EFM®32

... the world's most energy friendly microcontrollers

USER MANUAL

Development Kit EFM32G-DK3550

0 1 2 3 4

The EFM32 Gecko Development Kit is a feature rich development platform for evaluation, prototyping and application development for the EFM32 Gecko MCU family with the ARM Cortex-M3 CPU core.

Main features:

- Advanced Energy Monitoring provides real-time information about the energy consumption of an application or prototype design.
- Integrated emulator providing full debug and trace capability
- Exchangeable prototyping board for custom application circuit development

ENERGY® micro

www.energymicro.com

1 Introduction

1.1 Description

The EFM32G-DK3550 is a highly flexible development platform demonstrating some of the EFM32 Gecko microcontroller's many capabilities. The rich feature set makes the kit an excellent platform for evaluating the microcontroller as well as a good starting point for application development.

The EFM32G-DK3550 kit consists of three separate boards:

- 1 x BRD3201A EFM32 Development Kit Motherboard
- 1 x BRD3302A EFM32 G890 MCU plugin board
- 1 x BRD3500B EXP32 prototyping board

The EFM32 G890 MCU is mounted on the plugin board, which plugs into the Motherboard. All the EFM32 GPIO pins are available through pin headers on the prototyping board.

The different peripherals on the motherboard are *not* permanently connected to the MCU. Instead, they are isolated when not in use, in order to prevent undesired current leakage. Connecting peripherals to the MCU can be done by the board controller through a simple API in the kit Board Support Package.

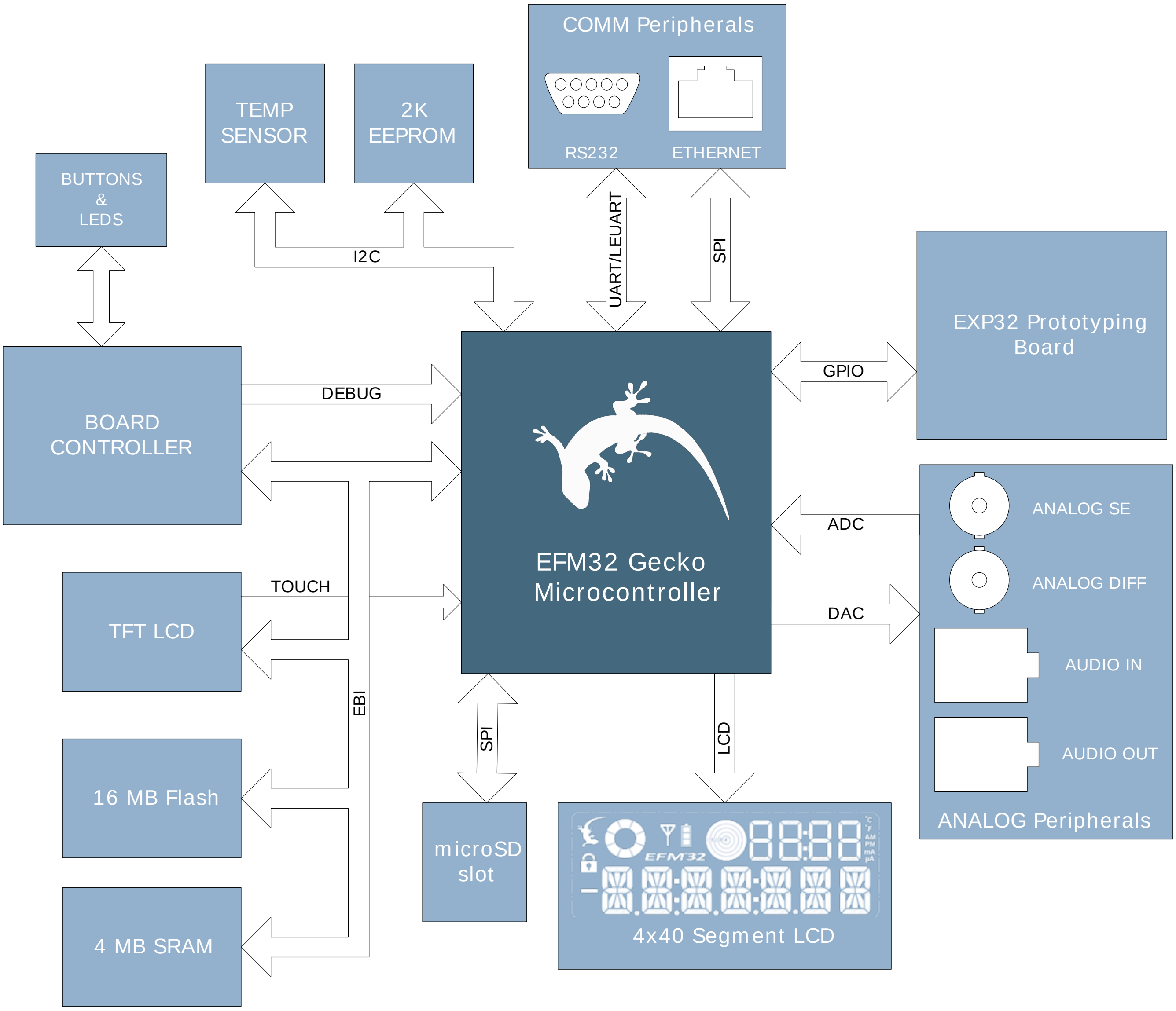
1.2 Features

- 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 trace support and debug out functionality.
- 3.5-inch TFT-LCD 320x240 pixel RGB color display with resistive touch film.
- Smart Board Controller handles configuration and signal routing.
- Single ended and differential ADC inputs.
- Line-in stereo audio input amplifier.
- Line-out stereo audio output amplifier.
- 1 x RS232 Serial Port (DSUB-9).
- 10/100 Mbps Ethernet MAC+PHY with SPI interface
- MicroSD card reader (SPI mode).
- 2Meg x 16 (4MB) PSRAM with 70ns access time.
- 8Meg x 16 (16MB) NOR Flash with 90ns access time.
- 2Kb I2C EEPROM.
- Temperature sensor with I2C interface.
- 5 way joystick.
- 4 User buttons, 4-bit DIP switch and 16 user LEDs.
- 4x40 Segment LCD

2 Kit Block Diagram

An overview of the EFM32 Gecko Development Kit is shown in Figure 2.1 (p. 3)

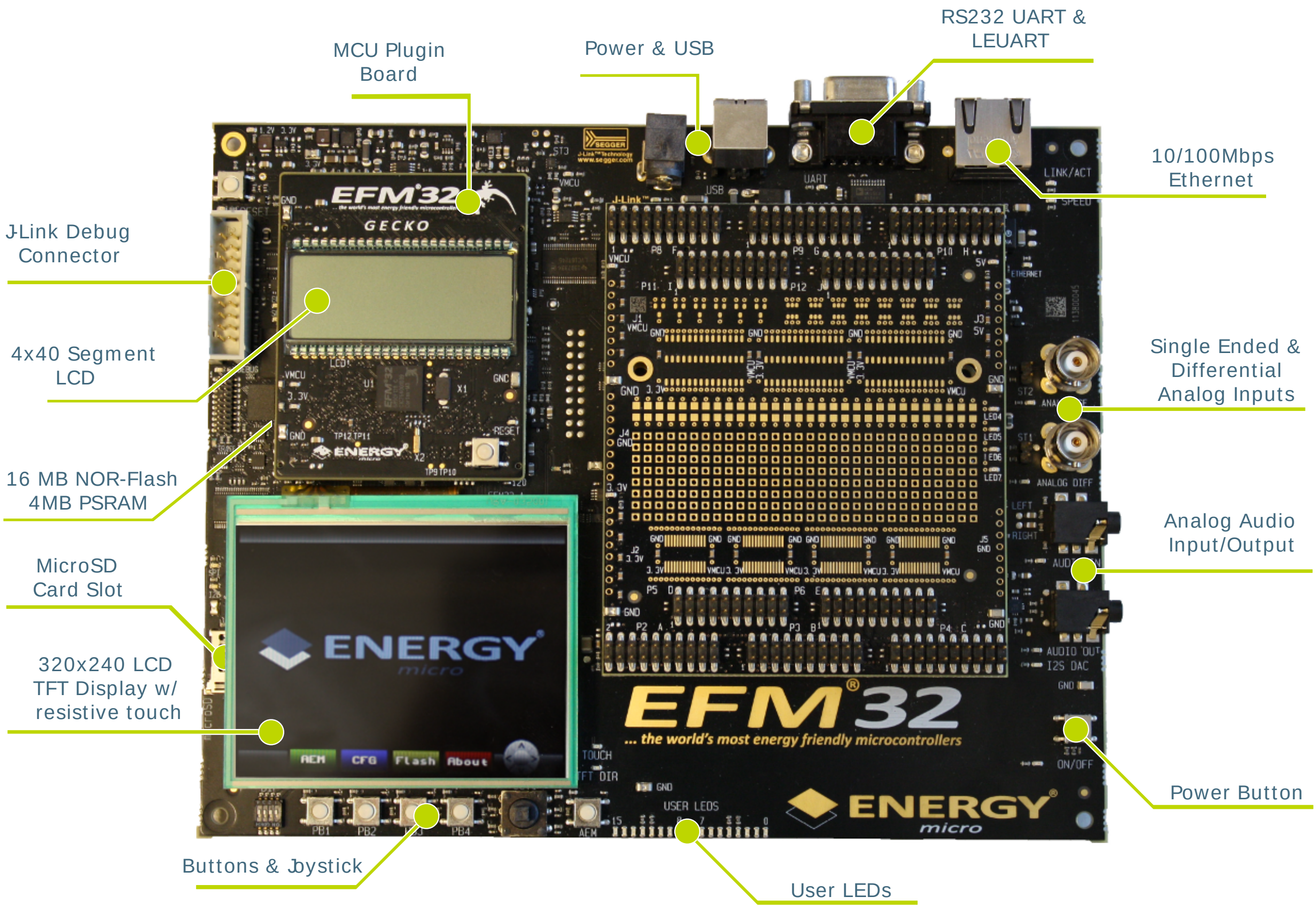
Figure 2.1. EFM32G-DK3550 Block Diagram



3 Kit Hardware Layout

The layout of the EFM32 Gecko Development Kit is shown below.

Figure 3.1. EFM32G-DK3550 hardware layout



4 Using the EFM32G-DK3550

The EFM32 Gecko Development Kit is intended to be a complete platform for developing applications for the EFM32 microcontroller. The embedded debugger allows applications to be downloaded and debugged directly. A set of useful peripherals is provided, and custom hardware can be developed on the prototyping area, where all the microcontroller's IO pins are made available.

By default the peripherals on the board are *not* connected to the MCU. Interfacing the peripherals is done entirely without using jumpers, but instead through the kit's board controller. Two main approaches exist to configuring the board for an application: from within the application itself with the BSP, or by using the kit's user interface.

4.1 Board Support Package

The kit board support package, or BSP, is provided to allow an application to control various aspects of the EFM32G-DK3550 kit. Peripherals can be connected or disconnected with simple calls to the API. The user buttons and LEDs are also accessed through the BSP.

The easiest way to obtain the latest version of the BSP is through Simplicity Studio. It can also be downloaded at: [http://www.energymicro.com/downloads/software].

The BSP can be configured to use two different methods of communication toward the board controller: *SPI mode* or *EBI mode*. In SPI mode the EFM32 communicates with the board controller using a simple 4-wire SPI bus, and in EBI mode the board controller becomes a memory-mapped peripheral in the EFM32's address space using the EFM32's External Bus Interface module.

SPI mode uses fewer pins to communicate with the board controller, but the external memory devices and TFT-display are not available in this mode. Use this mode when the IO taken up by the EBI are needed for other purposes. To enable the board controller in SPI-mode use the BSP function from within the application code:

```
DVK_Init( DVK_Init_SPI );
```

EBI mode is the preferred mode of interfacing to the board controller. This gives access to all the board functions as well as the external memory devices (PSRAM and Flash) and the TFT-LCD display. To enable the board controller in EBI-mode use the BSP function from within the application code:

```
DVK_Init( DVK_Init_EBI );
```

In order to configure the BSP, some dedicated GPIO pins are used. These pins are listed in Table 4.1 (p. 5) , and are normally controlled by the BSP. No manual configuration of these pins is necessary.

Table 4.1. GPIO's used for BSP functions

MCU Pin	Description
PB15	Board controller configuration line 1.
PD13	Board controller configuration line 2.
PE2	Interrupt request from board controller.

Once the BSP has been configured, the different peripherals and kit functions can be accessed through the BSP API. Please refer to Chapter 6 (p. 8) for detailed information about the different kit peripherals and how to access them with the BSP. It can also be useful to take a look at some of the software examples found in Simplicity Studio.

4.2 User Interface

In addition to using the API provided by the BSP, the kit can also be configured through the graphical user interface consisting of the TFT-LCD display together with the buttons PB1 to PB4 and the 5-way joystick located below. The board controller provides a simple menu system, allowing most aspects of the kit to be configured directly.

The user is encouraged to explore the menu system and the different functions provided. Some useful functions that can be performed using the menu system are:

- Enabling or disabling access for individual peripherals.
- Displaying information about the different boards on the kit.
- Getting and displaying information about the EFM32 MCU part mounted on the MCU board
- Displaying real-time current consumption of the EFM32 MCU.
- Uploading and running example applications stored in the kit.
- Adjusting the MCU voltage (VMCU).
- Selecting the debugging mode (IN/OUT/MCU/OFF)

Since the TFT display and keys are shared between the board controller and the EFM32, a separate button labeled "AEM" is present to switch control of the buttons and display. By default, when the kit has been started up, control is given to the board controller, and pressing the buttons interracts with the graphical user interface. Pressing the "AEM" button once switches control over to the EFM32, and pressing it again switches control back. The current state is shown in the top right corner of the display: "KEYS:AEM" means that the board controller has control, and "KEYS:EFM" indicates that the EFM32 has control.

4.3 Simplicity Studio

The first step in getting started with the EFM32 Gecko Development Kit is to download Simplicity Studio from:

[http://www.energymicro.com/simplicity]

The Simplicity Studio software package contains tools, software examples and documentation relevant to developing applications with the development kit. Some important tools included in Simplicity Studio are:

- *energyAware Commander*
- *energyAware Profiler*

The *energyAware Commander* is a tool for updating the kit's firmware, programming the MCU and launching demos.

The *energyAware Profiler* is the PC-side interface to the Advanced Energy Monitor. It provides the possibility to do energy-debugging and profiling of application code.

5 Power and Reset

5.1 USB Power

The EFM32G-DK3550 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).

5.2 External Power Supply

By using the DC jack plug located on the motherboard, the EFM32G-DK3550 can be powered by an external power supply. The voltage must be 5V and the supply must be able to deliver 500mA. This is mainly intended as a supplement to the USB power, for example when a custom circuit on the prototyping board needs more power.

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.

5.3 ON/OFF Button

A power button is situated on the lower right corner of the motherboard. Press once to turn on the kit, and press once again to turn off.

5.4 MCU Reset

The primary user reset for the MCU is the reset button on the MCU board. This will only reset the MCU. The MCU can also be reset by an emulator, either by the on-board debugger, or an externally connected emulator.

5.5 Board Controller Reset

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

6 Peripherals

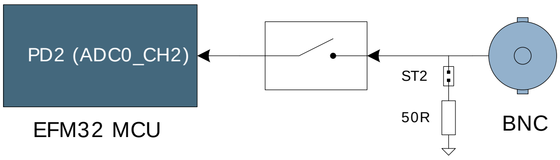
The peripherals on the EFM32 Gecko Development Kit are all isolated from the EFM32's IO pins by default. Peripherals are isolated to prevent excess current leakage into unused peripherals. The different peripherals can be connected using simple functions in the kit's board control software package.

This chapter describes the different peripherals that can be connected to the EFM32, together with the BSP functions required to do this. Before any of the described functions can be called, the board must first be configured in either EBI or SPI mode.

6.1 Single-ended Analog Input

A BNC connector is available for directly connecting an analog signal to the ADC of the EFM32. The input can also be used for digital I/O. If required, 50 ohm termination can be added by soldering in a jumper, ST2.

Figure 6.1. ANALOG SE



The single-ended analog input can be connected by calling:

```
DVK_peripheralAccess( DVK_ANALOG_SE, true );
```

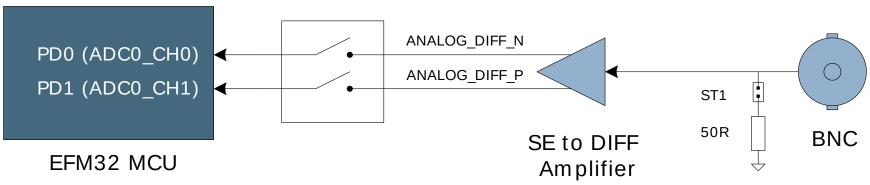
Note

The pin PD2 is shared between the Analog SE, and the Ethernet Controller peripherals. As a consequence, these kit features cannot be used simultaneously.

6.2 Differential Analog Input

The ANALOG DIFF input consists of a BNC connector and a differential operational amplifier with ground as reference. The op-amp output common mode voltage is 1.65V, and also implements a low-pass active filter with a cut-off frequency of 4MHz.

Figure 6.2. ANALOG Diff



This peripheral can be connected by calling:

```
DVK_peripheralAccess( DVK_ANALOG_DIFF, true );
```

Note

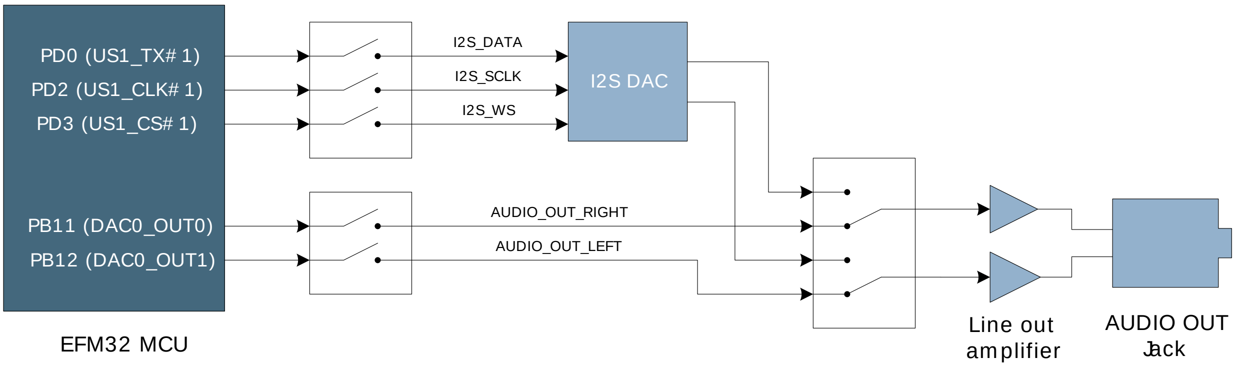
The pins PD0 and PD1 are shared between the Analog Diff peripheral and the Ethernet Controller. As a consequence, these kit features cannot be used simultaneously.

6.3 Audio Out

The kit contains an audio output amplifier with filter connected to a 3.5 mm jack. The gain of the amplifier is fixed to 6 dB and is referenced to ground. The filter is a 3-pole linear phase MFB filter with a cutoff frequency (at -3 dB) of 27 kHz. Two possibilities exist to drive the audio output amplifier:

- Using the internal DAC of the EFM32
- Using the external I2S DAC on the motherboard

Figure 6.3. Audio Out Block Diagram



As shown in the block diagram above, a multiplexer is used to select between the two possible audio sources. The multiplexer and isolation switches are controlled by the board controller, and can be enabled by calling the appropriate function in the BSP API:

- The audio output amplifier can be connected to the EFM32's internal DAC by calling

```
DVK_peripheralAccess(  DVK_AUDIO_OUT, true );
```

- The audio output amplifier can be connected to I2S DAC by calling

```
DVK_peripheralAccess( DVK_I2S, true );
```

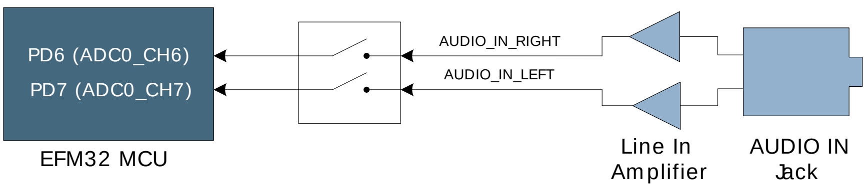
Note

The EFM32G890F128 does not support I2S mode on it's USARTs, and so the I2S DAC peripheral is not natively supported on this kit configuration. It is however possible to write code to use the GPIO's to control the DAC.

6.4 Audio In

An audio input amplifier with filter is present, and can be connected to the ADC of the EFM32. 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 6 dB attenuation.

Figure 6.4. Audio In Block Diagram



The line in amplifier can be connected directly to the EFM32 by calling

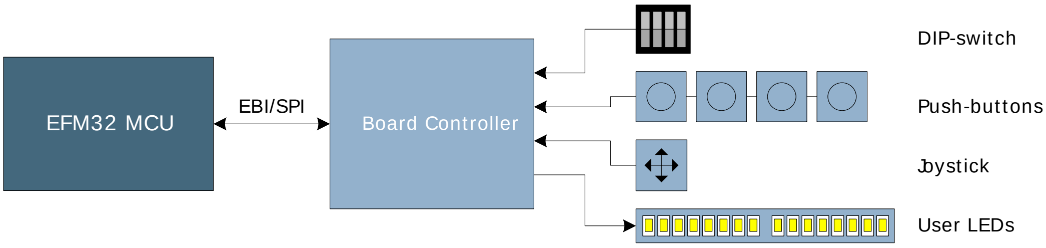
```
DVK_peripheralAccess( DVK_AUDIO_IN, true );
```

6.5 User Interface Peripherals

A set of buttons and LEDs are provided as a simple way of interfacing to applications. These peripherals include:

- A 4-way DIP Switch
- 4 Push-Buttons
- A 5-way Joystick
- 16 User LEDs

Figure 6.5. User interface



The buttons and LEDs are not directly connected to the MCU, instead the board controller is used to read button states and set the LED outputs. This can be done with a set of BSP API functions

- `uint16_t DVK_getPushButtons( void );`
- `uint16_t DVK_getJoystick( void );`
- `uint16_t DVK_getDipSwitch( void );`
- `void DVK_setLEDs( uint16_t leds );`
- `uint16_t DVK_getLEDs( void );`

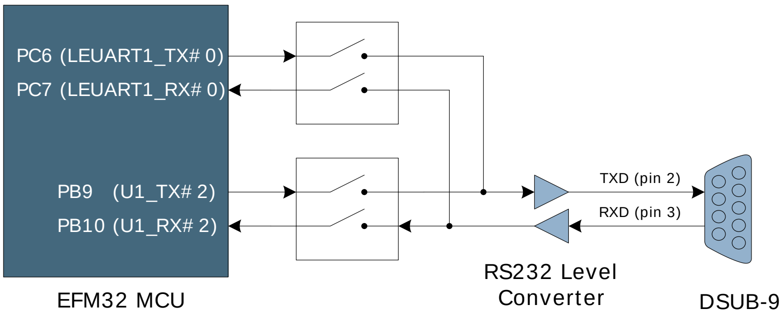
Note

The push-buttons are also used to control the Advanced Energy Monitor (AEM) application. A separate button, labeled "AEM" is used to switch the role of the push-buttons.

6.6 RS232

An RS232 level converter together with a DSUB-9 connector is provided for serial communication between the EFM32 and an external device. The pinout is such that the kit is the DCE (Data Circuit-terminating Equipment). Hardware flow-control signals are not used.

Figure 6.6. RS232



The RS232 peripheral can be connected either to a standard UART peripheral, or to the Low Energy UART (LEUART) of the EFM32.

- The audio output amplifier can be connected to the EFM32's UART peripheral by calling

  `DVK_peripheralAccess( DVK_RS232_UART, true );`

- The audio output amplifier can be connected to the EFM32's LEUART peripheral by calling

  `DVK_peripheralAccess( DVK_RS232_LEUART, true );`

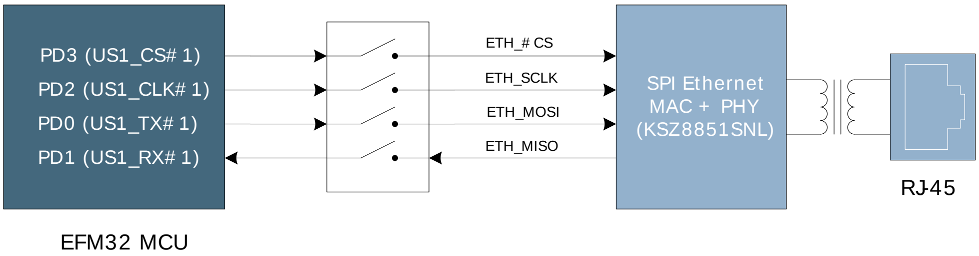
The RS232 transceiver can also be shut down to prevent excess current leakage when the UART or LEUART is not in use, without disconnecting the switches. This can be done with:

```
DVK_peripheralAccess( DVK_RS232_SHUTDOWN, true );
```

6.7 Ethernet

The kit contains a single-chip Fast Ethernet controller consisting of a 10/100 physical layer transceiver (PHY), a MAC and an SPI interface. Also present are the required magnetics and RJ-45 connector to provide network connectivity to an application.

Figure 6.7. SPI Ethernet MAC+PHY



The Ethernet controller has 12KB buffer memory on the receive queue and 6KB on the transmit queue, and supports HP Auto-MDIX. Two LEDs are placed next to the RJ-45 connector to indicate link speed and activity.

The Ethernet interface can be enabled and connected to the EFM32 by calling

```
DVK_peripheralAccess( DVK_ETH, true );
```

> ***Note***
>
> The pins PD0 to PD3 are shared between the Ethernet Controller, and the Analog Input peripherals. As a consequence, these kit features cannot be used simultaneously.

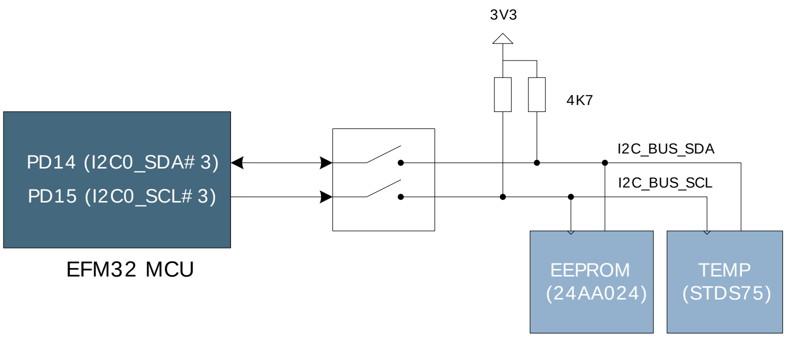
6.8 I²C EEPROM and Temperature Sensor

Two devices are attached to an I²C bus which can be connected to the EFM32. These devices are:

- Temperature Sensor
- 2Kb EEPROM

Both devices support a maximum I²C bus speed of 400 kHz.

Figure 6.8. I2C Bus



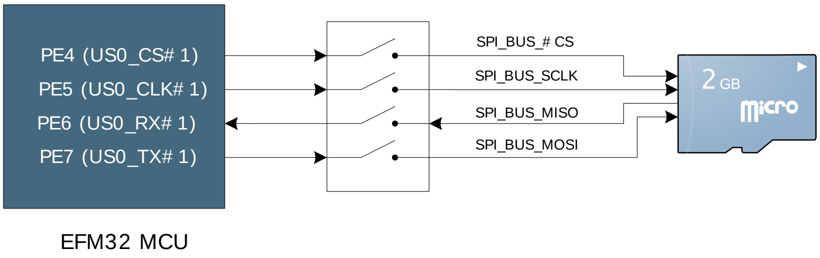
The EEPROM device consists of 256 bytes (256 x 8) and has a lifetime of 1,000,000 erase/write cycles. The EEPROM's I²C address is 0xA0.

The temperature sensor can measure temperatures from -55 to +125°C, with selectable resolution between 9 and 12 bits. The temperature sensor's I²C address is 0x90.

6.9 microSD

A microSD card can be connected to the EFM32 through the serial peripheral bus. The card slot is situated under the LCD display. This allows for applications with large storage and/or file system requirements.

Figure 6.9. microSD



The microSD card slot can be connected to the EFM32 with the BSP function:

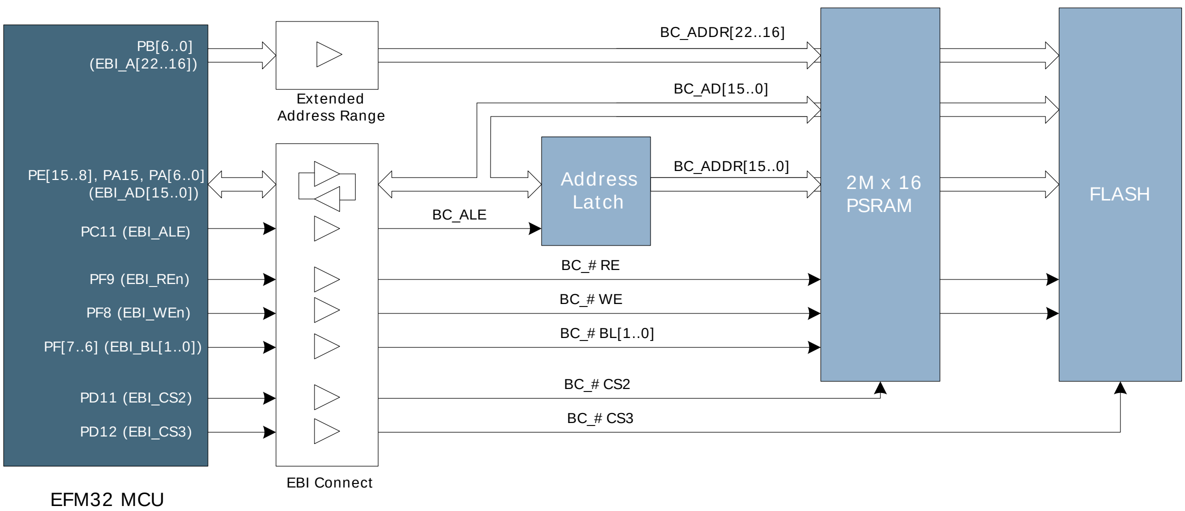
```
DVK_peripheralAccess( DVK_MICROSD, true );
```

6.10 Flash and PSRAM

Two memory devices are available through the EFM32's external bus interface:

- A 4MB (2M x 16) PSRAM
- A 16MB (8M x 16) NOR Flash

Figure 6.10. EBI peripherals



As shown in figure Figure 6.10 (p. 12), the PSRAM and FLASH devices are selected by the EBI_CS2 and EBI_CS3 signals, respectively. The PSRAM and Flash devices map to the EFM32's address space as following:

- PSRAM: 0x88000000 to 0x883FFFFF
- Flash: 0x8C000000 to 0x8CFFFFFF

The EBI is automatically configured by the BSP for all external memory devices when the board is configured in EBI mode with:

```
DVK_Init( DVK_Init_EBI );
```

Although the EFM32 Gecko MCU's EBI module does not support extended addressing, the extended address range of the connected memories can still be utilized by using GPIO's (PB[6..0]) to control the higher address bits. For the DK3550, the extended address range is disabled by default, but can be enabled by calling:

```
DVK_EBI_extendedAddressRange( true );
```

Note

With extended addressing mode *disabled*, only 128 KB of PSRAM and only 128 KB of flash is available.

6.11 TFT-LCD Display

The EFM32 Gecko Development Kit contains a 320x240 pixel TFT-LCD display, which is used both as a graphical user interface toward the kit itself, as well as a possible output device for the EFM32 MCU. The "AEM" button is used to switch control of the TFT-LCD display between the board controller and the EFM32 MCU.

The TFT-LCD display has a built-in SSD2119 controller, which can be interfaced with the EFM32's EBI module as a memory mapped peripheral.When configured, the display is mapped to the base address 0x84000000. Please refer to one of the TFT software examples on how to use this peripheral.

6.12 Resistive Touch Screen

The TFT-LCD display is covered by a resistive touch panel, which is connected to some ADC pins of the EFM32 microcontroller.

Figure 6.11. Resistive Touch Film

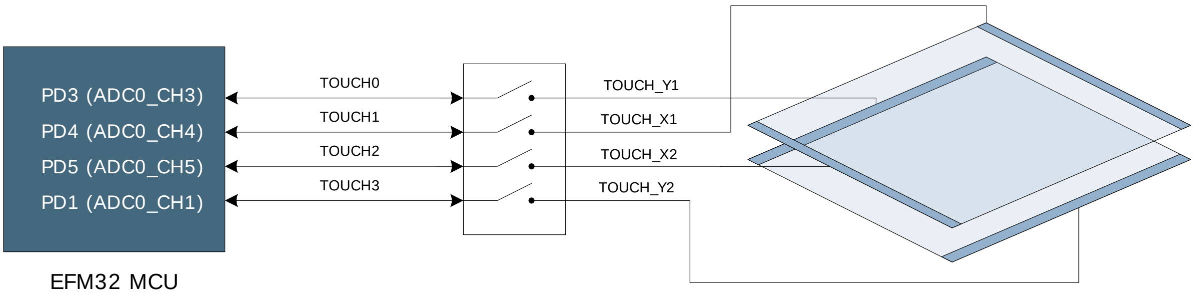


Figure shows how the Resistive Touch Film is connected. When touched, the X-position can be read out by applying a voltage between the X1 and X2 electrodes and measuring the voltage on the Y1 or Y2 electrodes. Likewise, the Y-position can be read out by applying a voltage accross the Y1 and Y2 electrodes and measuring the X1 or X2 electrodes.

The resistive touch screen can be accessed with the BSP command:

```
DVK_peripheralAccess( DVK_TOUCH, true );
```

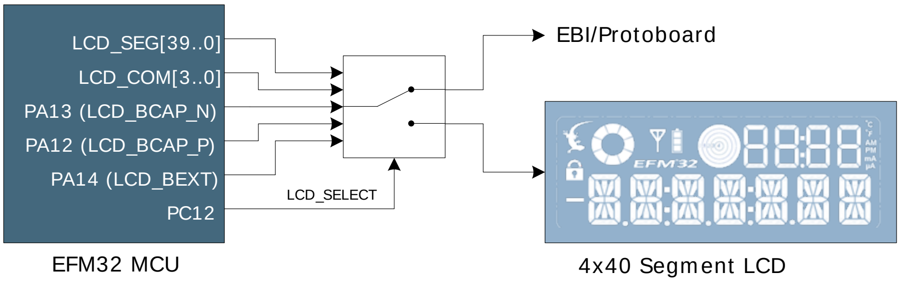
Note

The pins PD1 and PD3 are shared between the Resistive Touch, the Ethernet Controller, the I2S DAC and the Analog Diff peripherals. As a consequence, these peripherals cannot be used simultaneously.

6.13 Segment LCD

The EFM32G890 plug-in board contains a 160 Segment LCD that can be driven directly by the EFM32. In order to support both EBI peripherals and the Segment LCD, analog muxes are used to switch between the two functions. Pin PC12 is used to controll these switches.

Figure 6.12.



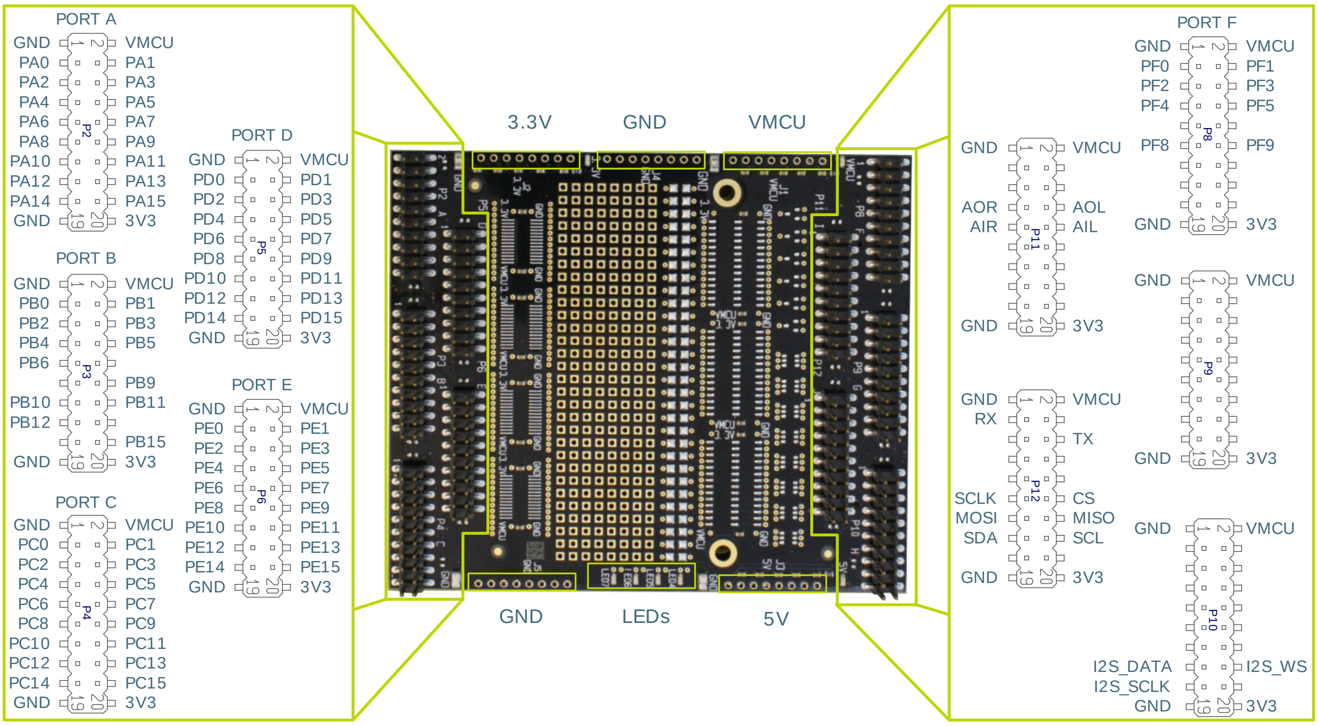
When the LCD_SELECT signal is enabled, a 22nF capacitor is connected between LCD_BCAP_N and LCD_BCAP_P, and a 1uF capacitor is connected between LCD_BEXT and GND. This allows the use of the internal voltage booster to power the LCD if desired.

7 Prototyping Board

The Prototyping Board is a plugin board that contains a large area for constructing custom circuits. It contains a "veroboard" area and many unpopulated footprints which can be used for different SMT parts. Each TSSOP and SSOP site had decoupling capacitors close by.

*All the EFM32 GPIO pins are made available on pin headers*Figure 7.1 (p. 15) is an illustration which shows how the MCU GPIO pins are mapped to the Prototyping Board.

Figure 7.1. Prototyping Board



Additionally, the Prototyping Board also contains connection points for different voltages: 3.3V, 5V, GND, and VMCU. Any current drawn from the VMCU pins will also be measurable with the Advanced Energy Monitor, allowing the whole circuit to be evaluated.

7.1 Peripheral Signals

In addition to the mapping of MCU pins, some of the kit's peripherals are also mapped directly to the Prototyping Board. In Figure 7.1 (p. 15) the pins labeled "Xn" are extra peripheral functions. Note that these pins are connected to the peripherals "after" the isolation switches, so calls to the BSP are not necesarry to enable/connect them. The table below shows which peripheral function signals are mapped to which pins on the Prototyping board

This can be useful when a peripheral cannot be used normally because the required pins on the EFM32 are already used for another purpose. Custom connections between EFM32 pins and some kit peripherals can then be made on the Prototyping Board.

Table 7.1. Peripheral functions mapped directly to the Prototyping Board

Prototyping Board pin	Signal Name	Description
P11.7	AUDIO_OUT_RIGHT	Audio out right channel (before audio out mux)
P11.8	AUDIO_OUT_LEFT	Audio out left channel (before audio out mux)
P11.9	AUDIO_IN_RIGHT	Audio in right channel

Prototyping Board pin	Signal Name	Description
P11.10	AUDIO_IN_LEFT	Audio in left channel
P12.3	IF_RS232_RX	RS232 receive signal
P12.6	IF_RS232_TX	RS232 transmit signal
P12.11	SPI_BUS_SCLK	MicroSD serial clock
P12.12	SPI_BUS_#CS	MicroSD chip select
P12.13	SPI_BUS_MOSI	MicroSD data in
P12.14	SPI_BUS_MISO	MicroSD data out
P12.15	I2C_BUS_SDA	I^2C EEPROM and Temperature sensor serial data
P12.16	I2C_BUS_SCL	I^2C EEPROM and Temperature sensor serial clock
P10.15	I2S_DATA	I2S DAC serial data
P10.16	I2S_WS	I2S DAC word select
P10.17	I2S_SCLK	I2S DAC serial clock

8 Advanced Energy Monitor

8.1 Usage

The AEM (Advanced Energy Monitor) data is collected by the board controller and can be displayed by the energyAware Profiler, available through Simplicity Studio. By using the energyAware Profiler, current consumption and voltage can be measured and linked to the actual code running on the EFM32 in real time.

The current consumption data can also be viewed directly on the TFT-LCD display of the kit, by selecting the "AEM" menu function. The scale is logarithmic, and the time scale of the graph can be adjusted (AEM > CFG > Graph x scale).

8.2 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 start-up of the kit, and when VMCU is changed, an automatic calibration of the AEM is performed. This calibration compensates for the offset error in the sense amplifiers.

8.3 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 the accuracy of the two current sense amplifiers in different ranges.

Table 8.1. 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.

9 Debugging

The EFM32 Gecko Development Kit contains a built-in J-Trace for Cortex-M3 from Segger. It is a fully functional debugger capable of both serial wire debugging and trace (ETM). The embedded debugger can also be used to download flash and debug external targets. In addition to the internal debugger, using an external debugger is also supported.

9.1 Debug Modes

The different debug modes are referred to as Debug IN, Debug OUT, Debug MCU and Debug OFF, and are summarized in Table 9.1 (p. 18). Switching between the different debugging modes can either be done with the User Interface (CFG > Debug Control), or through the *energyAware Commander* tool.

Table 9.1. Debug modes

Mode	Description
Debug MCU	In this mode the built-in debugger is connected to EFM32 on the MCU plugin board. The debug connector on the kit is not used.
Debug IN	In this mode the built-in debugger is disconnected, and an external debugger can be connected to debug the EFM32 on the MCU plugin board.
Debug OUT	In this mode the EFM32 on the MCU plugin board is disconnected, and the built-in debugger can be used to debug an EFM32 in an external application.
Debug OFF	In this mode both the debug connector and the built-in debugger is disconnected.

9.2 Trace

Additional debugging modes are provided for Trace functionality. The Trace modes are similar to the Debug modes, but have Trace enabled as well as SWD.

Table 9.2. Trace modes

Mode	Description
Trace MCU	In this mode the built-in J-Trace is connected to EFM32 on the MCU plugin board, and Trace is enabled. The debug connector on the kit is not used.
Trace IN	In this mode the built-in debugger is disconnected, and an external Trace emulator can be connected to debug the EFM32 on the MCU plugin board.
Trace OUT	In this mode the EFM32 on the MCU plugin board is disconnected, and the built-in J-Trace can be used to run Trace on an EFM32 in an external application.

10 Schematics, Assy Drawings and BOM

The schematics for the three different boards included in the EFM32 Gecko Development Kit are available through Simplicity Studio when the correct kit documentation package has been installed.

11 Kit Errata

12 Document Revision History

12.1 Revision 0.21

2012-03-16

Minor update. Fixed some typo's and added segment LCD figure

12.2 Revision 0.20

2012-01-30

Updated peripheral and protoboard section. Added section on segment LCD.

12.3 Revision 0.10

2011-10-04

First revision with revision history.

A Disclaimer and Trademarks

A.1 Disclaimer

Energy Micro AS intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Energy Micro products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Energy Micro reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Energy Micro shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Energy Micro. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Energy Micro products are generally not intended for military applications. Energy Micro products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

A.2 Trademark Information

Energy Micro, EFM32, EFR, logo and combinations thereof, and others are the registered trademarks or trademarks of Energy Micro AS. ARM, CORTEX, THUMB are the registered trademarks of ARM Limited. Other terms and product names may be trademarks of others.

B Contact Information

B.1 Energy Micro Corporate Headquarters

Postal Address	Visitor Address	Technical Support
Energy Micro AS P.O. Box 4633 Nydalen N-0405 Oslo NORWAY	Energy Micro AS Sandakerveien 118 N-0484 Oslo NORWAY	support.energymicro.com Phone: +47 40 10 03 01

www.energymicro.com
Phone: +47 23 00 98 00
Fax: + 47 23 00 98 01

B.2 Global Contacts

Visit **www.energymicro.com** for information on global distributors and representatives or contact **sales@energymicro.com** for additional information.

Americas	Europe, Middle East and Africa	Asia and Pacific
www.energymicro.com/americas	www.energymicro.com/emea	www.energymicro.com/asia

Table of Contents

List of Figures

List of Tables

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

www.energymicro.com