contains an I2C driver that on some occasions hangs. When this happens, user configuration may no longer be stored. A power cycle of the DVK fixes the problem. This errata is fixed in firmware version 1.1.1 or newer.

19.2.3 Debug mode setting

Firmware Revisions

1.0.1

Description

When changing debug mode using the GUI, the actual configuration of the debug interface is not always the same as the debug mode set by the GUI. An upgrade to firmware version 1.1.1 or newer fixes this errata.

19.2.4 Serial Wire Output

Firmware Revisions

1.0.1

Description

Enabling Serial Wire Output (SWO) would cause onboard firmware to fail. An upgrade to firmware version 1.1.1 or newer fixes this errata.

EFM32 Development Kit

Board Function	Page
Analog Interfaces	2
Sensors, SPI bus and I2C bus	3
User Interface	4
Display Interface	5
Flash and SRAM	6
Board Control - Control MCU	7
Board Control - Buses	8
Board Control - Misc	9
Board Control - JTAG	10
Control MCU	11
Control MCU - BC Interface	12
Debug Interface	13
Board Control - Level shift	14
EXP32 Assignments #1	15
EXP32 Assignments #2	16
EFM32 Board Connectors	17
EXP32 Board Connectors	18
Main Power Regulators	19
EFM Power Regulators and AEM	20
Power monitoring	21

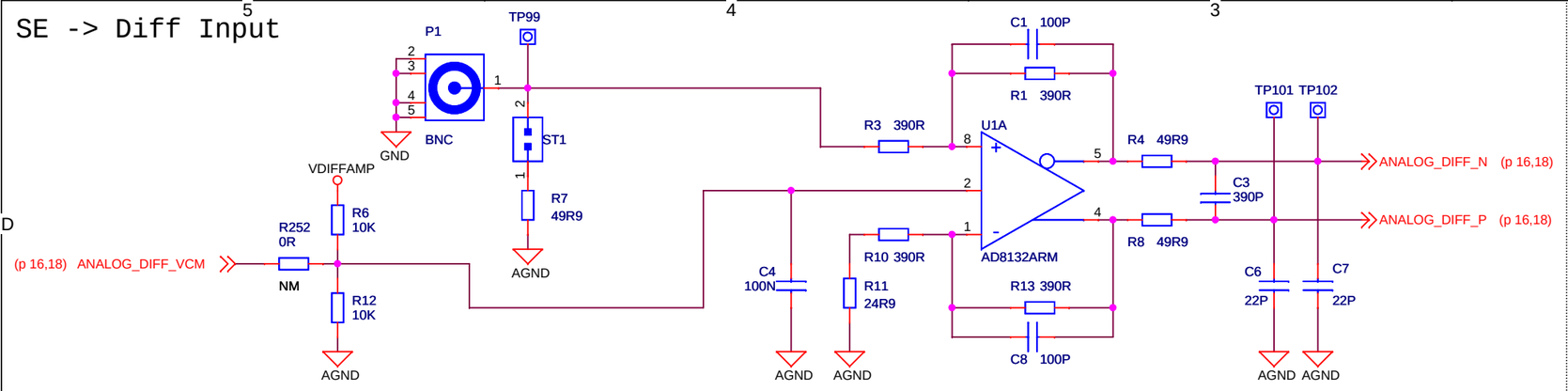
Revision History

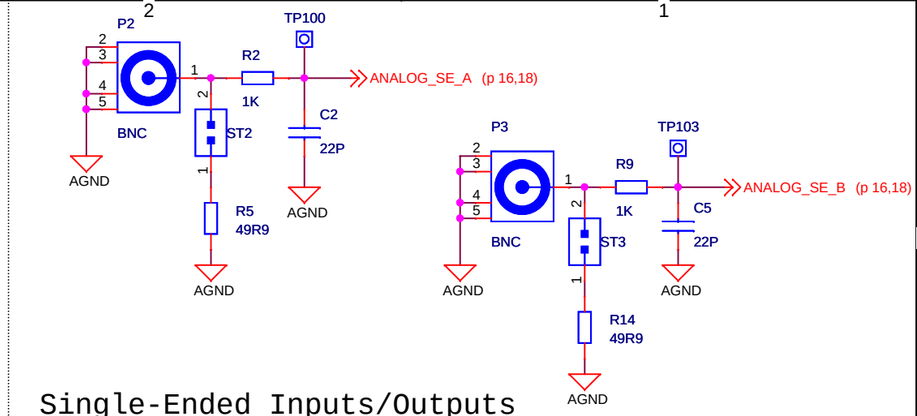
Rev.	Description
A00	Initial release
A01	Minor update.

<Schematic Path>
TOP

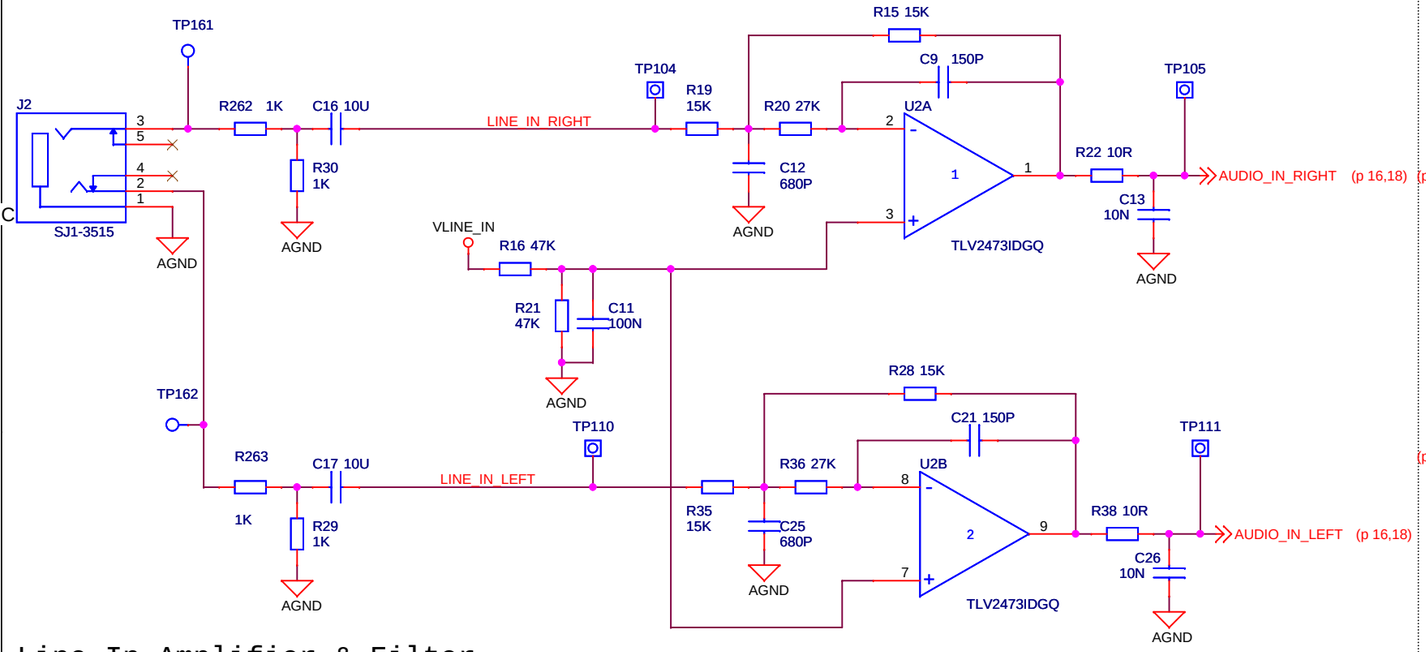
		Schematic Title	**EFM32 Development Kit - Mainboard**
		Page Title	**Title Page**
Designed: JNO	Approved: JNO	Document number **BRD3200C**	Revision A01
Size A3	BOM Doc No: <Cage Code>		
Design Created Date: Wednesday, December 03, 2008		Sheet Created Date Saturday, March 21, 2009	Sheet Modified Date Monday, June 28, 2010 / Sheet 1 of 21

SE -> Diff Input

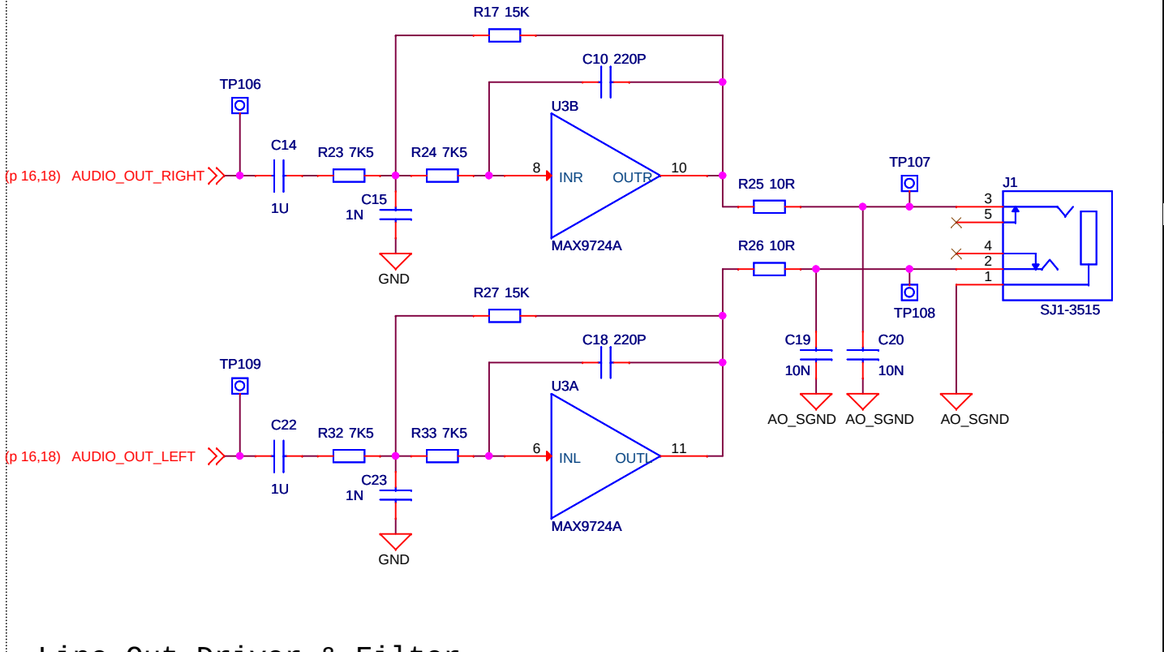




Single-Ended Inputs/Outputs

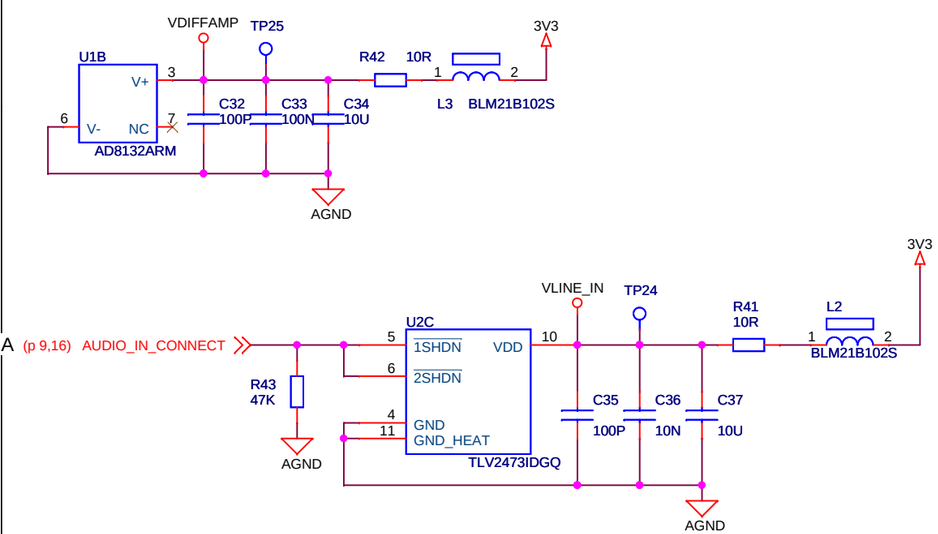


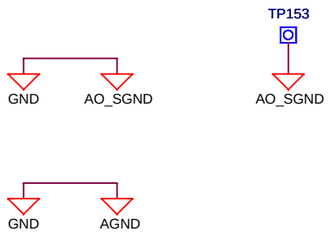
Line In Amplifier & Filter

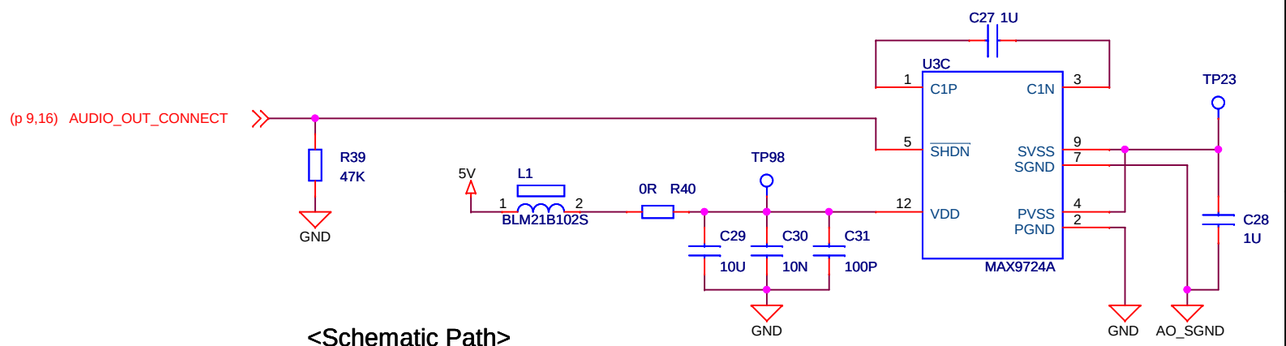


Line Out Driver & Filter

Power Supply and Decoupling

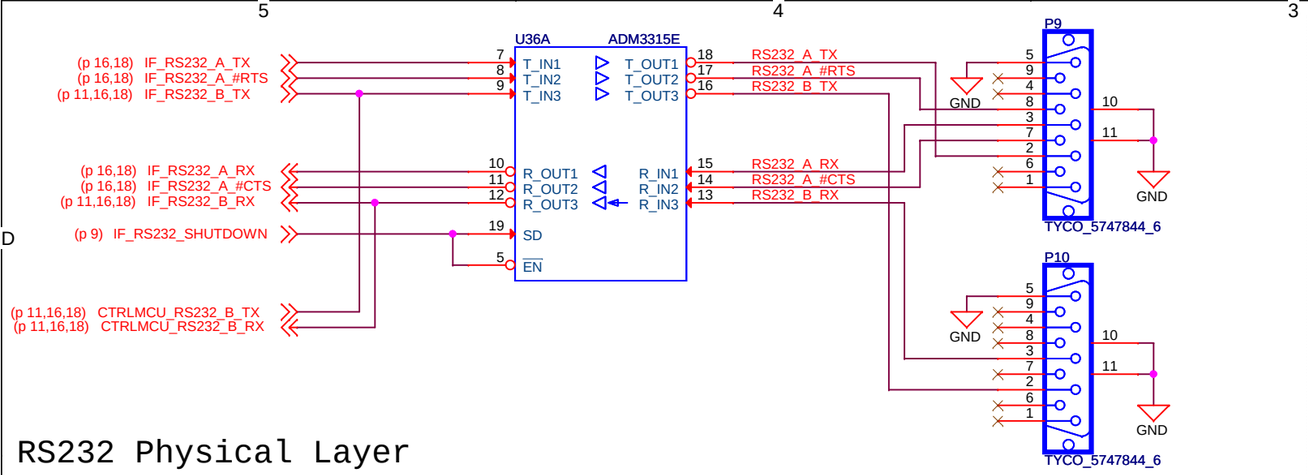


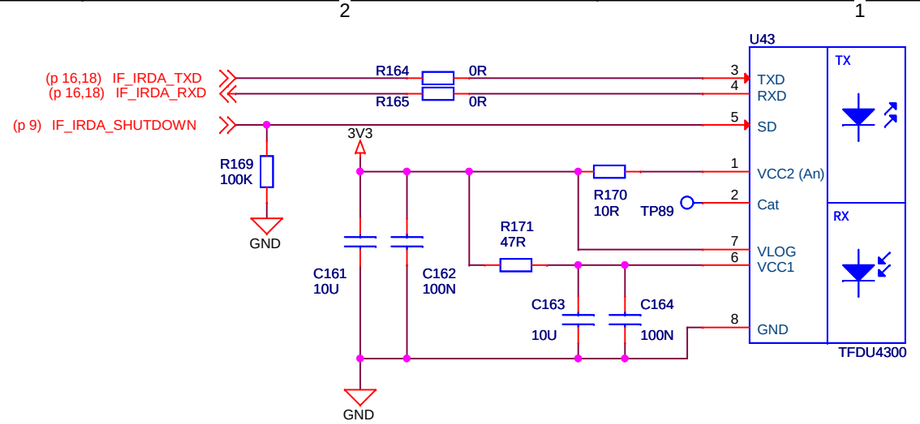


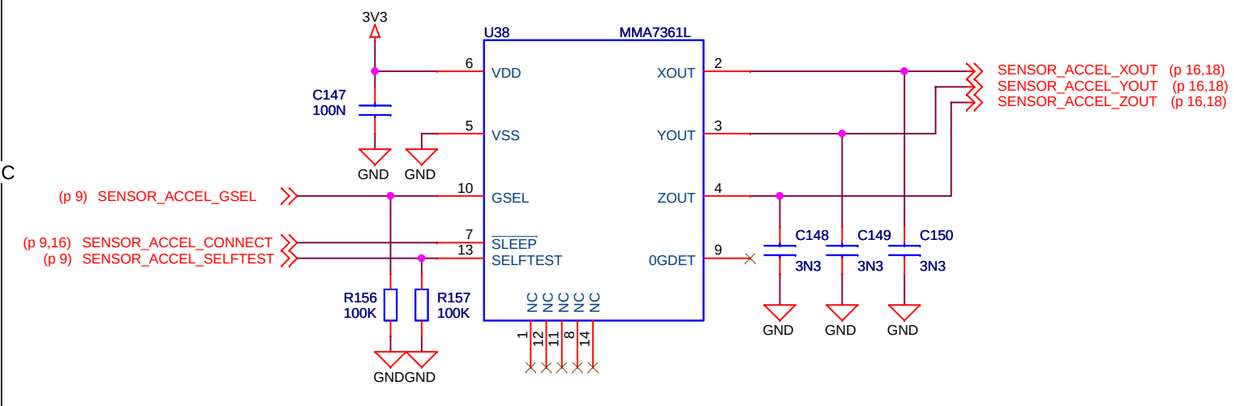


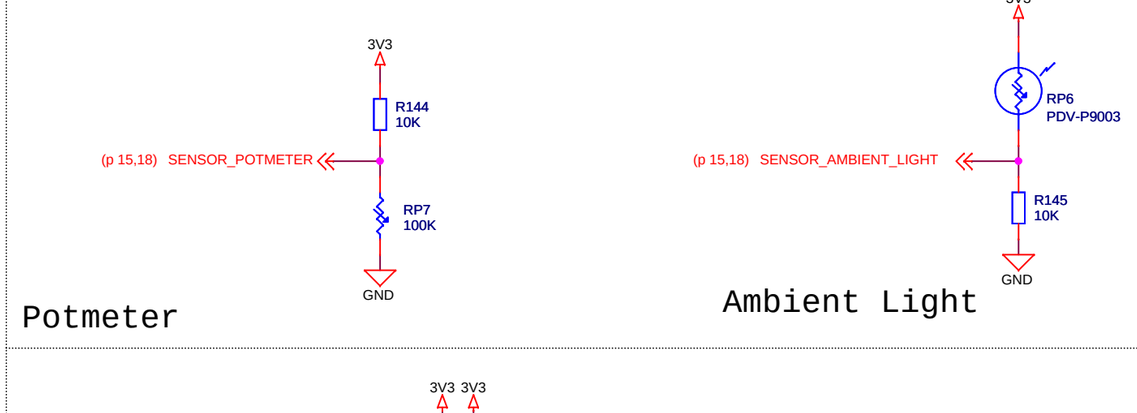
<Schematic Path>
TOP

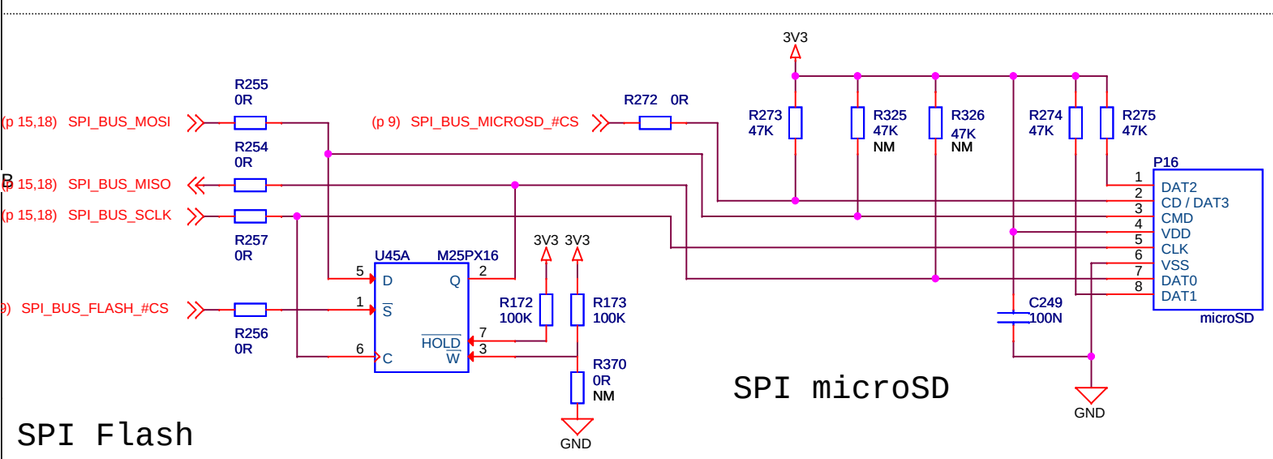
Schematic Title	**EFM32 Development Kit - Mainboard**
Page Title	**Analog Interfaces**
Designed: JNO / Approved: JNO	Document number **BRD3200C** / Revision A01
Size A3 / BOM Doc No: <Cage Code>	
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet 2 of 21

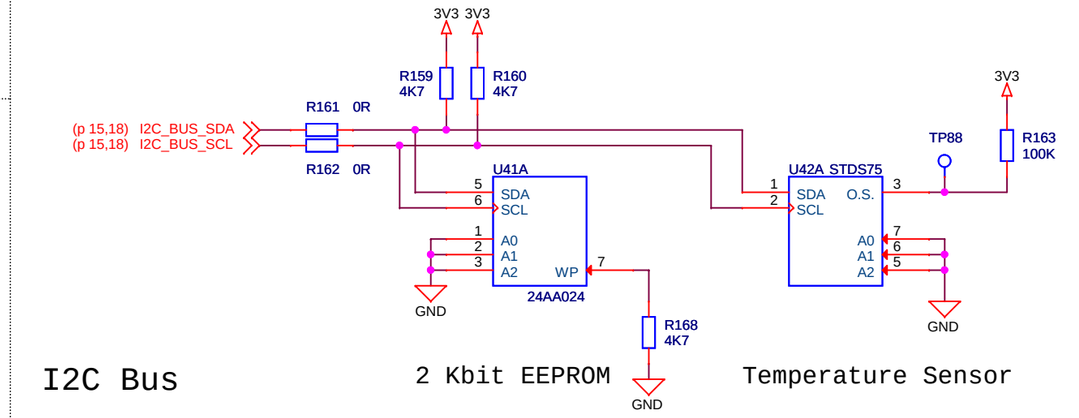




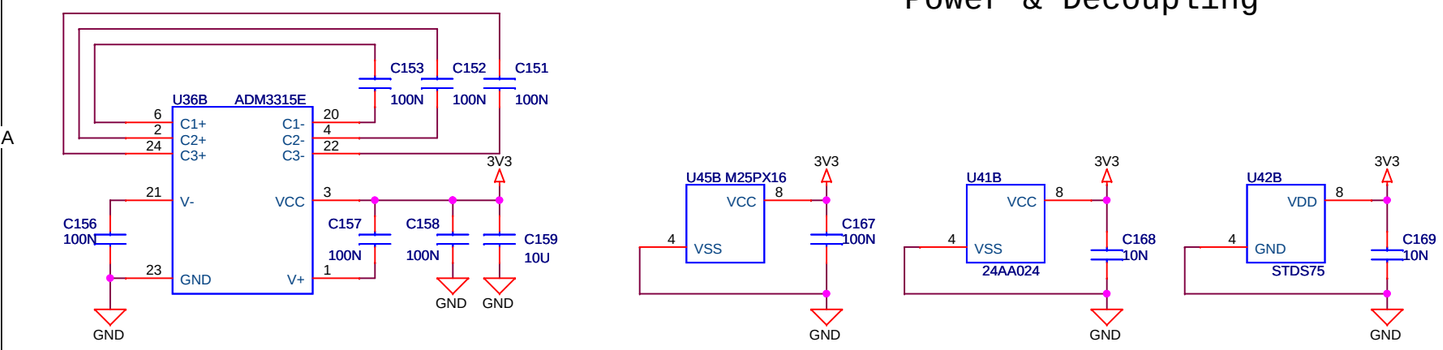








Power & Decoupling

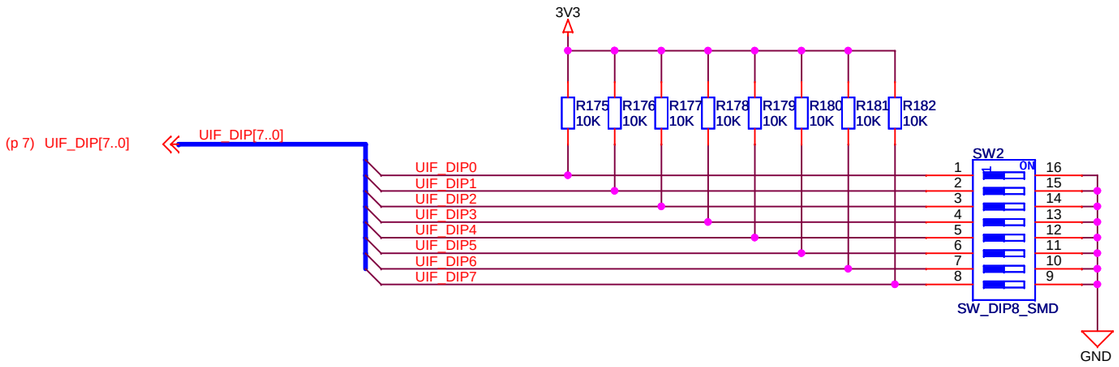


<Schematic Path>
TOP

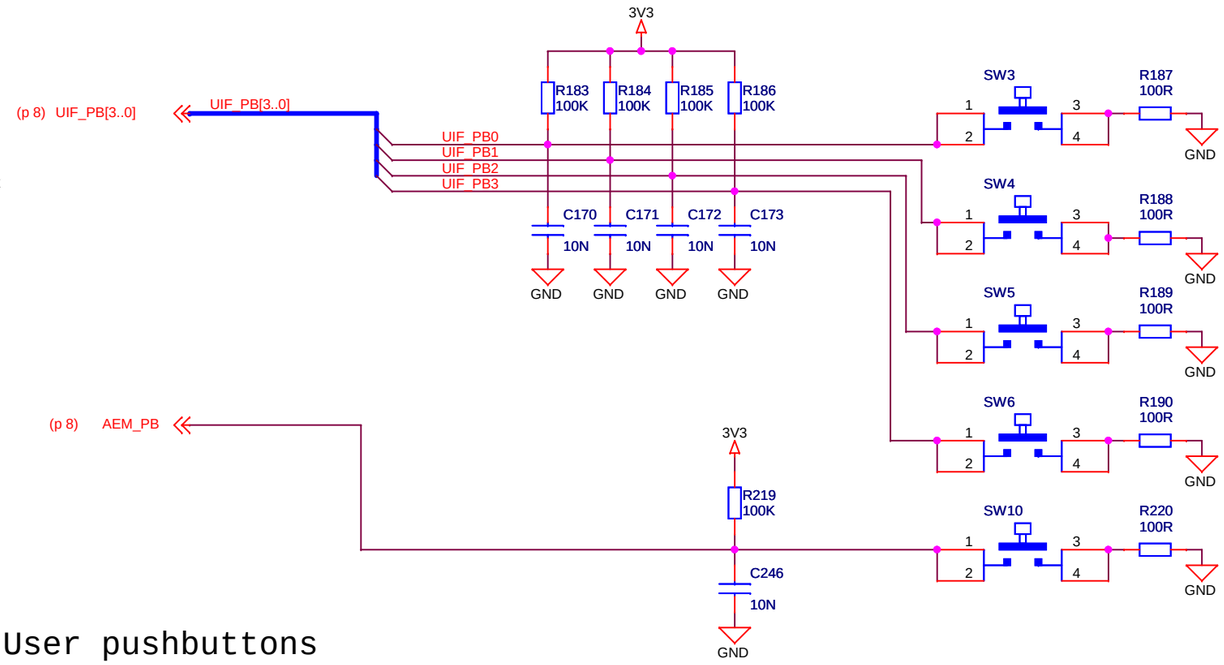
Schematic Title
EFM32 Development Kit - Mainboard

Page Title
Sensors, SPI bus, I2C bus and IO

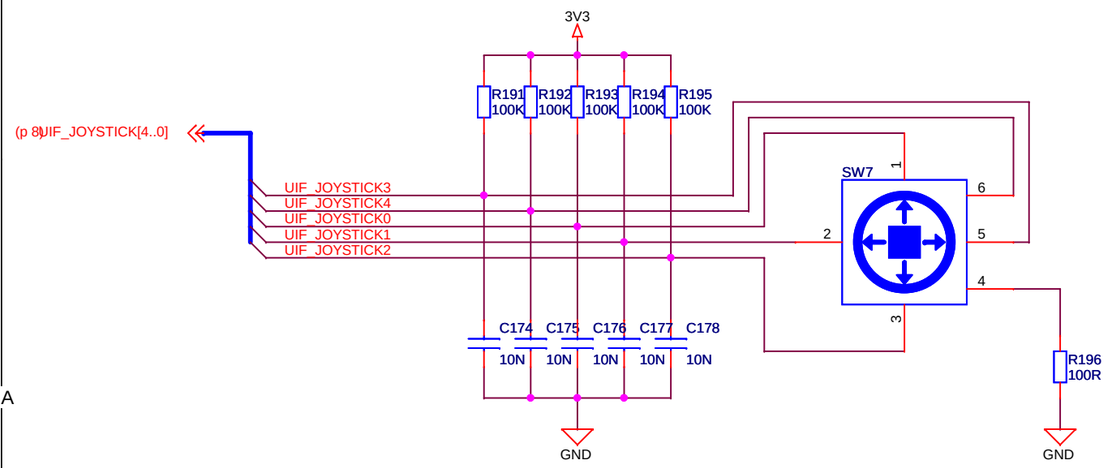
Designed: JNO	Approved: JNO		
Size: A3	BOM Doc No: <Cage Code>	Document number: BRD3200C	Revision: A01
Design Created Date: Wednesday, December 03, 2008		Sheet Created Date: Saturday, March 21, 2009	Sheet Modified Date: Monday, June 28, 2010
			Sheet: 3 of 21



User switches

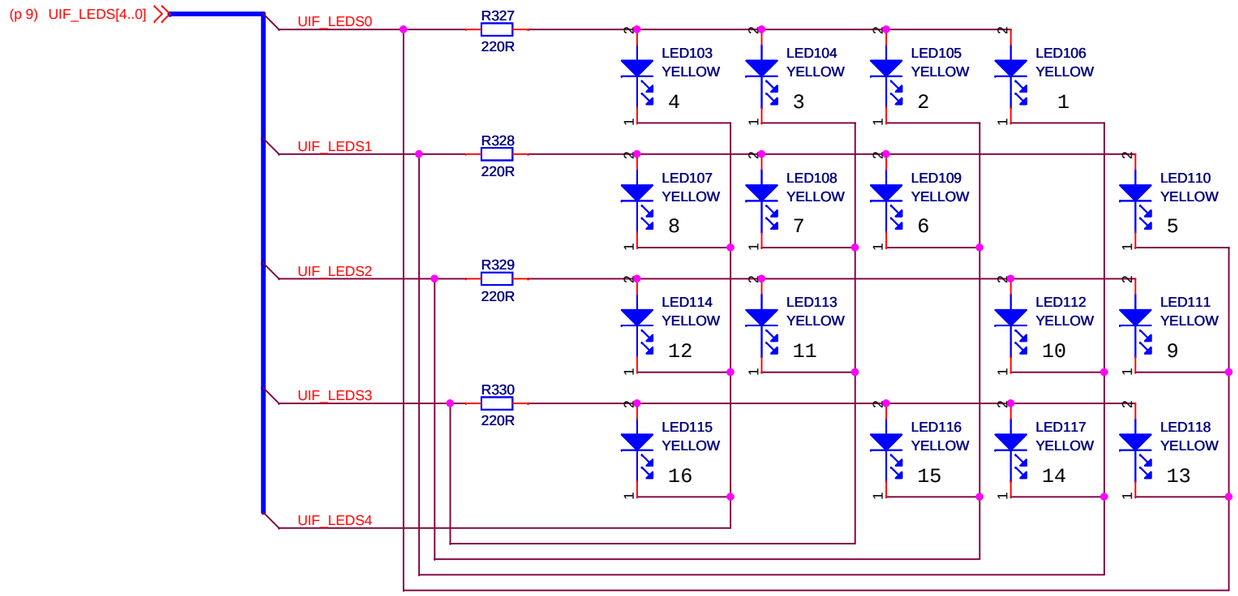


User pushbuttons



Joystick

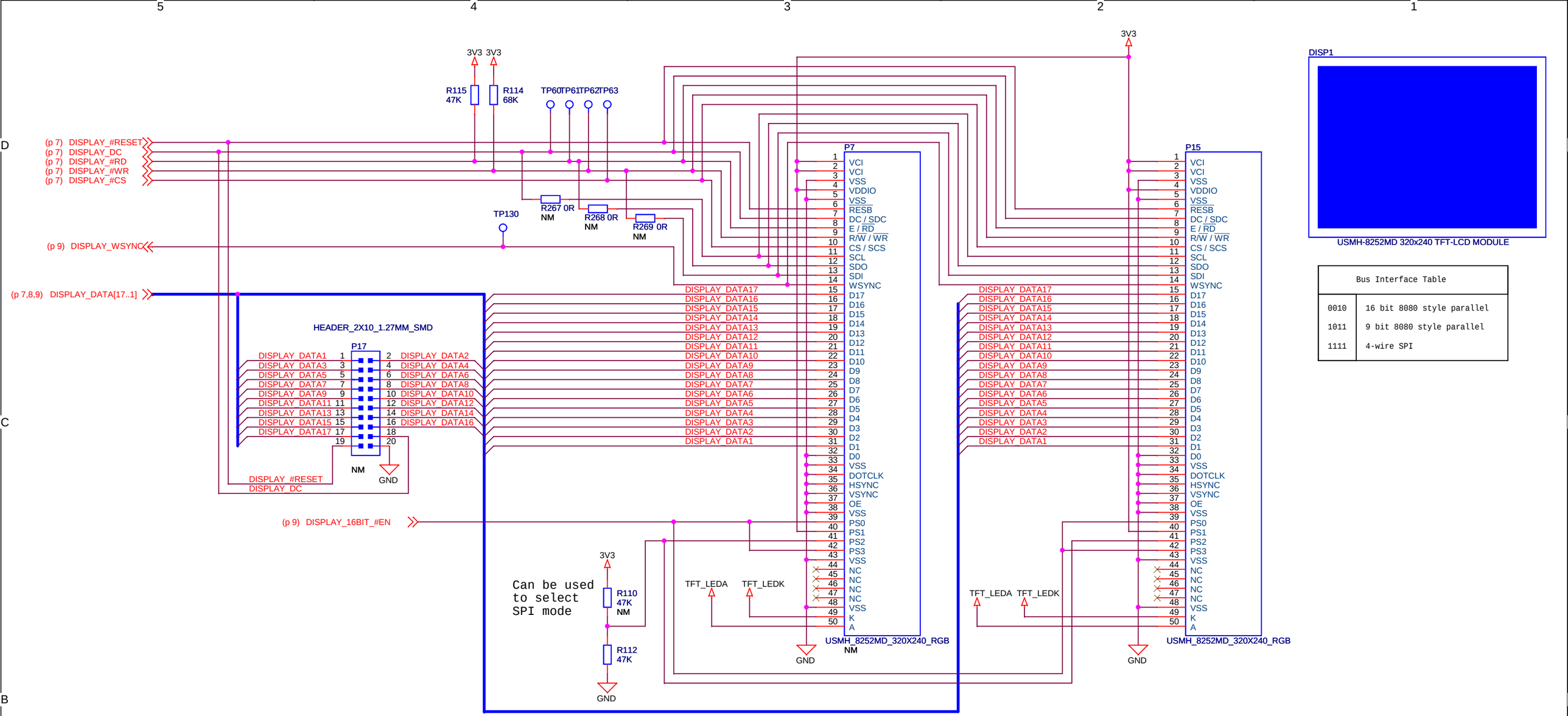
User LEDs

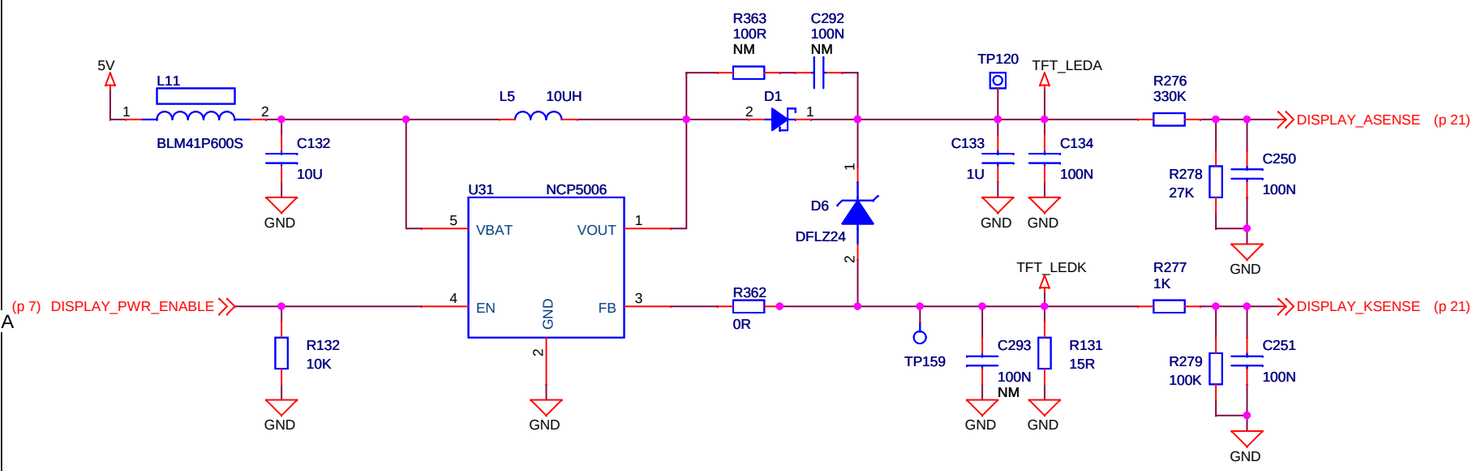


Inputs	LED
zzz01	1
zz0z1	2
z0zz1	3
0zzz1	4
zzz10	5
zz01z	6
z0z1z	7
0zz1z	8
zz1z0	9
zz10z	10
z01zz	11
0z1zz	12
z1zz0	13
z1z0z	14
z10zz	15
01zzz	16

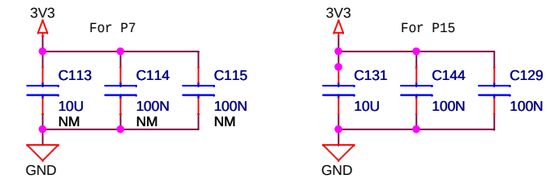
<Schematic Path>
TOP

	Schematic Title: **EFM32 Development Kit - Mainboard**
	Page Title: **User Interfaces**
Designed: JNO / Approved: JNO	
Size: A3 / BOM Doc No: <Cage Code>	Document number: **BRD3200C** / Revision: A01
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet: 4 of 21





TFT-LCD power regulator



<Schematic Path>
TOP

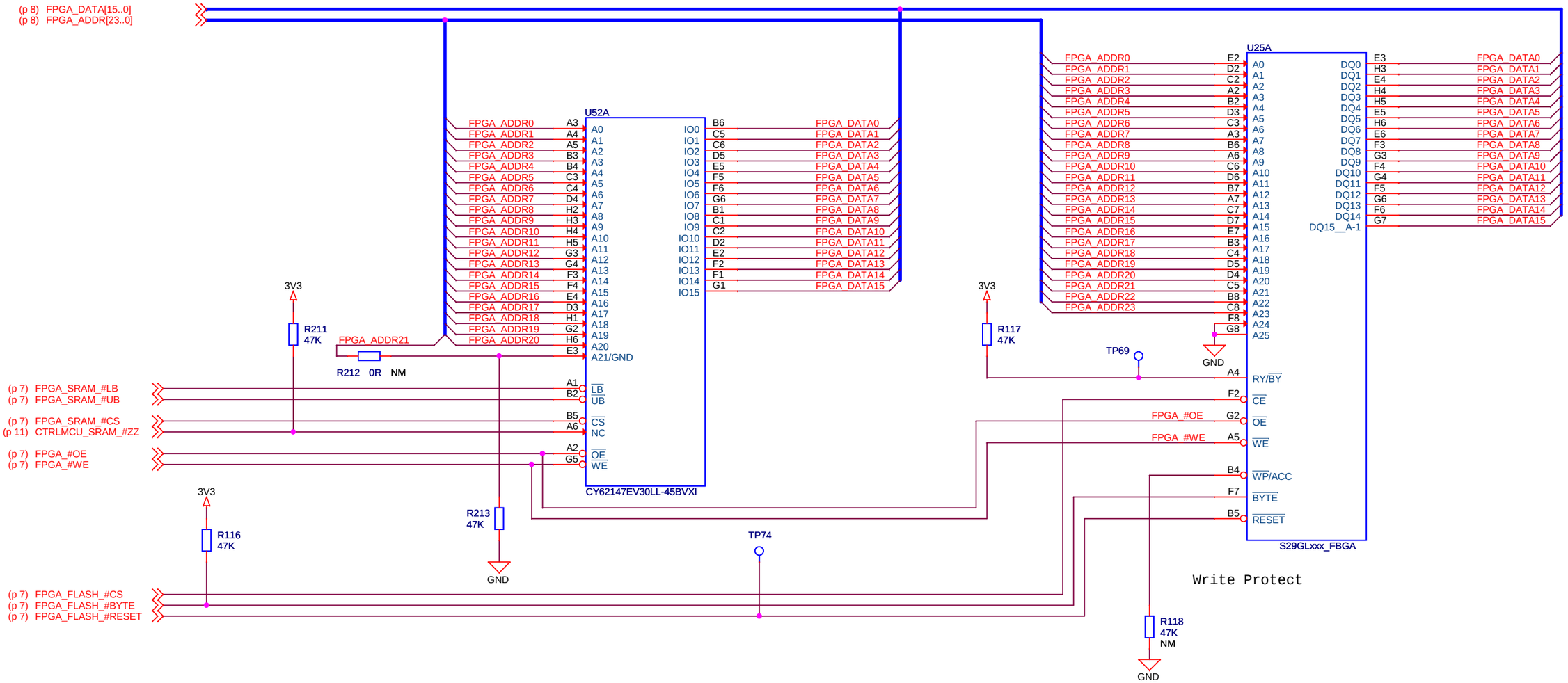
Designed: JNO	Approved: JNO
Size: A3	BOM Doc No: <Cage Code>
Design Created Date: Wednesday, December 03, 2008	

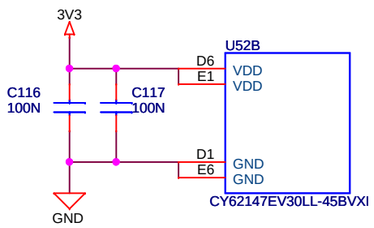
Schematic Title
EFM32 Development Kit - Mainboard

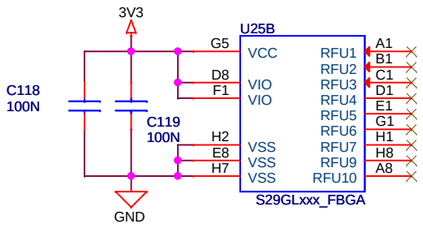
Page Title
Display Interface

Document number	Revision
BRD3200C	A01

Sheet Created Date	Sheet Modified Date	Sheet
Saturday, March 21, 2009	Monday, June 28, 2010	5 of 21

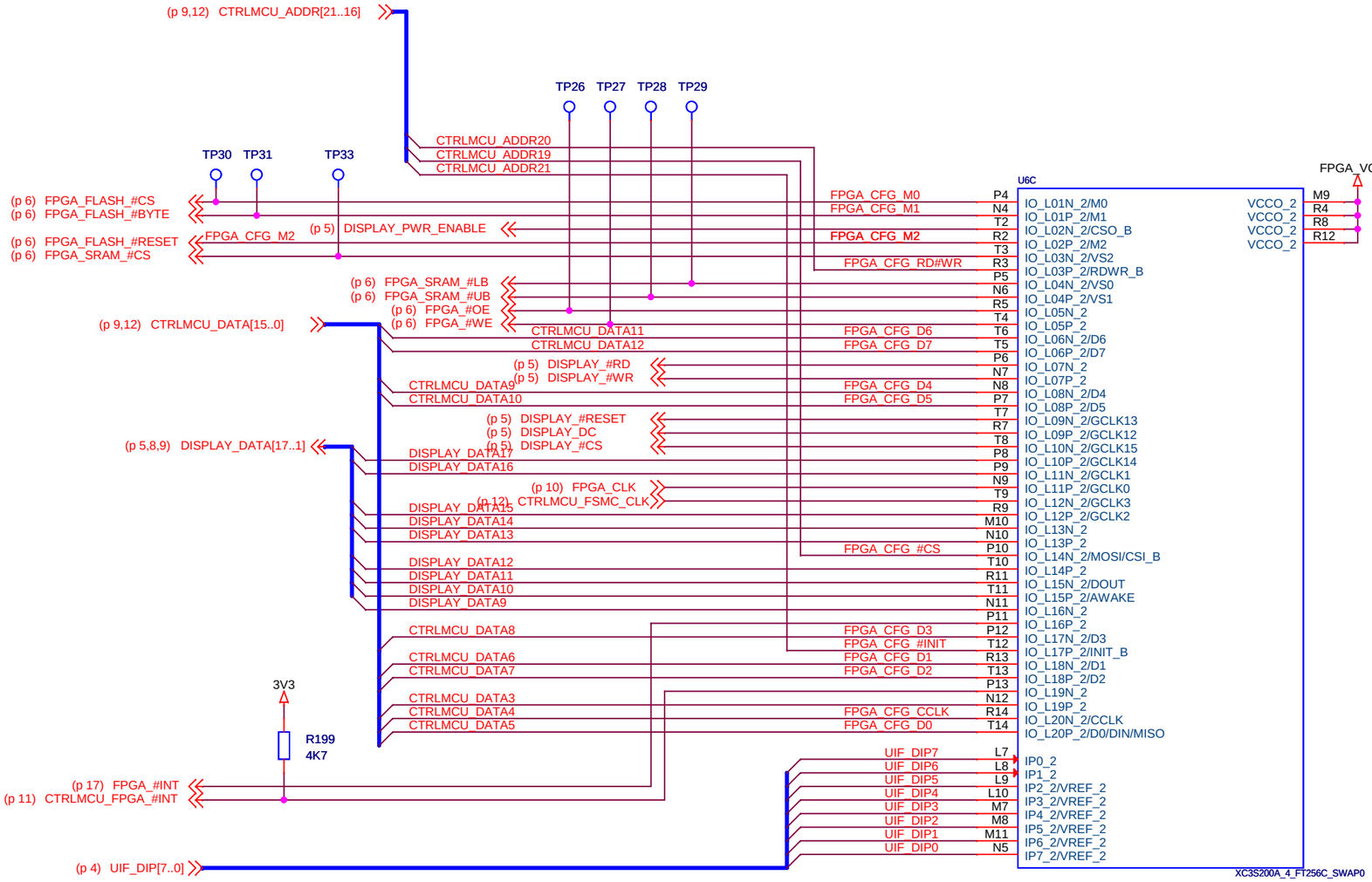




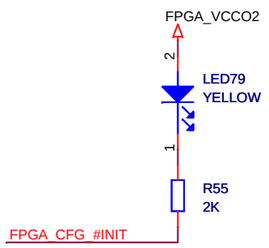


<Schematic Path>
TOP

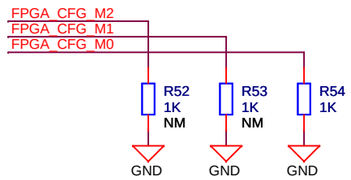
	Schematic Title: **EFM32 Development Kit - Mainboard**
	Page Title: **Parallel bus memory**
Designed: JNO / Approved: JNO	Document number: **BRD3200C** / Revision: A01
Size: A3 / BOM Doc No: <Cage Code>	
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet: 6 of 21



FPGA Init LED

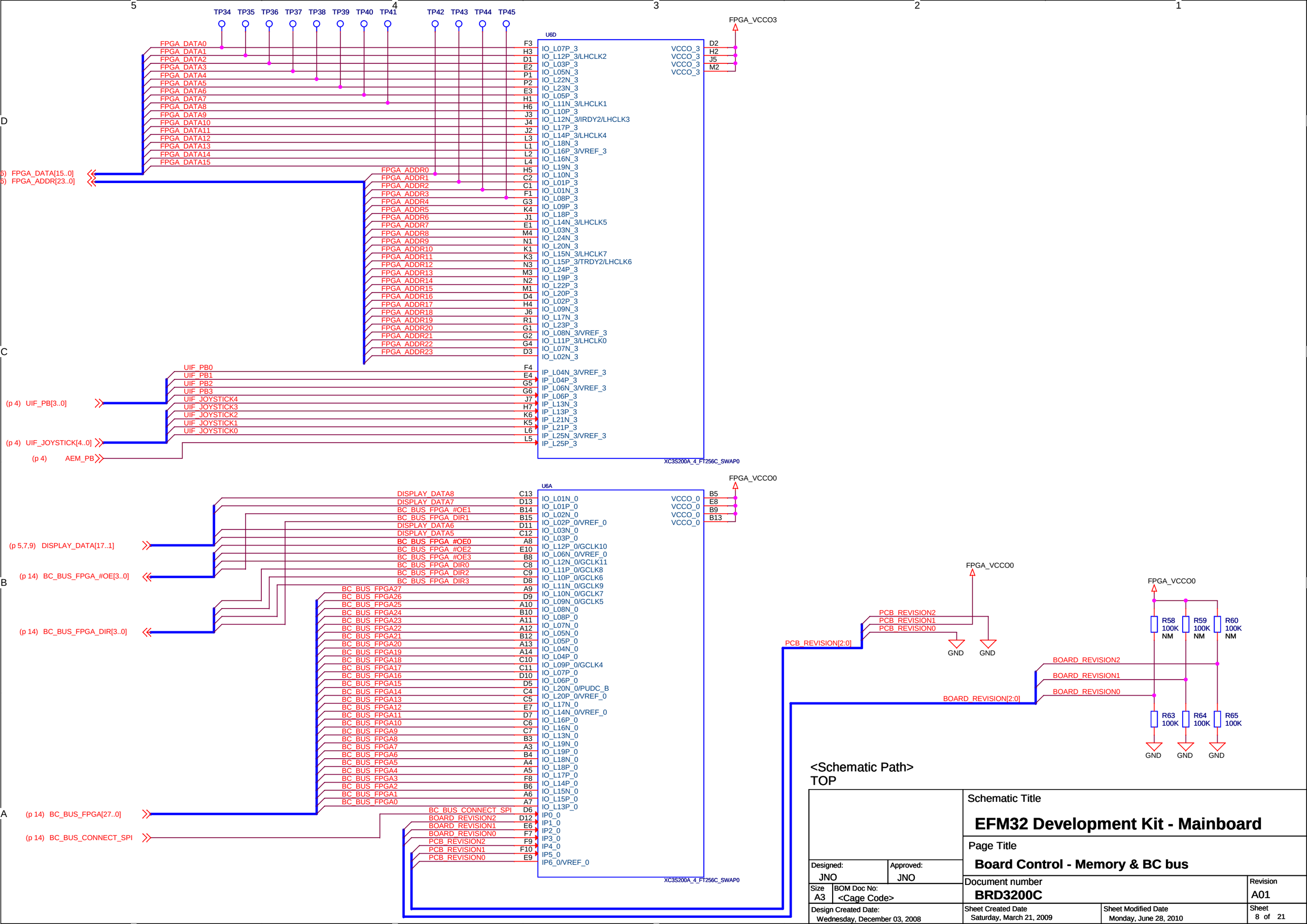


FPGA Configuration Mode



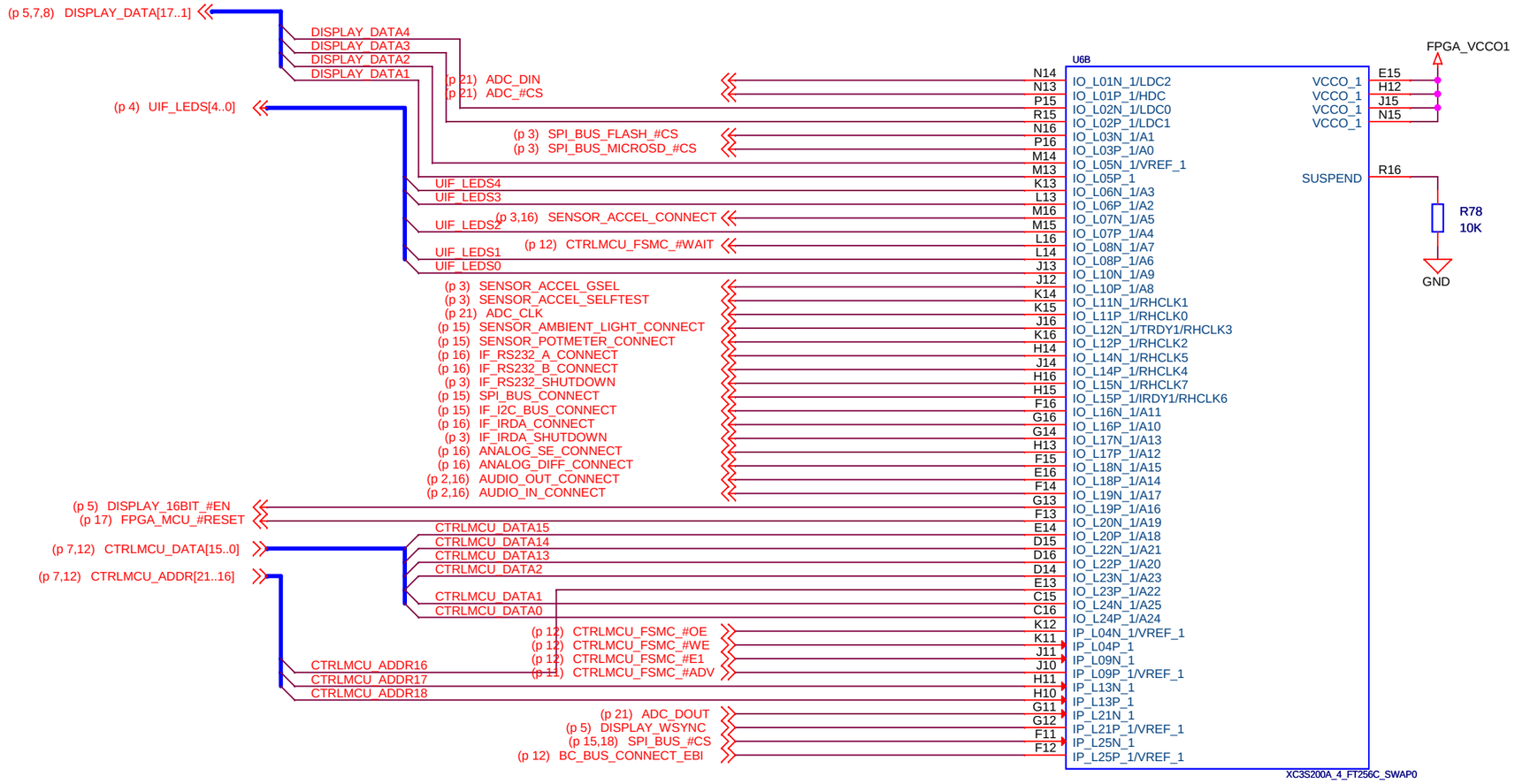
<Schematic Path>
TOP

	Schematic Title: **EFM32 Development Kit - Mainboard**
	Page Title: **Board Control - Control MCU Interface**
Designed: JNO / Approved: JNO	Document number: **BRD3200C** / Revision: A01
Size: A3 / BOM Doc No: <Cage Code>	
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet: 7 of 21



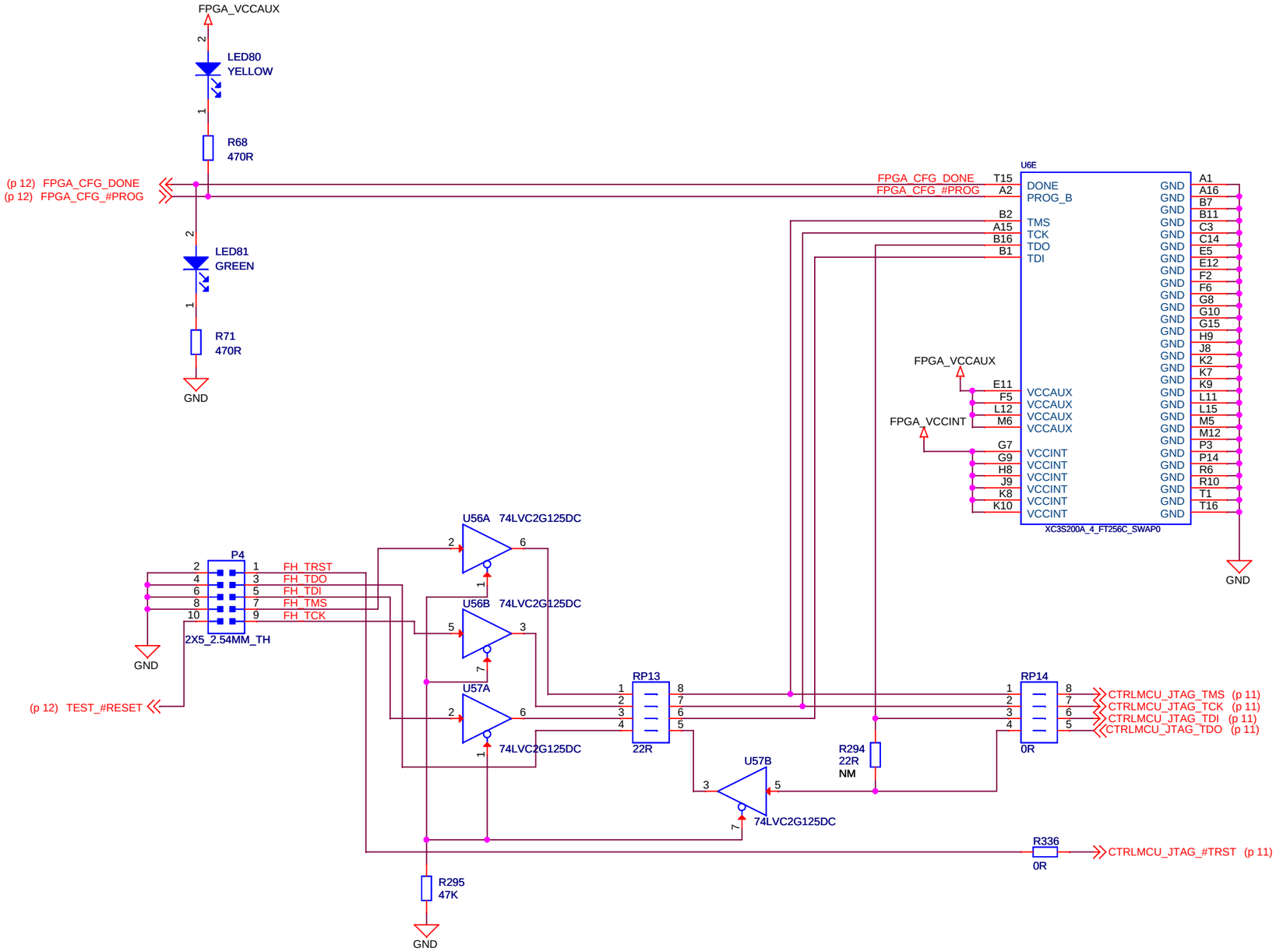
<Schematic Path>
TOP

Schematic Title	**EFM32 Development Kit - Mainboard**
Page Title	**Board Control - Memory & BC bus**
Designed: JNO / Approved: JNO	
Size: A3 / BOM Doc No: <Cage Code>	Document number: **BRD3200C** / Revision: A01
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet: 8 of 21

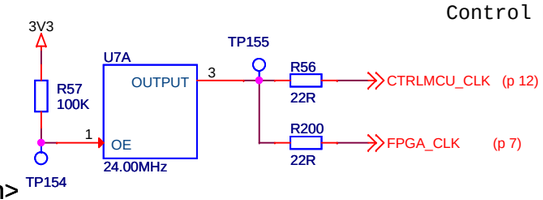


<Schematic Path>
TOP

	Schematic Title: **EFM32 Development Kit - Mainboard**
	Page Title: **Board Control - Misc**
Designed: JNO / Approved: JNO	Document number: **BRD3200C** / Revision: A01
Size: A3 / BOM Doc No: <Cage Code>	
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet: 9 of 21



Control MCU and FPGA clock



<Schematic Path>
TOP

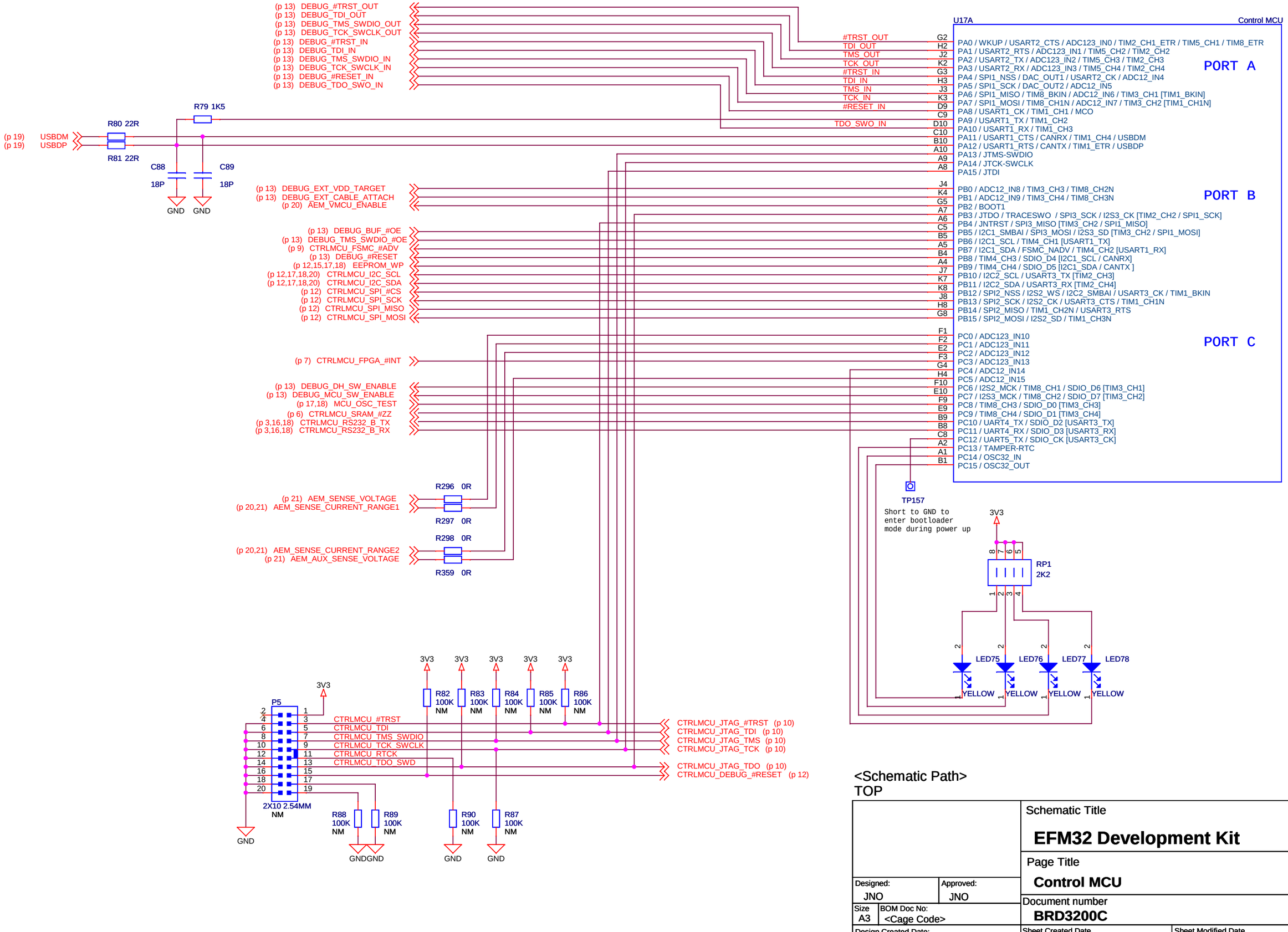
Designed: JNO	Approved: JNO
Size: A3	BOM Doc No: <Cage Code>
Design Created Date: Wednesday, December 03, 2008	

Schematic Title
EFM32 Development Kit - Mainboard

Page Title
Board Control - FPGA Power / Debug

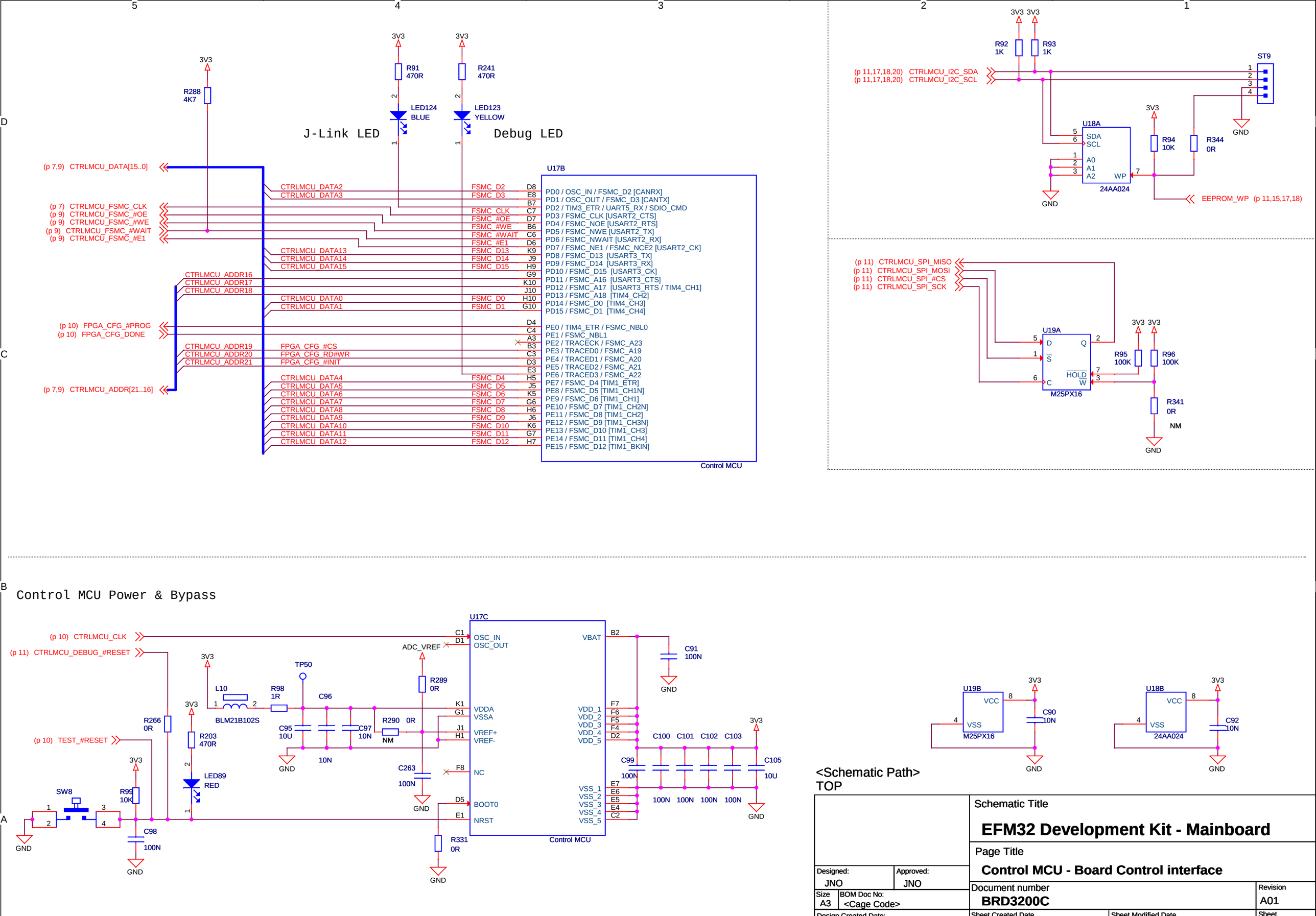
Document number	Revision
BRD3200C	A01

Sheet Created Date	Sheet Modified Date	Sheet
Saturday, March 21, 2009	Monday, June 28, 2010	10 of 21



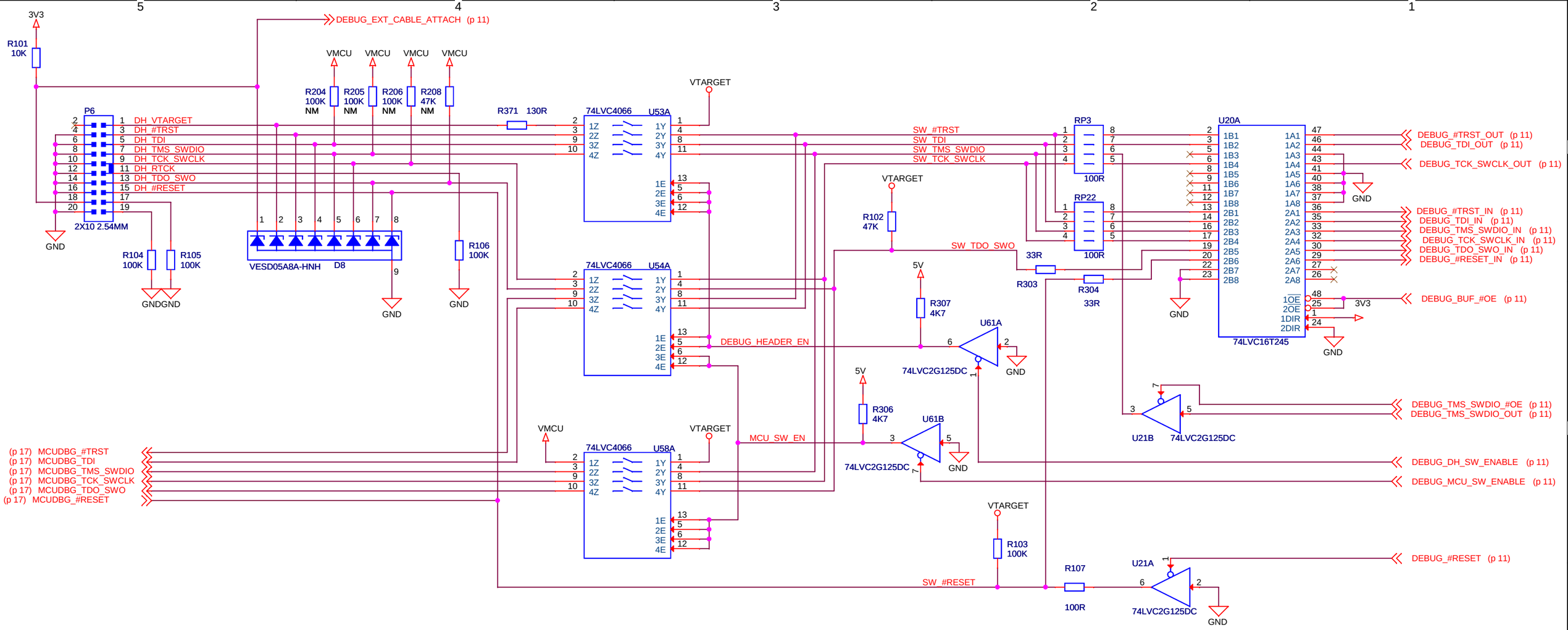
<Schematic Path>
TOP

Schematic Title	**EFM32 Development Kit**
Page Title	**Control MCU**
Designed: JNO / Approved: JNO	
Size: A3 / BOM Doc No: <Cage Code>	Document number: **BRD3200C** / Revision: A01
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet 11 of 21

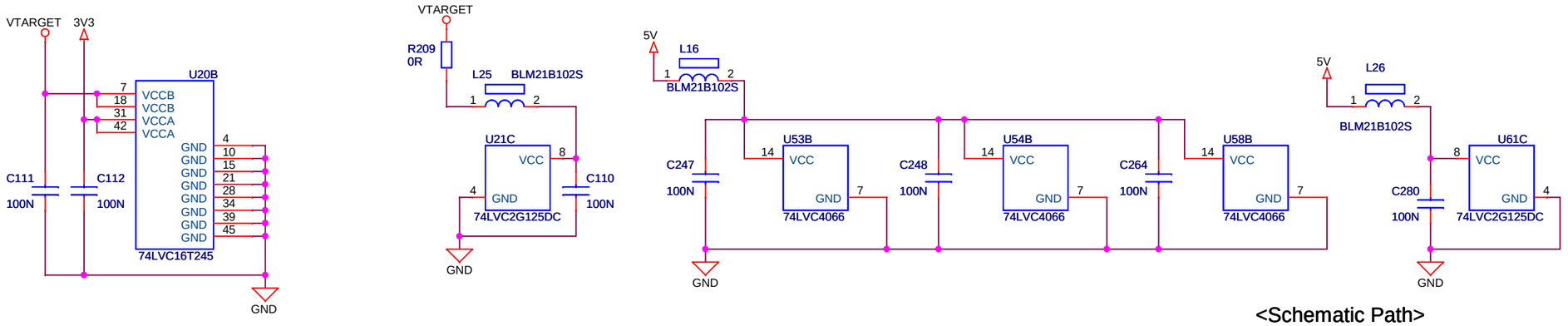


Control MCU Power & Bypass

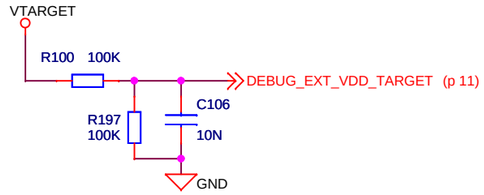
Schematic Title	**EFM32 Development Kit - Mainboard**
Page Title	**Control MCU - Board Control interface**
Designed: JNO / Approved: JNO	
Size: A3 / BOM Doc No: <Cage Code>	Document number: **BRD3200C** / Revision: A01
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet: 12 of 21



Power & Decoupling



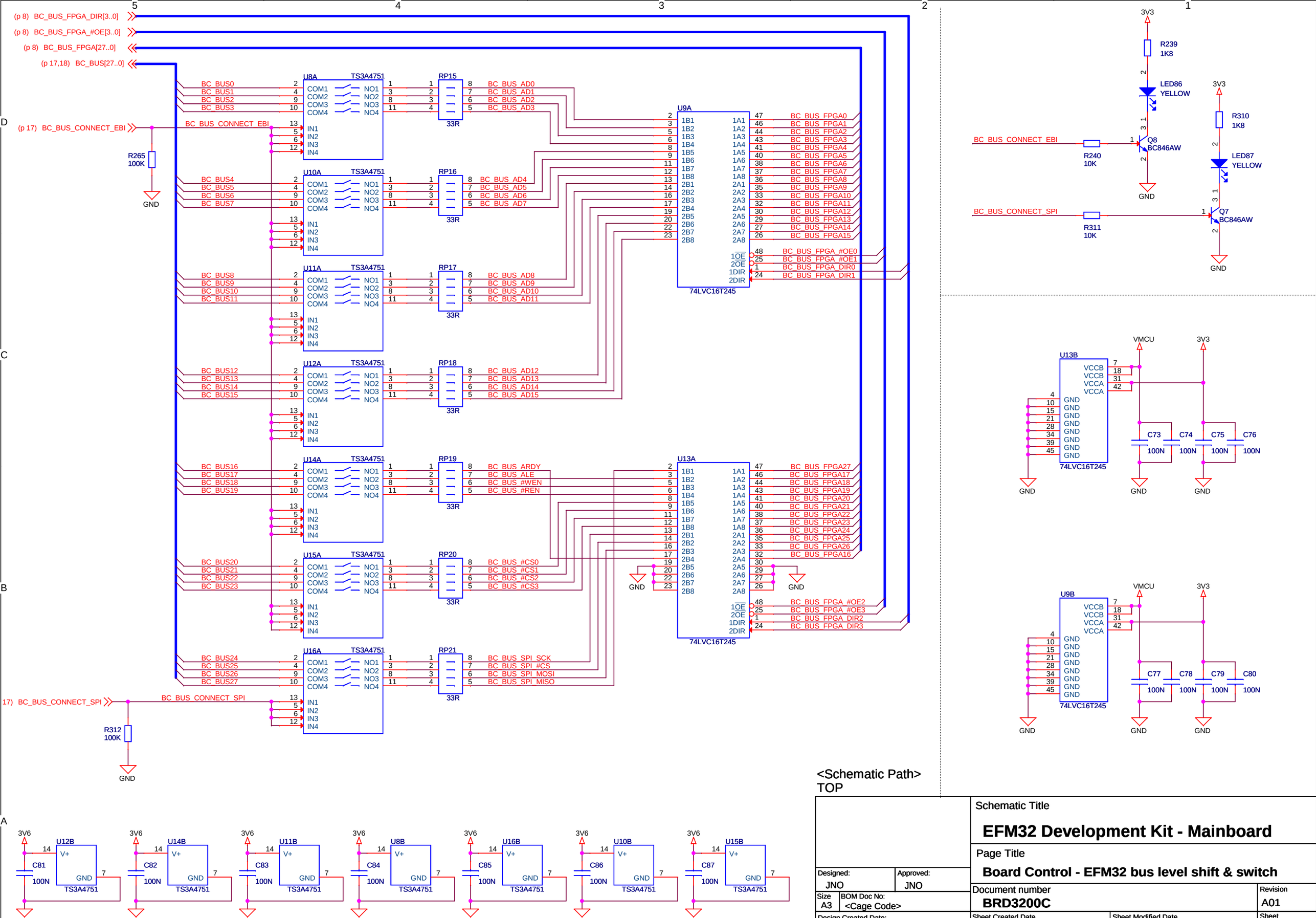
Ext. Debug Voltage Measurement



<Schematic Path>
TOP

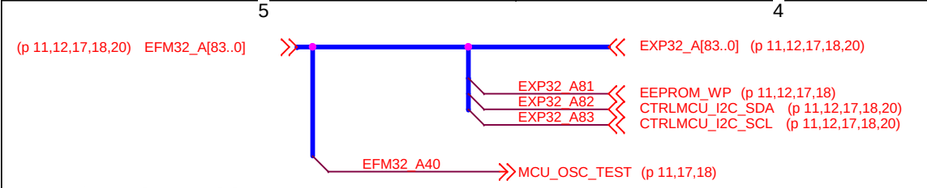
Mode	DEBUG_MCU_SW_ENABLE	DEBUG_DH_SW_ENABLE	DEBUG_BUF_#OE	DH_VTARGET	VTARGET
Debug Out	0	1	0	External voltage	External voltage
MCU Debug	1	0	0	Disconnected	VMCU
Debug In	1	1	1	VMCU	VMCU

Schematic Title	EFM32 Development Kit - Mainboard
Page Title	Debug Interface
Designed: JNO	Approved: JNO
Size: A3	BOM Doc No: <Cage Code>
Document number: BRD3200C	Revision: A01
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009
Sheet Modified Date: Monday, June 28, 2010	Sheet: 13 of 21

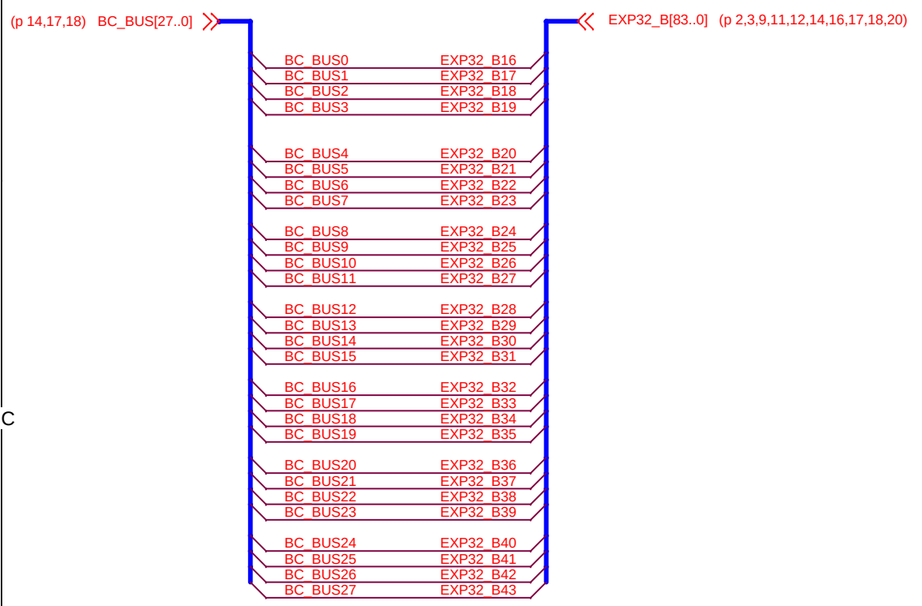


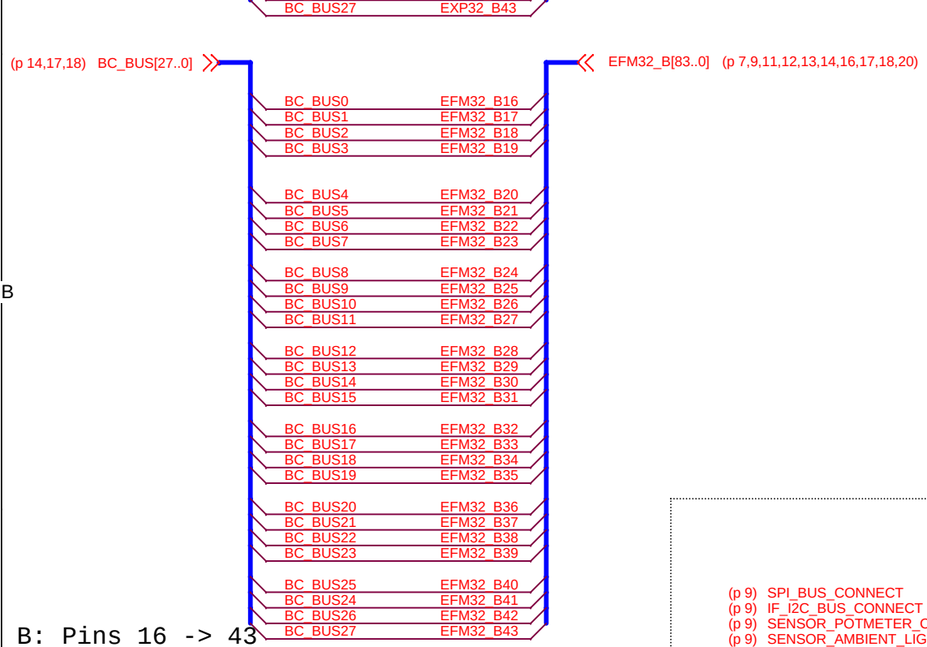
<Schematic Path>
TOP

Schematic Title	**EFM32 Development Kit - Mainboard**
Page Title	**Board Control - EFM32 bus level shift & switch**
Designed: JNO / Approved: JNO	Document number: **BRD3200C** / Revision: A01
Size: A3 / BOM Doc No: <Cage Code>	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010
Design Created Date: Wednesday, December 03, 2008	Sheet: 14 of 21

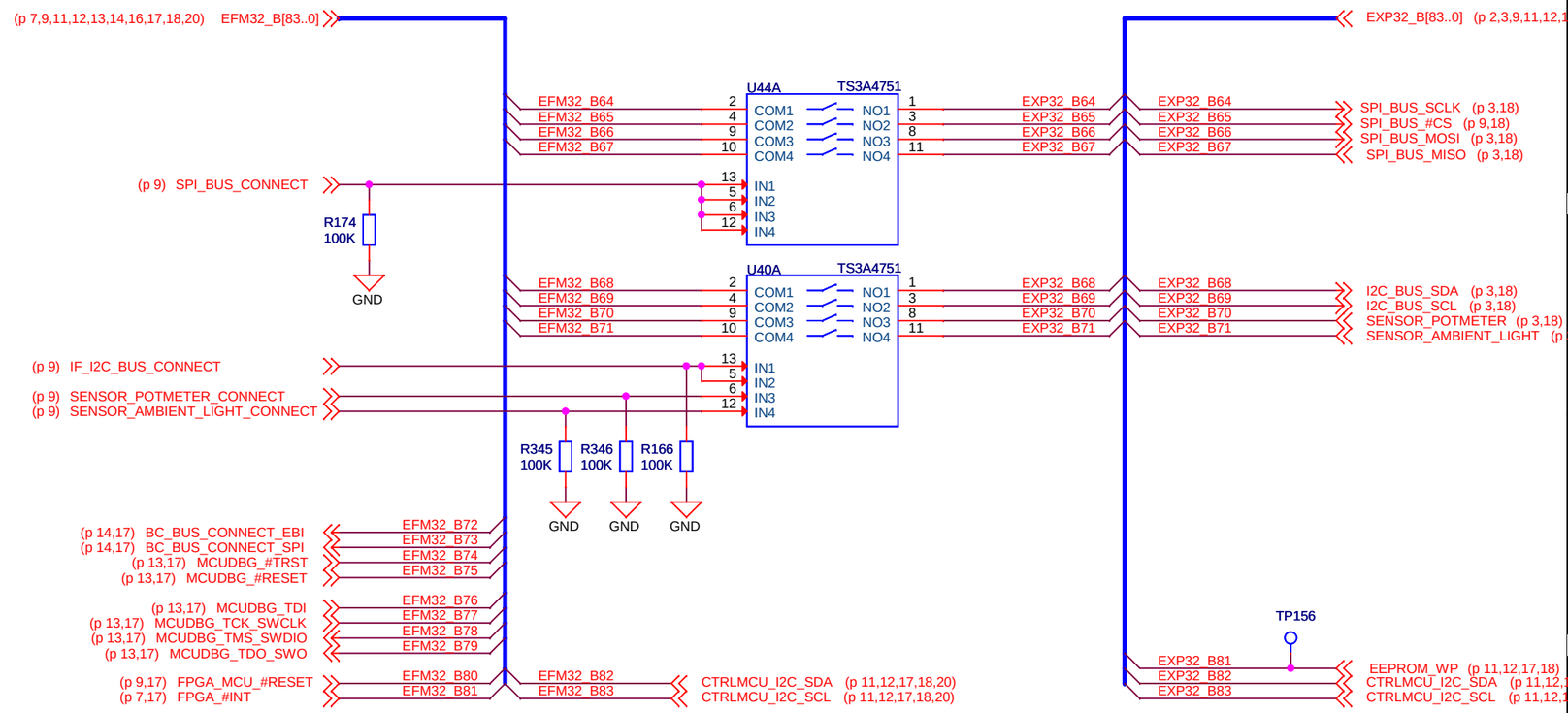


A: Pins 0 -> 83

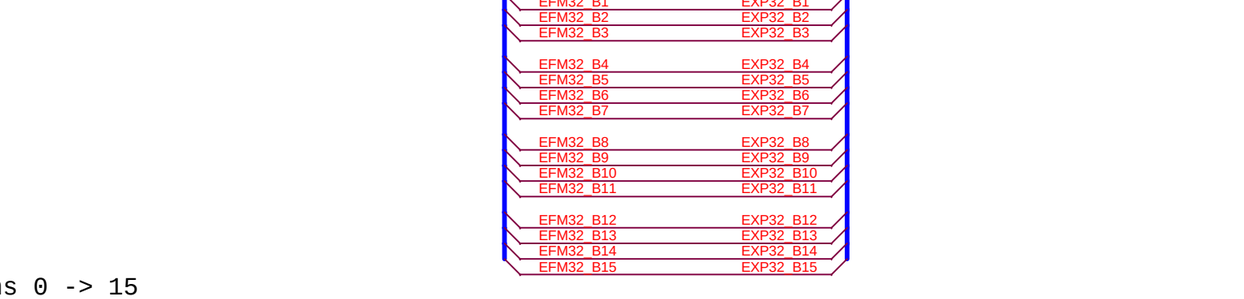




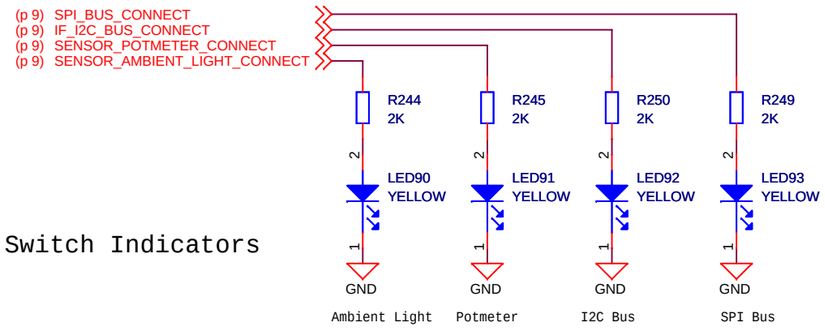
B: Pins 16 -> 43



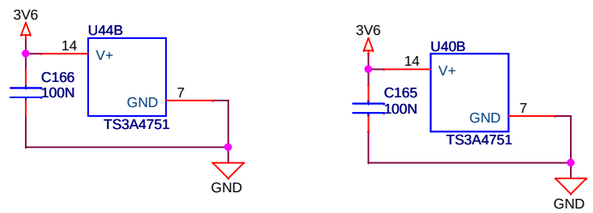
B: Pins 64 -> 83



B: Pins 0 -> 15

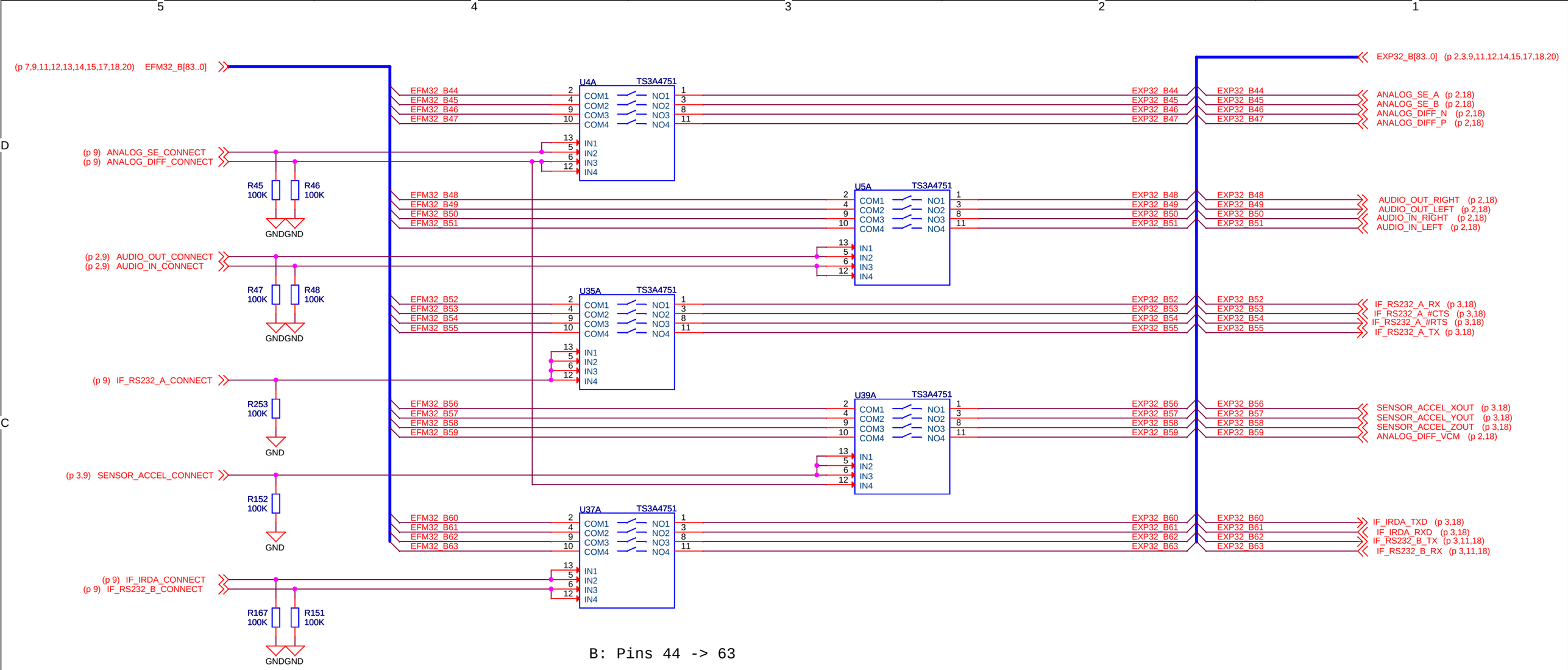


Switch Indicators

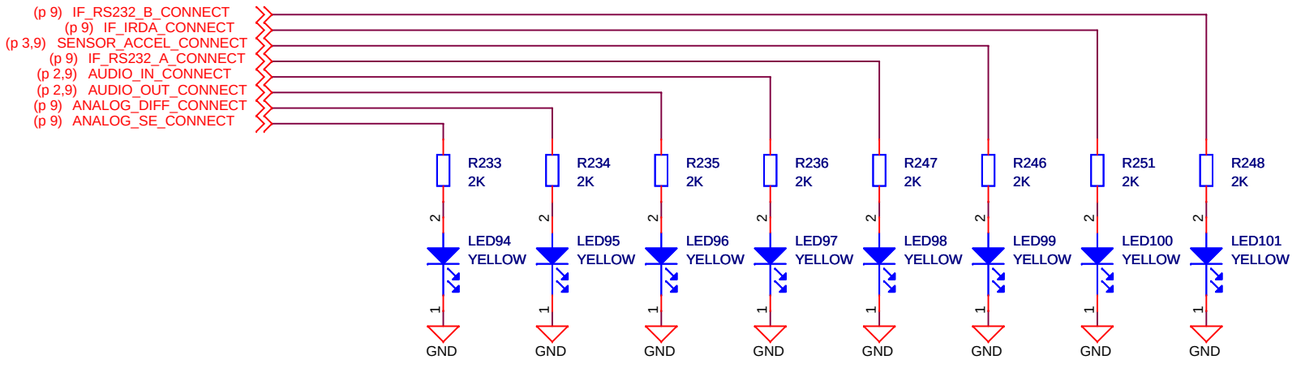


<Schematic Path>
TOP

Schematic Title	**EFM32 Development Kit**
Page Title	**EXP32 signal assignments #1**
Designed: JNO / Approved: JNO	
Size: A3 / BOM Doc No: <Cage Code>	Document number: **BRD3200C** / Revision: A01
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet 15 of 21

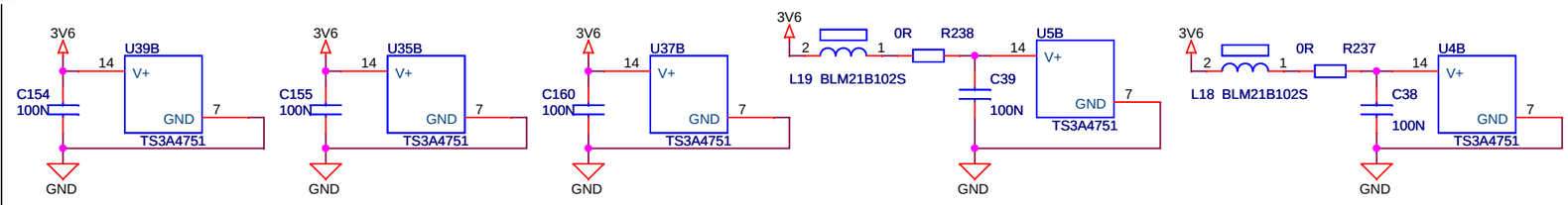


B: Pins 44 -> 63



Switch Indicators

Power & Bypass



<Schematic Path>
TOP

Designed: JNO	Approved: JNO
Size: A3	BOM Doc No: <Cage Code>
Design Created Date: Wednesday, December 03, 2008	

Schematic Title
EFM32 Development Kit - Mainboard

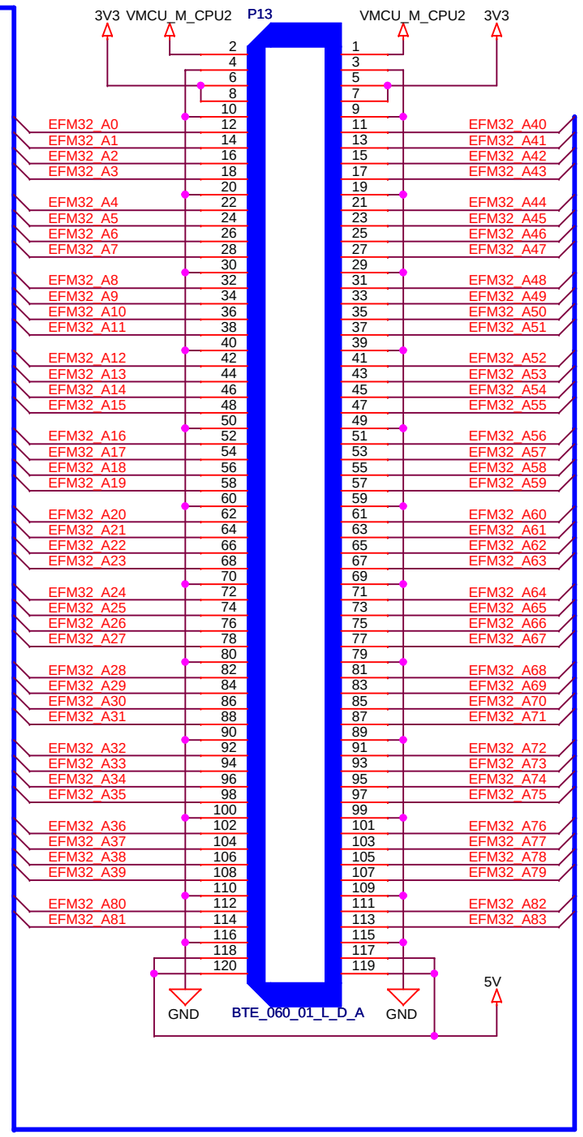
Page Title
EXP32 signal assignments #2

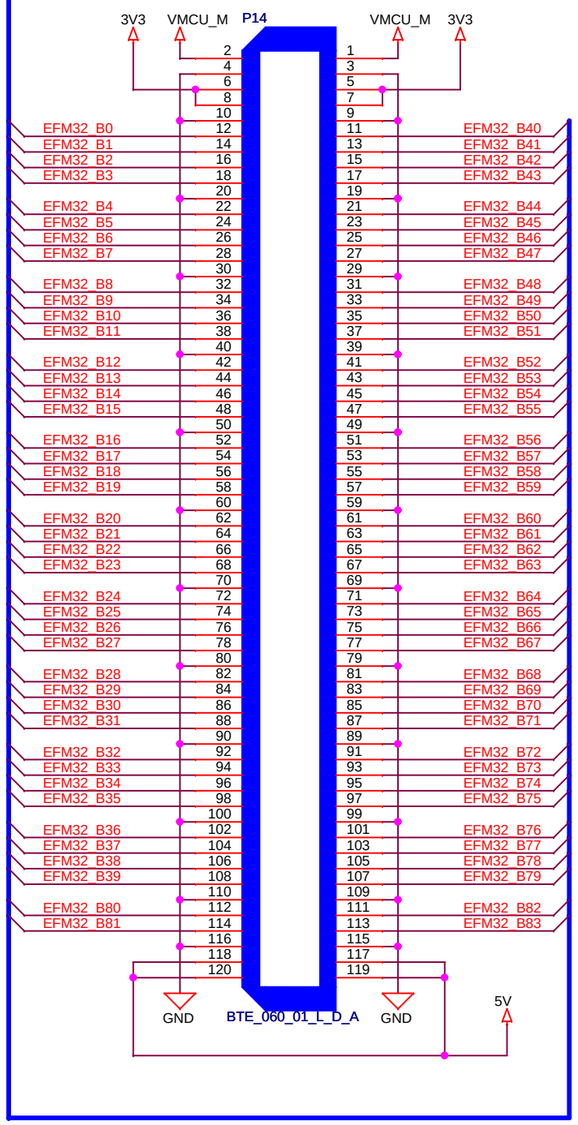
Document number	Revision
BRD3200C	A01

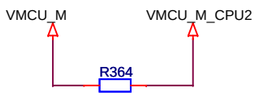
Sheet Created Date	Sheet Modified Date	Sheet
Saturday, March 21, 2009	Monday, June 28, 2010	16of 21

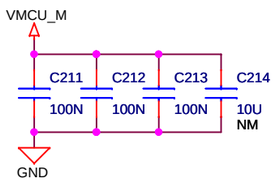
(p 7,9,11,12,13,14,15,16,18,20) EFM32_B[83..0]

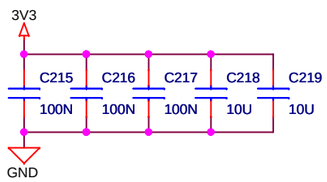
(p 11,12,15,18,20) EFM32_A[83..0]

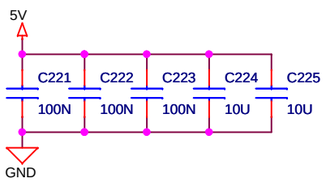


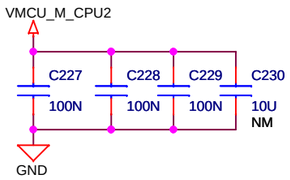


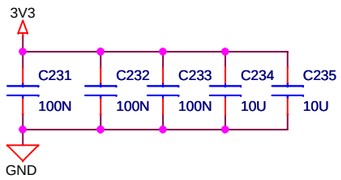


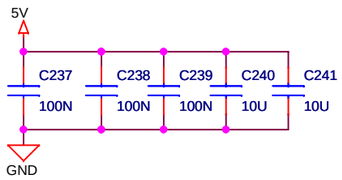






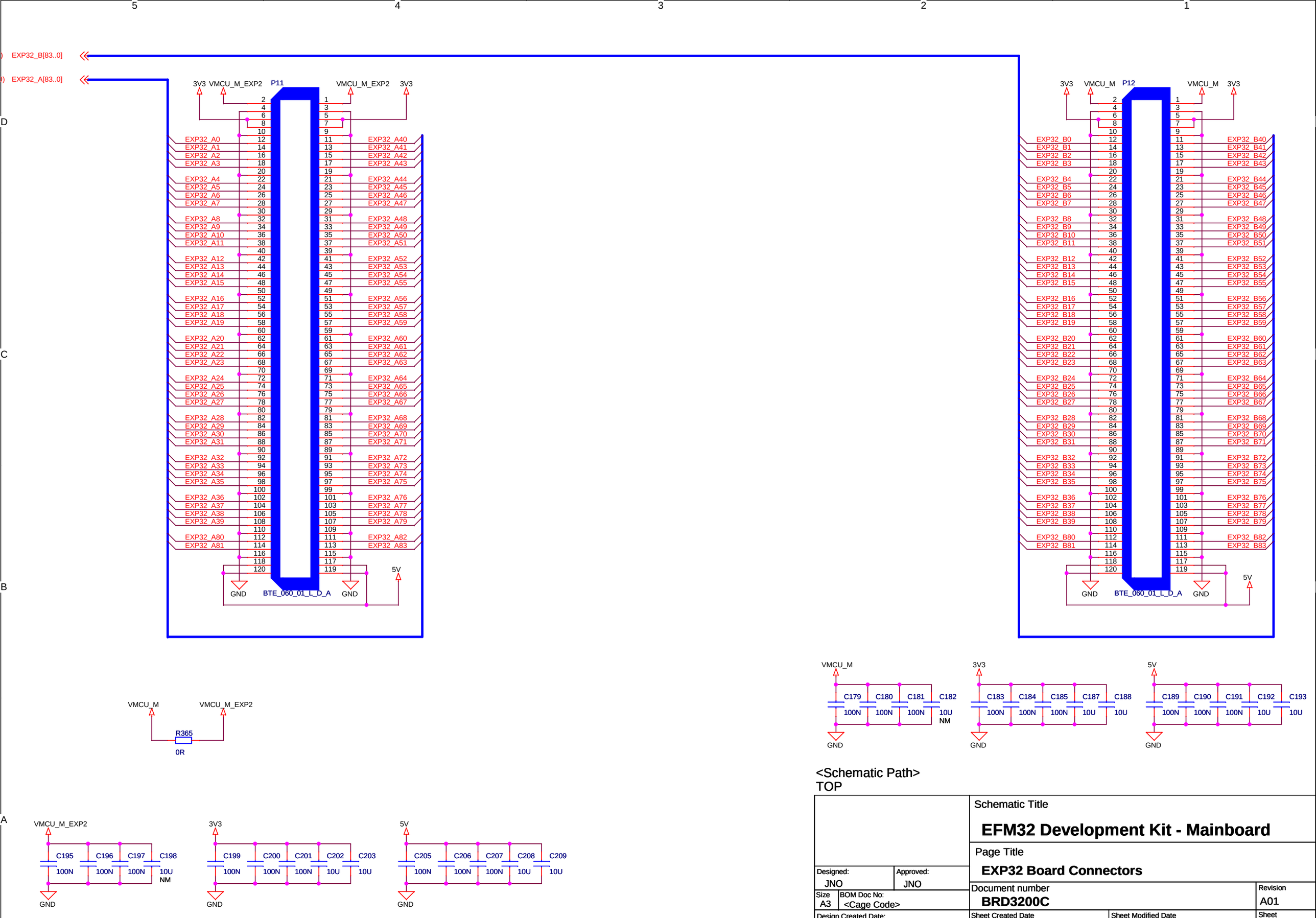






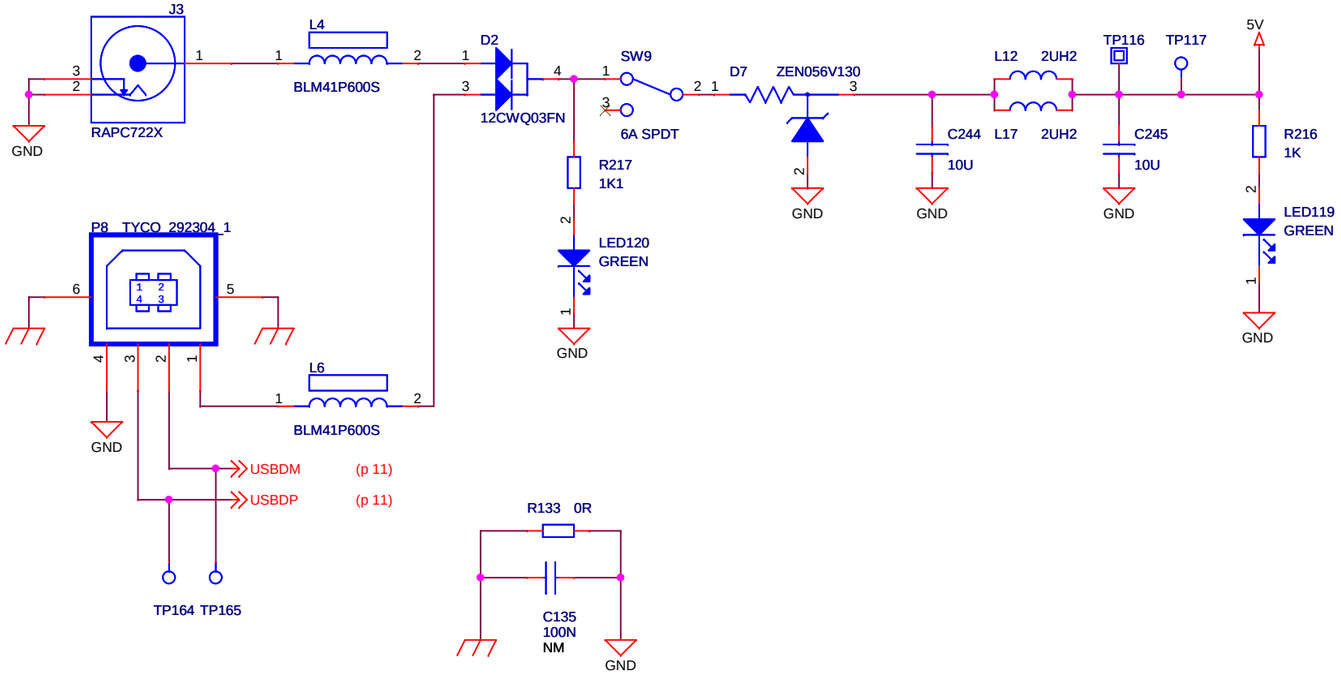
<Schematic Path>
TOP

	Schematic Title: **EFM32 Development Kit - Mainboard**
	Page Title: **EFM32 Board Connectors**
Designed: JNO / Approved: JNO	
Size: A3 / BOM Doc No: <Cage Code>	Document number: **BRD3200C** / Revision: A01
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet: 17 of 21

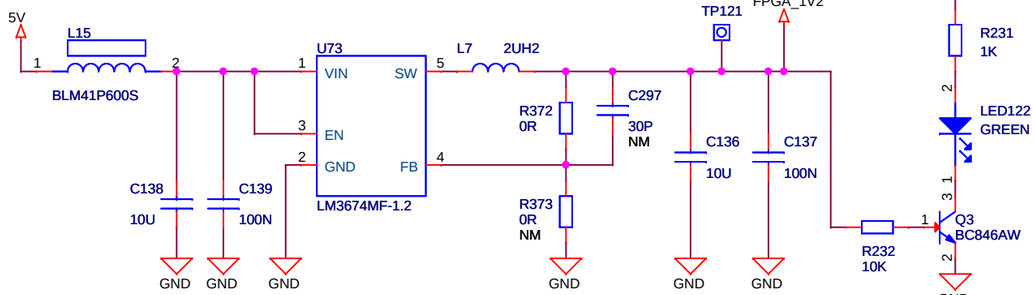


<Schematic Path>
TOP

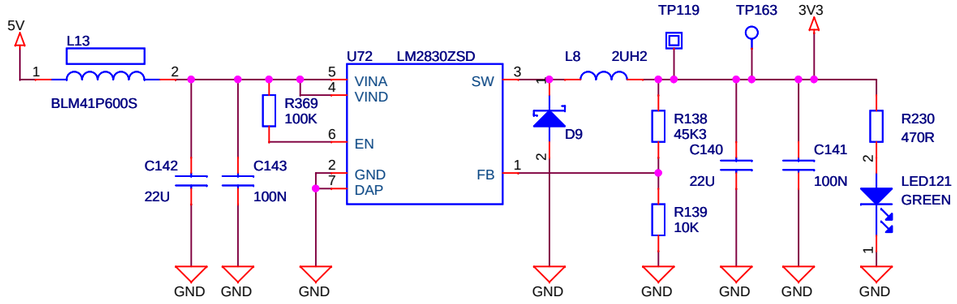
	Schematic Title: **EFM32 Development Kit - Mainboard**
	Page Title: **EXP32 Board Connectors**
Designed: JNO / Approved: JNO	Document number: **BRD3200C** / Revision: A01
Size: A3 / BOM Doc No: <Cage Code>	
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet: 18 of 21



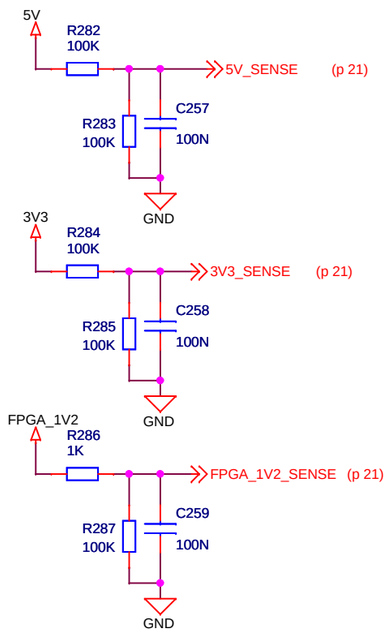
Power input



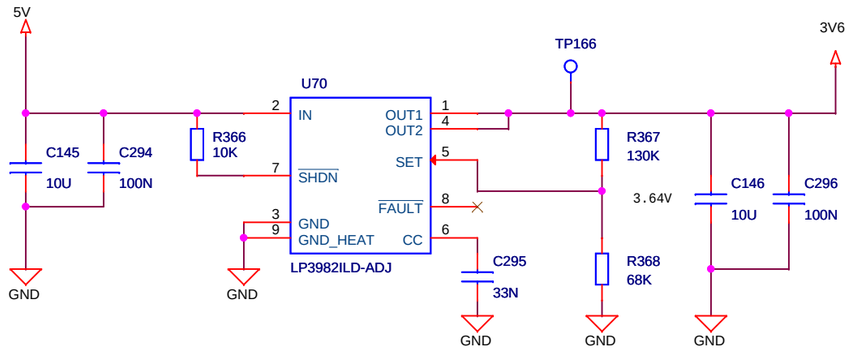
FPGA core voltage regulator

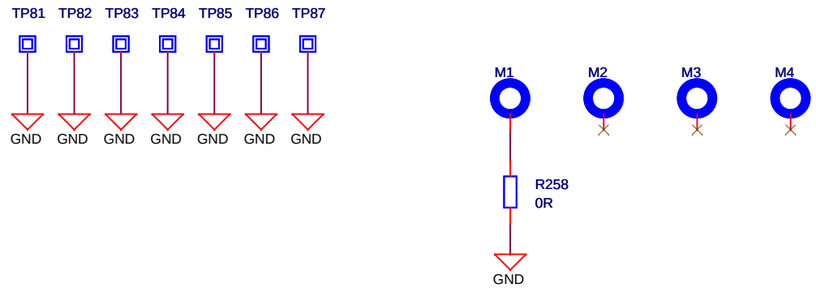


3.3V power regulator



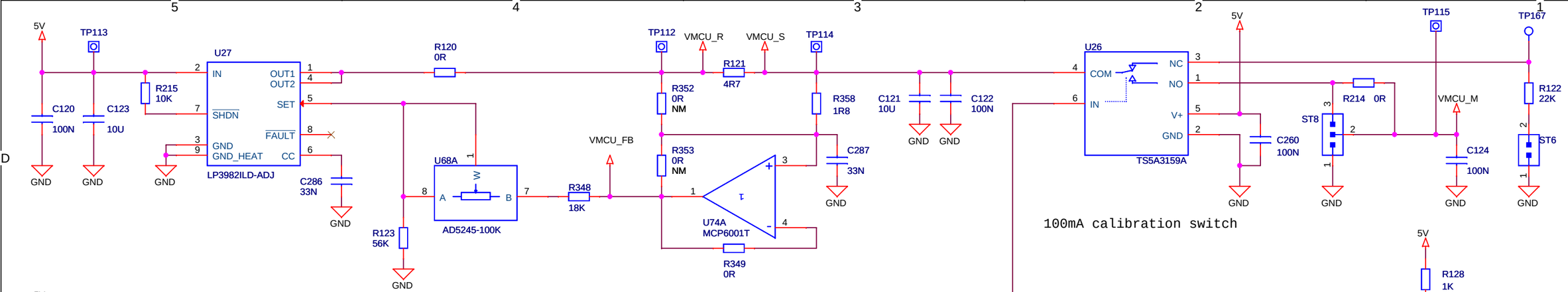
3.6V power regulator

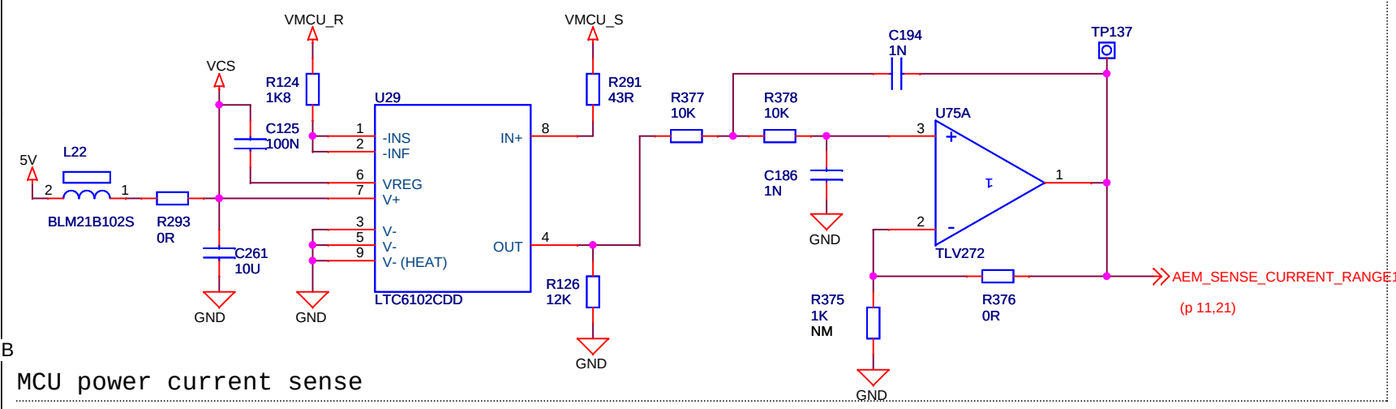


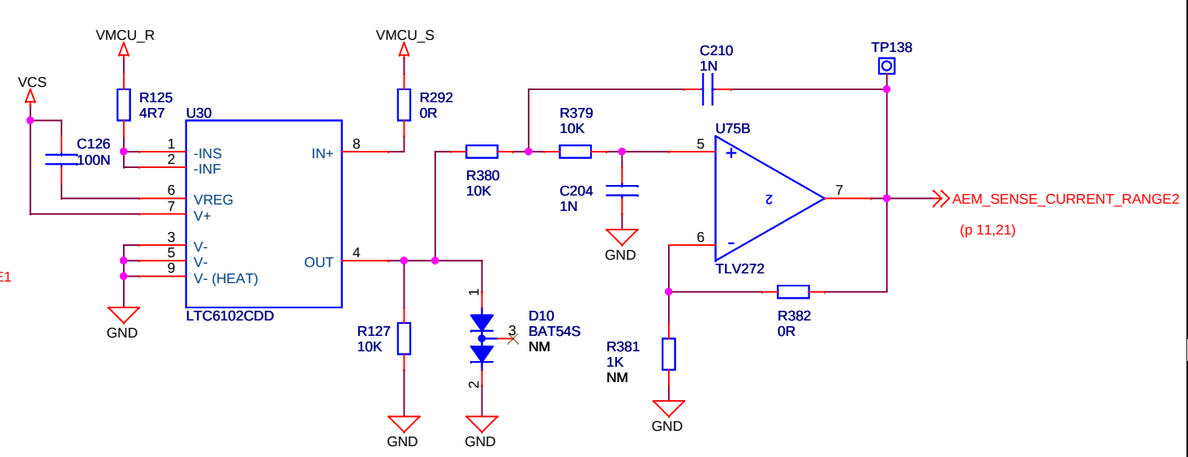


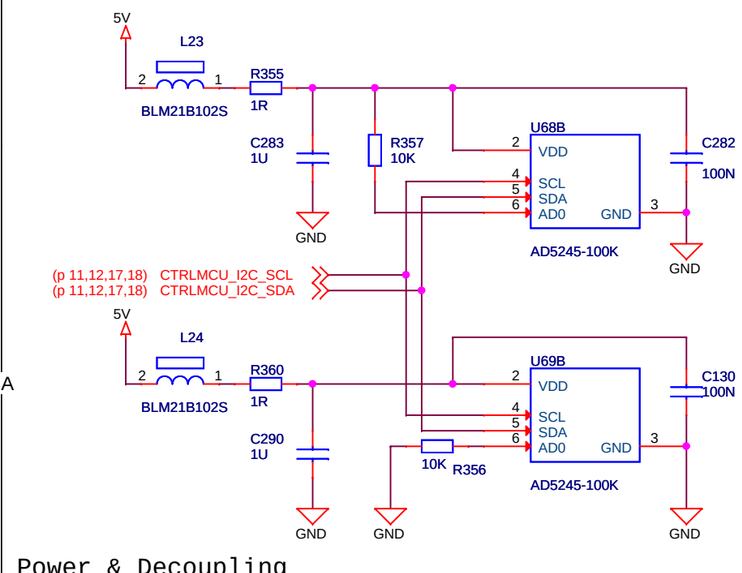
<Schematic Path>
TOP

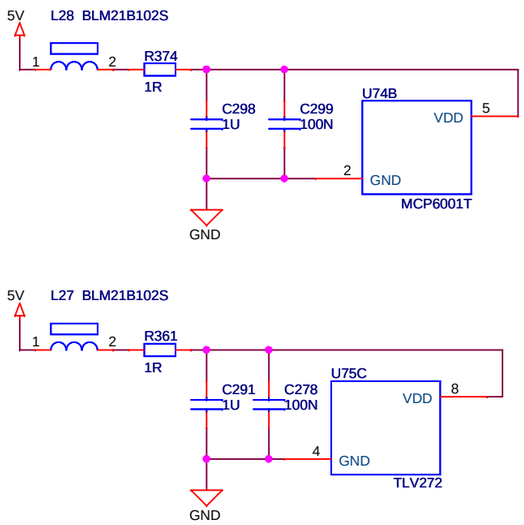
Schematic Title	**EFM32 Development Kit - Mainboard**
Page Title	**Power**
Designed: JNO / Approved: JNO	
Size: A3 / BOM Doc No: <Cage Code>	Document number: **BRD3200C** / Revision: A01
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Saturday, March 21, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet 19 of 21





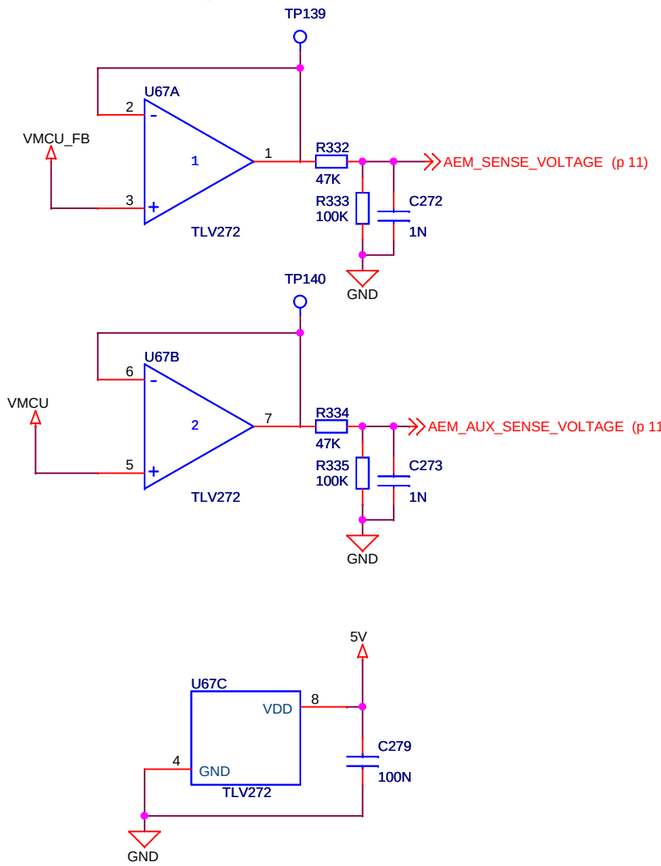


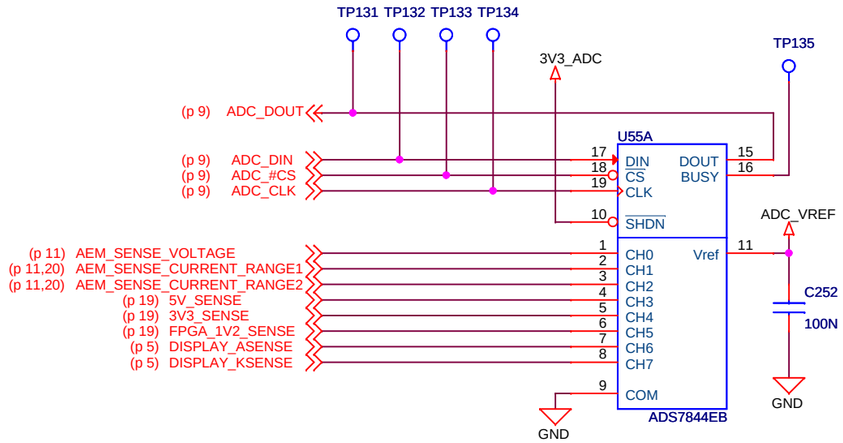


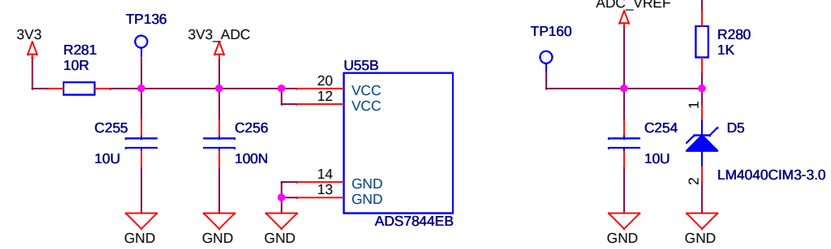


Schematic Title	**EFM32 Development Kit - Mainboard**
Page Title	**EFM Power and AEM**
Designed: JNO / Approved: JNO	Document number: **BRD3200C** / Revision: **A01**
Size: A3 / BOM Doc No: <Cage Code>	
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Wednesday, August 19, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet: 20 of 21

VMCU voltage sense

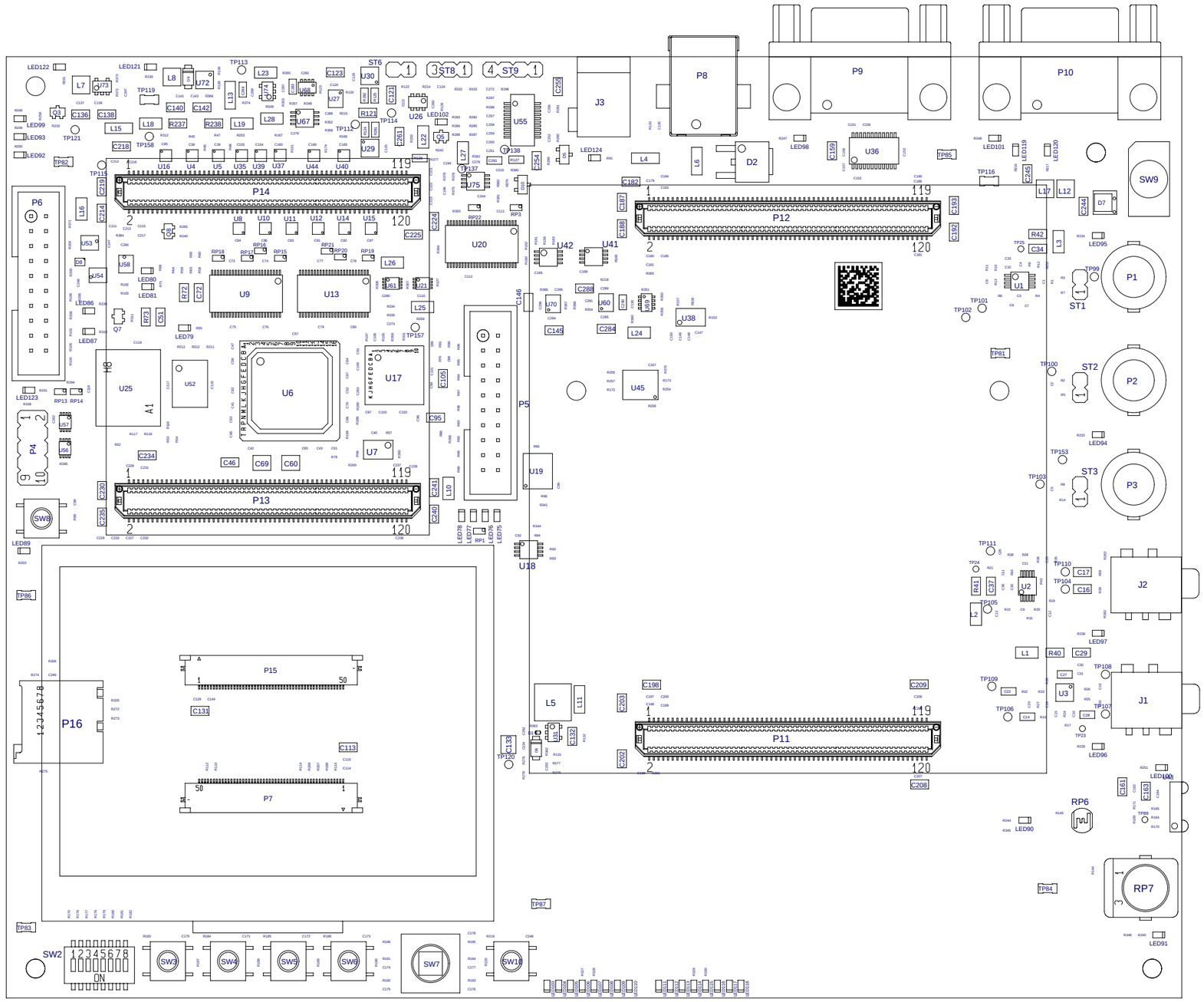




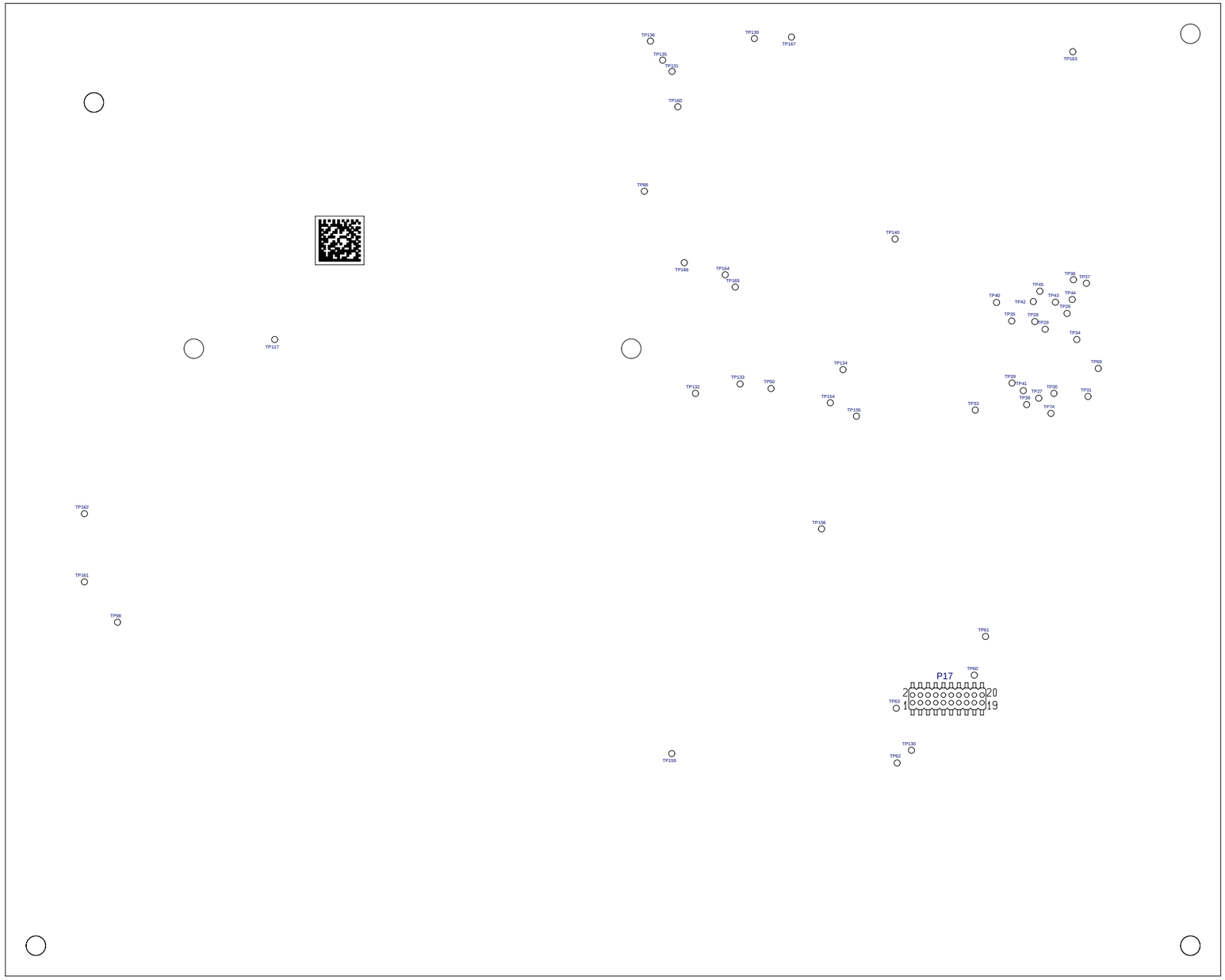


<Schematic Path>
TOP

	Schematic Title: **EFM32 Development Kit - Mainboard**
	Page Title: **Power Monitoring**
Designed: JNO / Approved: JNO	Document number: **BRD3200C** / Revision: A01
Size: A3 / BOM Doc No: <Cage Code>	
Design Created Date: Wednesday, December 03, 2008	Sheet Created Date: Tuesday, September 08, 2009 / Sheet Modified Date: Monday, June 28, 2010 / Sheet: 21 of 21



ENERGY micro	Utført (også faktaansvarig hvis annen) - *Prepared (also subject responsible if other)*		
Referanser - *References*	Dokansv/Godk - *Doc respons/Approved*	Dato - *Date*	Rev
-		2010-01-08	C1
EFM32 DEVELOPMENT KIT	Produktbetegnelse - *Product name*		
	PRINTED BOARD ASSEMBLY		
(Scale 1.25 : 1)	Dokumentnr - *Document No.*		Side - *Sheet*
	BRD3200C1 (PCB3200 REV. 3.1)		1 (2)



ENERGY micro	Utført (også faktaansvarig hvis annen) - *Prepared (also subject responsible if other)*		
Referanser - *References*	Dokansv/Godk - *Doc respons/Approved*	Dato - *Date*	Rev
-		2010-01-08	C1
EFM32 DEVELOPMENT KIT	Produktbetegnelse - *Product name*		
	PRINTED BOARD ASSEMBLY		
(Scale 1.25 : 1)	Dokumentnr - *Document No.*		Side - *Sheet*
	BRD3200C1 (PCB3200 REV. 3.1)		2 (2)

Table of Contents

1. Introduction 2
    1.1. Features 2
    1.2. Board Configuration 2
2. Kit Block Diagram 3
3. Mainboard hardware layout 4
4. Power supply 5
    4.1. USB 5
    4.2. External power supply 5
5. Reset infrastructure 6
    5.1. MCU 6
    5.2. Board controller 6
6. Peripherals 7
    6.1. Pushbuttons 7
    6.2. DIP switches 7
    6.3. Joystick 7
    6.4. LEDs 7
    6.5. Differential analog input 7
    6.6. Single ended analog inputs 7
    6.7. Line in / Audio in 7
    6.8. Line Out / Audio out 8
    6.9. RS232 8
    6.10. Accelerometer 8
    6.11. IrDA 8
    6.12. Potmeter 8
    6.13. Ambient light sensor 8
    6.14. I^2C EEPROM 9
    6.15. I^2C Temperature sensor 9
    6.16. SPI Flash 9
    6.17. microSD 9
    6.18. TFT LCD 9
    6.19. SRAM 9
    6.20. NOR Flash 9
7. Board Support Package 10
    7.1. Installation location 10
    7.2. Resource usage 10
    7.3. Application Programming Interface 10
    7.4. Example Applications 11
    7.5. How to include in your own applications 12
    7.6. Chip errata 12
8. Configuration 13
    8.1. MCU voltage 13
    8.2. Debug settings 13
    8.3. Peripheral configuration 13
    8.4. Program MCU 13
    8.5. Upload files 13
9. Advanced Energy Monitor 14
    9.1. AEM Display 14
    9.2. AEM configuration 14
    9.3. AEM theory of operation 14
    9.4. AEM accuracy and performance 14
10. Board controller 15
    10.1. Register Map 15
    10.2. Register Description 16
11. Connectivity 30
    11.1. Resource connections 30
12. Connectors 33
    12.1. EFM32 and EXP32 connectors 33
    12.2. Debug connector 33
13. Debugging 34
14. Integrated Development Environments 35
    14.1. IAR Embedded Workbench for ARM 35
    14.2. Rowley Associates - CrossWorks for ARM 35
    14.3. CodeSourcery - Sourcery G++ 35
    14.4. Keil - MDK-ARM 35
15. energyAware Commander and Upgrades 36
    15.1. GUI Operation 36
    15.2. Command Line operation 36
    15.3. Upgrades 37
16. Version information 38
17. MCU board 39
18. Prototyping Board 40

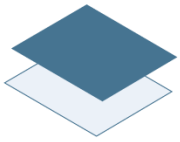
18.1. Overview 40
19. Errata 41
19.1. DVK hardware errata 41
19.2. DVK firmware errata 41
20. Schematic 43

List of Figures

2.1. EFM32-G8XX-DK Block Diagram 3
3.1. EFM32-G8XX-DK hardware layout 4

List of Tables

7.1. GPIO Usage 10
9.1. AEM accuracy 14
11.1. Connections 30
11.2. Nomenclature 32
12.1. Debug connector pinout 33
13.1. Debug modes 34
15.1. energyAware Commander command line options 36
16.1. Current versions 38



ENERGY®
micro

Energy Micro AS
Sandakerveien 118
P.O. Box 4633 Nydalen
N-0405 Oslo
Norway

www.energymicro.com