

EVE Kernels User's Guide



Last updated: October 6, 2015

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components, which TI has specifically designated as military grade or "enhanced plastic", are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

| | |
|------------------------------|--|
| Audio | www.ti.com/audio |
| Amplifiers | amplifier.ti.com |
| Data Converters | dataconverter.ti.com |
| DLP® Products | www.dlp.com |
| DSP | dsp.ti.com |
| Clocks and Timers | www.ti.com/clocks |
| Interface | interface.ti.com |
| Logic | logic.ti.com |
| Power Mgmt | power.ti.com |
| Microcontrollers | microcontroller.ti.com |
| RFID | www.ti-rfid.com |
| OMAP Applications Processors | www.ti.com/omap |
| Wireless Connectivity | www.ti.com/wirelessconnectivity |

Applications

| | |
|-----------------------------|--|
| Automotive & Transportation | www.ti.com/automotive |
| Communications & Telecom | www.ti.com/communications |
| Computers & Peripherals | www.ti.com/computers |
| Consumer Electronics | www.ti.com/consumer-apps |
| Energy and Lighting | www.ti.com/energyapps |
| Industrial | www.ti.com/industrial |
| Medical | www.ti.com/medical |
| Security | www.ti.com/security |
| Space, Avionics & Defense | www.ti.com/space-avionics-defense |
| Video & Imaging | www.ti.com/video |

TI E2E Community e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright© 2013, Texas Instruments Incorporated

Read This First

About This Manual

This document describes how to install and work with Texas Instruments' (TI) kernels implementation on the EVE Subsystem. It also provides a detailed Application Programming Interface (API) reference and information of the kernels.

Intended Audience

This document is intended for system engineers who want to develop vision applications using EVE subsystem.

This document assumes that you are fluent in the C language, have a good working knowledge of embedded system and basic computer architecture concepts like DMA, Cache, interrupts etc.

How to Use This Manual

This document includes the following chapters:

- ❑ **Chapter 1 - Introduction** introduces the EVE subsystem and kernels components. It also provides an overview of supported features.
- ❑ **Chapter 2 - Installation Overview**, describes how to install, build, and run the kernels.
- ❑ **Chapter 3 – Computer Vision Library API Reference** describes the data structures and interface functions used for the Computer Vision Kernels.
- ❑ **Chapter 4 – Image and Signal Processing Library API Reference** describes the data structures and interface functions used for the Image and Signal Processing Kernels
- ❑ **Chapter 5 - Frequently Asked Questions**, provides answers to few frequently asked questions related to using the EVE kernels

Related Documentation From Texas Instruments

The following documents describe EVE subsystem details. To obtain a copy of any of these TI documents, visit the Texas Instruments website at www.ti.com.

- ❑ *Embedded Vector Engine (EVE) Programmer's Guide* (literature number SPRUHC1B) describes EVE architecture and all modules of EVE subsystem from programmer's view

- ❑ *EVE Subsystem Reference Guide* (literature number SPRUHF5A) describes the function description of EVE subsystem and its Register set
- ❑ *ARP32 CPU and Instruction Set* (literature number SPRUHC9) describes ARP32 architecture, Instruction Set and programming model
- ❑ Enhanced Direct Memory Access (EDMA3) Controller User's Guide (literature number SPRUEQ5) for complete details on the EDMA3Abbreviations
- ❑ Enhanced Direct Memory Access (EDMA3) Controller User's Guide (literature number SPRUEQ5) for complete details on the EDMA3Abbreviations
- ❑ VisionSurround28 Super/High/Mid Vision28 Super/High/Mid ADAS Applications Processor (SPRS884D)
- ❑ EVE Starter ware User's Guide
- ❑ EVE software Getting Started Document

The following abbreviations are used in this document.

Table 5-1. List of Abbreviations

| Abbreviation | Description |
|--------------|---|
| ARP32 | 32-bit Advanced RISC Processor |
| DMEM | Data Memory in EVE Sub System |
| EVE | Embedded Vision Engine |
| EDMA | Enhanced Direct Memory Access |
| IBUF | Image Buffer in EVE Sub System |
| IPC | Inter-Processor Communication |
| MMU | Memory Management Unit |
| RAM | Random Access Memory |
| SCTM | Sub System Counter and Timer Module |
| SMSET | Software Message and System Event Trace |
| VCOP | Vision Co-Processor |
| WBUF | Work Buffer in EVE Sub System |

Text Conventions

The following conventions are used in this document:

- ❑ Text inside back-quotes (“) represents pseudo-code.
- ❑ Program source code, function and macro names, parameters, and command line commands are shown in a `mono-spaced` font.

Product Support

When contacting TI for support on this eve kernels software component, quote the product name *EVE Subsystem Kernels* and version number. The version number of the product is included in the Release Notes that accompanies this product.

Trademarks

Code Composer Studio, DSP/BIOS, eXpressDSP, TMS320, EVE are trademarks of Texas Instruments.

All trademarks are the property of their respective owners.

DRAFT

This page is intentionally left blank

DRAFT

Contents

| | |
|---|------------|
| Read This First | 0-1 |
| About This Manual..... | 0-1 |
| Intended Audience | 0-1 |
| How to Use This Manual | 0-1 |
| Related Documentation From Texas Instruments..... | 0-1 |
| Text Conventions | 0-2 |
| Product Support | 0-3 |
| Trademarks | 0-3 |
| Contents | 0-5 |
| Figures | 0-6 |
| Tables | 0-7 |
| Introduction | 1-1 |
| 1.1 Overview of EVE Subsystem | 1-1 |
| 1.2 Overview of EVE Kernels..... | 1-2 |
| 1.3 Supported Services and Features..... | 1-2 |
| Installation Overview | 2-1 |
| 2.1 System Requirements..... | 2-1 |
| 2.1.1 Hardware | 2-1 |
| 2.1.2 Software | 2-1 |
| 2.2 Installing the Component | 2-1 |
| 2.3 Building EVE Kernels..... | 2-2 |
| Computer Vision Library API Reference..... | 3-4 |
| IMAGE and SIGNAL Processing Library API Reference..... | 4-6 |
| Frequently Asked Questions..... | 5-8 |
| 5.1 Release Package..... | 5-8 |
| 5.2 Code Build and Execution..... | 5-8 |
| 5.3 Issues with tools | 5-8 |

Figures

| | |
|--|----------|
| Figure 1-1. Block diagram of EVE Subsystem..... | 1 |
| Figure 2-1. Component Directory Structure | 2 |

DRAFT

Tables

| | |
|--|--|
| | |
| | |
| | |

| | |
|--|-----|
| Table 1-1. List of Abbreviations | 0-2 |
| Table 2-1. Component Directories | 2-2 |

DRAFT

This page is intentionally left blank

DRAFT

Introduction

The EVE subsystem consists of a vision co-processor (VCOP), several system modules including a mailbox, a cache controller, Memory Management Unit (MMU), an EDMA controller, and an interrupt controller, plus an ARP32 scalar processor that controls all of these modules. The EVE kernels package is an OS agnostic set of kernels (accelerated functions) that are available at ARP32 in “C” calling convention and use the VCOP of EVE to accelerate these.

This chapter provides brief overview of the hardware blocks in the EVE subsystem, and explains on EVE kernels offering. For detailed description, please refer to EVE Programmers Guide and EVE Subsystem Reference Guide. The purpose of this User's Guide is to provide detailed information regarding the software kernels provided with EVE software package and their interface.

1.1 Overview of EVE Subsystem

The Embedded Vision Engine (EVE) module is a programmable imaging and vision-processing engine, intended to be used in devices that serves consumer electronics imaging and vision applications.

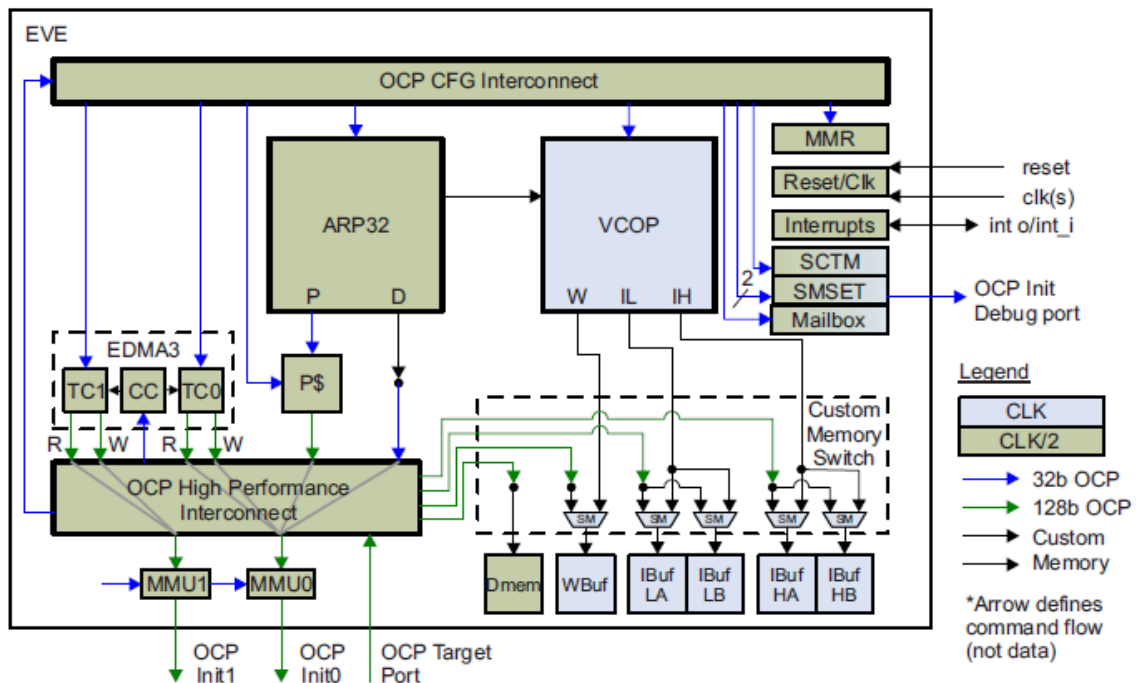


Figure 1-1. Block diagram of EVE Subsystem

Its programmability allows late-in-development or post-silicon processing requirements to be met, and allows third party or customers to add differentiating

features in imaging and vision products. The EVE module is instantiated in Vision super/High/Mid ADAS Application Processors

As shown in Figure 1-1 there are different programmable module in EVE subsystem like EDMA, Mailbox, SCTM, MMU, Program Cache, Interrupts, ARP32 and VCOP. VCOP is main compute engine for vector processing and ARP32 is the scalar processor. Other blocks are control and data transfer modules.

1.2 Overview of EVE Kernels

The EVE kernels package is set of OS agnostic C callable APIs that runs primarily on the ARP32 and utilize VCOP for acceleration. Primary purpose of this package is to abstract low-level details of VCOP and share a rich set of kernels for different applications spaces – vision, imaging and signal processing.

1.3 Supported Services and Features

This user guide accompanies EVE kernels example test code. This version of the product has the following supported features.

- ❑ Kernels for Vision applications – refer Product Data sheet for the list of supported functions/kernels. APIs for this category are described in this document
- ❑ Kernels for Imaging and signal processing applications - refer Product Data sheet for the list of supported functions/kernels. APIs for this category are described in this document

Installation Overview

This chapter provides a brief description on the system requirements and instructions for installing the EVE kernels component. It also provides information on building and running the example application.

2.1 System Requirements

This section describes the hardware and software requirements for the normal functioning for this component.

2.1.1 *Hardware*

EVE kernels has been built and tested on the EVE subsystem based devices like TDA1MEV /Vision28 Super (Vayu). In this document vme implies TDA1MEV and Vayu implies Vision28 Super.

2.1.2 *Software*

Please refer docs\eve_sw_getting_started.pdf document for the details.

2.2 Installing the Component

The EVE kernels component is released as a windows/unix installer along with other software modules for EVE. Figure 2-1 shows the sub-directories created after installation.

Note:

The source folders under drivers are not present in case of a library based (object code) release. In this document vme implies TDA1MEV and Vayu implies Vision28 Super.

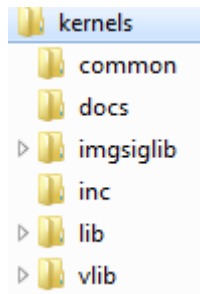


Figure 2-1. Component Directory Structure

Table 2-1 provides a description of the sub-directories created in the starterware directory.

Table 2-1. Component Directories

| Sub-Directory | Description |
|---------------|---|
| /docs | Contains Documents related to EVE kernels including this document |
| /common | Common files across multiple category of kernels – primarily common build related instruction and common profile/trace implementation related files |
| /vlib | Directory to contain computer vision related kernels |
| /imgsiglib | Directory to contain computer image and signal processing related kernels |
| /inc | Contains auto-generated header file for kernels |
| /lib | Contains debug and release pre-built library |

2.3 Building EVE Kernels

Please refer [eve_sw_getting_started.pdf](#) before reading this section to setup the build environment. This document is provided as part of the EVE software package.

The EVE kernels build expects pre build library of EVE starterware as `starterware\libs\vayu\eve\libevestarterware_eve.lib`. Please refer EVE starterware User's Guide to build this library

Kernels\makefile contains the make rules to build the kernels consumed by apps and BAM algorithms. This doesn't include the complete list of available kernels, one can add specific kernel to this make file to get included in the generated library

To build these one need to call

gmake

To clean these one need to call

gmake clean

This will produce libevekernels.eve.lib at Kernels\Lib directory

To build individual kernel as part of vlib and imsglib directories, one need to use the makefile as part of the specific kernel. If test folder is present inside kernel then go to test folder for building the kernel. If it is not present go to the src_C or src_kernelC folder for building natural C version or kernel C version respectively. For example to build vlib\vcop_bhattacharyaDistance, one needs to be at below directory level

vlib\vcop_bhattacharyaDistance\src_kernelC

and use below commands to build :

For target build :

gmake target

For host build using g++ compiler :

gmake

For host build using msvc(visual studio) compiler :

gmake msvc

For building kernel for target in debug mode :

gmake debug

For building kernel with profile points enable :

gmake profile

To clean :

gmake clean

By default host emulation builds are in debug mode. It is important to note that, not all kernels supports host builds. Hence for kernels for which host build is not supported user can directly use following command for building it in target :

gmake

gmake clean

Chapter 3

Computer Vision Library API Reference

This chapter provides a detailed description of various APIs of Computer Vision Kernels. Please refer below document for details

[Vision Library API Guide](#)

Note:

Currently Computer Vision Library API is a separate document. It will get merged with this document

This page is intentionally left blank

DRAFT

IMAGE and SIGNAL Processing Library API Reference

This chapter provides a detailed description of various APIs of Image and Signal Processing Kernels. Please refer below document for details

[Image and Signal Processing Library API Guide](#)

Note:

Currently Computer Vision Library API is a separate document. It will get merged with this document

This page is intentionally left blank

DRAFT

Frequently Asked Questions

This chapter provides answers to few frequently asked questions related to using this product.

5.1 Release Package

| Question | Answer |
|--|--|
| Can this product release be used on any eve-based platform? | Yes, you can use it. |
| Does it use one EVE or multiple EVE | This product only utilizes one EVE though any user of this product can build an multi-EVE application with different functionalities (kernels) executing on different EVEs |
| Why does this product is dependent on EVE starter ware product | The test code of the functions use SCTM for profiling and that functionality is available in starter ware component – as such just the kernels doesn't have any dependency on starter ware |

5.2 Code Build and Execution

| Question | Answer |
|---|--|
| Build is not working and not giving any error | Ensure that you have set proper environment variables as described EVE Getting started document |
| Build is not working on windows, giving error "gmake: *** No rule to make target" | Ensure that you have set proper environment variables as described in section 2.3. Make sure that your EVE_SW_ROOT path is not too long. If this path is too long in windows, build can fail. Other way to fix this problem is to create a link to this path. For this you can use subst utility in windows. This will create a new drive with the content of you directory. subst <DRIVE NAME> PATH e.g. subst X: \$(EVE_SW_ROOT) |
| My build is not happening and requesting for libevestarterware_eve.lib | Please refer section 2.3, the build has dependency in EVE starterware |

5.3 Issues with tools

| Question | Answer |
|----------|--------|
|----------|--------|