EVE USB
Datasheet

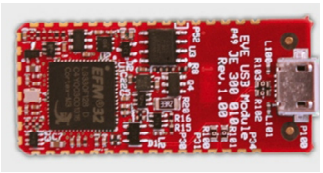
Main features

Software

- Micro-kernel with USB, scheduling, power and clock management
- Contiki OS
- Tickless design
- Drivers for peripherals
- USB stack:
  - OTG, host and device
  - CDC ACM and ECM
  - RNDIS for Android and Windows
  - ipheth for iPhone
  - http and JSON engine
- Remote software upgrade

Hardware

- 32-bits Cortex M3 microcontroller from Energy Micro
- 1 - 48MHz
- Standard version: 128k program FLASH, 32k RAM, 2M serial FLASH
- Extended version: 1024k program FLASH, 128k RAM, *TBD* serial FLASH
- 27 I/O including USB 2.0 OTG interface, ADC, DAC, SPI, UART, I²C, PWM, RTC etc.
- Built-in 3V voltage regulator and 5V step-up for USB host
- Extreme low power included voltage regulator:
  - Typical 1.3µA in power save mode
  - Typical 2.8µA with RTC



EVE USB – Actual size

Description

The EVE platform family is an integrated hardware and software platform solution engineered to provide a robust and flexible foundation for individual product development. Implementing EVE in new products fast-tracks the design process to a detail-design level and reduces the overall cost and time-to-market.

The EVE platform features standardized IO, common OS and drivers, ultra-low power consumption, homepage functionality based on USB and effective framework for homepage integration.

All EVE modules provide full remote operation capabilities from a web browser on a smartphone, tablet or PC via USB. System configuration, firmware upgrade, service functions and log and run data reading is easy to do from the homepage solution.

Applications

- Industrial control
- Security
- Measurement & data acquisition
- Residences and hotels
- Fire alarm systems
- Medical & Healthcare
- Telecom
- Ship & Offshore
- Yachts
- Recreation
- Consumer electronic

CONTENTS

1 INTRODUCTION

The EVE Platform Family is a robust, flexible and cost-effective technological platform for a wide range of electronic products and industries. EVE has been developed and engineered by industry professionals using proven technology and high quality components. All modules are fully tested and certified before delivery, eliminating the need for additional time or cost outlays.

EVE is designed for optimal integration and fulfills the common technical requirements for product development. All EVE modules are engineered with standardized IO, common OS and drivers, ultra-low power consumption, homepage functionality based on USB and effective framework for homepage integration.

The EVE Platform Family is a modular series built using the same core functionality and technology. EVE modules are differentiated by the communication carrier system they use.

The following features are identical for all EVE modules:

- Compact footprint (18x40mm)
- 27 IO (ADC, DAC, SPI, UART, I²C, PWM, RTC, USB etc.)
- Standard or extended version ultra-low power microcontroller
- OS, drivers and development environments
- Homepage integration and USB connection solutions

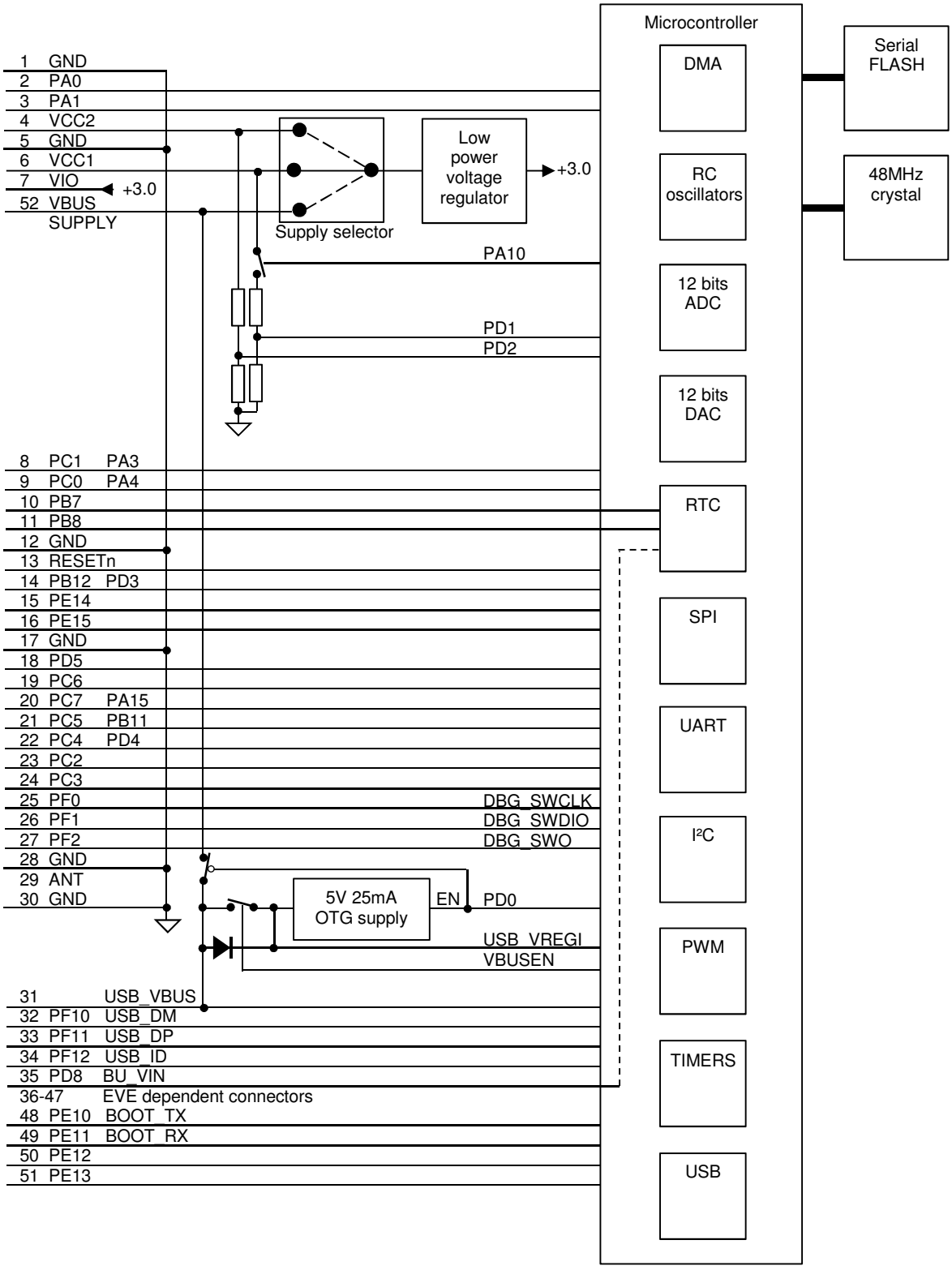
Note: Throughout this document, the EVE Platform Family is referred to as EVE. The EVE USB module is referred to as EVE USB.

2 EVE USB ELECTRICAL AND PHYSICAL CHARACTERISTICS

Parameter	Value
Supply voltage VCC1	3.1V – 5.25V
Supply voltage VCC2	3.6V – 5.25V
Supply voltage USB_VBUS	3.6V – 5.25V
Temperature range	-40℃ to +85℃
Current consumption (typical):	
• MCU active @ 1MHz	TBD
• MCU active @ 7MHz	TBD
• MCU active @ 14MHz	TBD
• MCU active @ 28MHz	TBD
• MCU active @ 48MHz	TBD
• Power down mode	1.5µA
• USB device in suspend	1.5µA
• USB device active (powered from VBUS, no traffic)	18mA
• USB device active (powered from VBUS, 1 Mbps data traffic)	20mA
• USB host active	23-30mA
• USB host idle (host cable connected, VBUS generated, device not connected)	1.5mA
Wakeup time (RC oscillator 1MHz, 7MHz, 14MHz, 28MHz)	TBD
Wakeup time (XTAL oscillator 48MHz)	TBD
IO output current capacity selection	0.5 mA 2 mA 6 mA 20 mA
Footprint	18 x 40 mm

3 BLOCK DIAGRAM

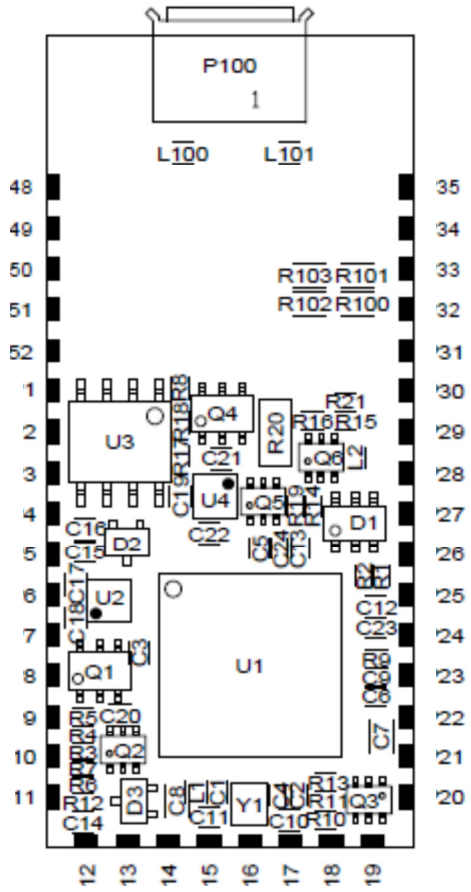
The block diagram shows edge connector number, power block, voltage monitoring, microcontroller and other components and wiring. The diagram is applicable to all EVEs.



4 IO DESCRIPTION

EVE modules are connected using SMD edge connectors. Most of the edge connectors are common to all EVE modules. EVE USB is not using any of the EVE dependent edge connectors 36-47.

4.1 EVE USB Pinout



4.2 Edge SMD connector

Some EVE edge connectors have duplicated microcontroller IOs. Normally only one of them is used at a time. Care must be taken to avoid parallel outputs. All EVE IOs are initialized disabled.

For more information about **Microcontroller pin function** and **Location** referenced in the table below, see the microcontroller datasheet.

The microcontroller pin functions in *italics* are available for internal use for some EVE. The EVE USB is not using these pins internally.

The table below shows the SMD connectors for EVE USB. Each group of **Microcontroller pin functions** has its own color code:

• Serial communications
• Analog functions
• Timer/counter functions
• PRS or sense functions
• XTAL or clocks
• EM4 functions, power or debug

EVE edge connector	EVE pin name	Microcontroller pin function	Location
1	GND		
2	PA0	TIM0_CC0	0/1/4
		LEU0_RX	4
		I2C0_SDA	0
		PRS_CH0	0
		GPIO_EM4WU0	---
3	PA1	TIM0_CC1	0/1
		I2C0_SCL	0
		CMU_CLK1	0
		PRS_CH1	0
4	VCC2		
5	GND		
6	VCC1		
7	VIO	+3.0V regulator output	
8	PC1	DAC0_OUT0ALT	1
		OPAMP_OUT0ALT	---
		ACMP0_CH1	---
		TIM0_CC2	4
		PCNT0_S1IN	2
		US1_RX	0
		I2C0_SCL	4
		LES_CH1	0
		PRS_CH3	0
	PA3	TIM0_CDTI0	0
		LES_ALTEX2	0
9	PC0	DAC0_OUT0ALT	0
		OPAMP_OUT0ALT	---
		ACMP0_CH0	---
		TIM0_CC1	4
		PCNT0_S0IN	2
		US1_TX	0
		I2C0_SDA	4
		LES_CH0	0
		PRS_CH2	0
	PA4	TIM0_CDTI1	0
		LES_ALTEX3	0

EVE edge connector	EVE pin name	Microcontroller pin function	Location
10	PB7	LFXTAL_P	---
		TIM1_CC0	3
		US1_CLK	0
11	PB8	LFXTAL_N	---
		TIM1_CC1	3
		US1_CS	0
12	GND		
13	RESETn	Active low	---
14	PB12	DAC0_OUT1	
		LETIM0_OUT1	1
		I2C1_SCL	1
	PD3	ADC0_CH3	---
		OPAMP_N2	
		TIM0_CC2	3
15	PE14	TIM3_CC0	0
		LEU0_TX	2
16	PE15	TIM3_CC1	0
		LEU0_RX	2
17	GND		
18	PD5	ADC0_CH5	---
		LEU0_RX	0
		OPAMP_OUT2	
19	PC6	ACMP0_CH6	
		LEU1_TX	0
		I2C0_SDA	2
		LES_CH6	0
20	PC7	ACMP0_CH7	
		LEU1_RX	0
		I2C0_SCL	2
		LES_CH7	0
	PA15	TIM3_CC2	0
21	PC5	OPAMP_N0	
		ACMP0_CH5	
		LETIM0_OUT1	3
		PCNT1_S1IN	0
		US2_CS	0
		I2C1_SCL	0
		LES_CH5	0
	PB11	DAC0_OUT0	
		OPAMP_OUT0	
		TIM1_CC2	3
		LETIM0_OUT0	1
		I2C1_SDA	1
22	PC4	OPAMP_P0	
		ACMP0_CH4	
		TIM0_CDTI2	4

EVE edge connector	EVE pin name	Microcontroller pin function	Location
		LETIM0_OUT0	3
		PCNT1_S0IN	0
		US2_CLK	0
		I2C1_SDA	0
		LES_CH4	0
	PD4	ADC0_CH4	
		OPAMP_P2	
		LEU0_TX	0
23	PC2	DAC0_OUT0ALT	2
		OPAMP_OUT0ALT	
		ACMP0_CH2	
		TIM0_CDTI0	4
		US2_TX	0
		LES_CH2	0
24	PC3	DAC0_OUT0ALT	3
		OPAMP_OUT0ALT	
		ACMP0_CH3	
		TIM0_CDTI1	4
		US2_RX	0
		LES_CH3	0
25	PF0	*TIM0_CC0*	*5*
		LETIM0_OUT0	*2*
		US1_CLK	*2*
		LEU0_TX	*3*
		I2C0_SDA	*5*
		DBG_SWCLK	0/1/2/3
26	PF1	*TIM0_CC1*	*5*
		LETIM0_OUT1	*2*
		US1_CS	*2*
		LEU0_RX	*3*
		I2C0_SCL	*5*
		DBG_SWDIO	0/1/2/3
		GPIO_EM4WU3	
27	PF2	TIM0_CC2	5
		LEU0_TX	4
		DBG_SWO	0
		GPIO_EM4WU4	
28	GND		
29	ANT	Reserved for antenna	
30	GND		
31		USB_VBUS	
32	PF10	USB_DM	
33	PF11	USB_DP	
34	PF12	USB_ID	
35	PD8	BU_VIN	
		CMU_CLK1	1

EVE edge connector	EVE pin name	Microcontroller pin function	Location
48	PE10	BOOT_TX	
		TIM1_CC0	1
49	PE11	BOOT_RX	
		TIM1_CC1	1
		LES_ALTEX5	0
50	PE12	TIM1_CC2	1
		US0_RX	3
		I2C0_SDA	*6*
		LES_ALTEX6	0
51	PE13	US0_TX	3
		I2C0_SCL	*6*
		LES_ALTEX7	0
		GPIO_EM4WU5	
52		VBUS_SUPPLY	

5 HARDWARE DESCRIPTION

The EVE platform family, including EVE USB, is engineered using a common hardware. The following information is applicable for all units of the EVE platform family. All EVE are manufactured in two versions: standard and extended.

5.1 Microcontroller

EVE modules are based on the EFM microcontroller from EnergyMicro.

	EVE USB Standard	EVE USB Extended
Microcontroller	EFM32LG330F128	EFM32GG330F1024
Program FLASH	128 kbytes	1024 kbytes
RAM	32 kbytes	128 kbytes
Serial flash	2 Mbytes	TBD

5.2 Serial FLASH

A part of the serial FLASH is used for holding home pages, firmware upgrades, log data etc. The rest of the serial FLASH is available for use by the application software.

5.3 USB

All EVE have USB OTG. Typically 25mA can be supplied to the USB connector in host mode. The USB data lines are overvoltage protected.

5.4 Power

5.4.1 Voltage regulator

All EVE have a 3.0V voltage regulator with typical current consumption less than 0.5µA in standby mode, and a 5V DC-DC converter for the USB.

5.4.2 Supply voltage selector

VCC2 and USB_VBUS are the primary voltage supplies. VCC1 is the secondary voltage supply, often used for battery. If both VCC2 and USB_VBUS are unavailable, VCC1 will automatically be selected.

5.4.3 Voltage monitoring

Supply sources VCC1 and VCC2 can be monitored based on measuring networks and ADC inputs. Each measuring networks sinks about 25-45µA depending of the supplied voltage. The measuring network for the VCC1 supply is normally turned off to save power in battery operated products, reducing the current consumption for the network to 0µA.

5.5 Power saving modes

The current consumption for an EVE depends on which energy saving mode and peripherals the microcontroller is using. The microcontroller has energy mode EM0-3. The OS will always select the lowest possible energy mode.

In EM3 all clocks are stopped while still keeping the RAM. In this mode the microcontroller uses about 100nA, and the EVE is typically consuming less than 2µA (including the voltage regulator).

6 FUNCTION DESCRIPTION

The following functional description is common across the EVE family.

6.1 Micro-kernel

The software shipped with the EVE implements a tiny and yet powerful micro-kernel, which provides basic system functionality such as:

- System clock, power and watchdog management
- Scheduling and delayed execution using workitems in µs and ms range
- Tickless system timer
- USB OTG
- USB device layer
- Drivers for EVE peripheral
- Built-in bootloader with in-circuit firmware upgrade functionality
- SWD debug print channel

6.2 OS

The EVEs are intended for Contiki OS environment. Contiki is a minimalistic OS, which has been designed, developed and maintained mainly by Adam Dunkels and Swedish Institute of Computer Science during the last 10 years under BSD license.

Contiki provides non-preemptive multitasking concept using protothreads, a type of lightweight stackless threads designed for severely memory constrained systems. Protothreads provides linear code execution for event-driven systems implemented in C.

In addition to multitasking and protothreads Contiki core provides synchronization primitives, event timers, static, pooled and dynamic memory allocation, as well as a lot of other primitives. EVE extends Contiki with functionality, provided by the micro-kernel, including USB stacks, microwork scheduling and tickless design.

Contiki provides support for IPv4 and IPv6 through uIP TCP/IP stack. EVE implements buffer management and retransmission primitives for the uIP, significantly increasing robustness of the stack. Shipped software includes implementation for USB network protocols such as RNDIS (device and host, typically used by Windows and Linux platforms) and ipheth (host, used by iPhone).

Contiki implements a file system (CoffeeFS), which can be rolled out on FLASH media with relatively small footprint. EVE provides external FLASH drivers and glue layer for the file system.

Contiki also provides a set of off-the-shelf applications which can be used as building blocks for products. The main focus within the set is communication protocols and stacks.

6.3 JSON engine

The EVEs proposes a lightweight integrated HTTP and JSON engine as a backend for the products homepage functionality. The engine serves directly two types of resources: static (typically files on the filesystem, firmware binary) and dynamic (in JSON format). The homepage itself (HTML/CSS/JS/graphics) is always static, dynamics is done by applying JSON data on top.

The advantage of the HTTP/JSON engine over classical server-based rendering is that it reduces resource requirements at the server side and thus reduces system cost. Static resources are put in serial FLASH memory. No MCU intervention is needed for processing the resources. All JSON data is handled by the same fixed-size code. Total cost of adding a new JSON variable to the system is as small as 16 byte in the MCU FLASH (plus the variable itself). Explicit distinguishing between data and representation allows to design quite complex homepage applications with very limited resources. This approach allows also updating (up to complete redesign) of the homepage without a modification of the MCU program (with no writes to the MCU internal FLASH).

The engine uses concept of JSON variable, a named entity, C variable, field in a structure, or just an abstraction like "command", exposed to the HTTP-based API. Variables can be grouped into structures, and the structures can be nested. The engine supports also arrays of variables. All variables are described in a table of content ("TOC table"), where variable properties like type, ranges and addresses are defined for each of them. Powered by C preprocessor, the TOC table definition generates a set of internal tables, used by the engine to access variables.

6.4 Web browser access to EVE

The EVE provides multicast DNS and LLMNR servers for addressing connected devices. The server resolves human-readable device addresses in .local zone (like "http://myproduct.local/") to the product IP addresses.

6.5 Bootloader

The EVE micro-kernel includes a bootloader which provides API for in-circuit reflashing of the MCU. The bootloader can operate in two modes:

- As a part of production routine the EVE can be programmed through an UART or an USB CDC class virtual UART. This functionality is provided by the bootloader itself.
- User can flash new software from home page interface via the HTTP protocol and JSON engine. This is a typical way of doing software updates.

6.6 Debug interface

The two-pin SWD interface is used for debugging, and SWO interface is used for debug prints and traces. This standard interface means that almost any SWD/SDO-compatible ICE debugger can be used with the EVE.

The EVE Reference board has a built-in SWD/SDO GDB-based ICE debugger with USB interface for debugging of the installed EVE. The debugger is supported by the EVE software development environment.

7 DRAWINGS

8 REVISION HISTORY

Revision:	Date:	Description
0.01	10. Oct. 2014	First preliminary release
0.02	21. Oct. 2014	Text updates

9 ORDERING INFORMATION
