

# Blackfin® USB-LAN EZ-Extender Board Support Package (BSP) v1.0.0 Release Notes

Thank you for installing the USB-LAN EZ-Extender Board Support Package (BSP). The BSP provides software and documentation in support of the USB-LAN EZ-Extender daughter board. This daughter board is used in conjunction with [ADSP-BF533 EZ-KIT® Lite](#), [ADSP-BF537 EZ-KIT® Lite](#), [ADSP-BF561 EZ-KIT® Lite](#).

The USB-LAN EZ-Extender is designed for use with CrossCore® Embedded Studio (CCES) for Analog Devices Processors software development tools. The CCES development environment aids advanced application code development and debug, such as:

- Create, compile, assemble, and link application programs written in C++, C, and assembly
- Load, run, step, halt, and set breakpoints in application programs
- Read and write data and program memory
- Read and write core and peripheral registers
- Plot memory

For more details on CCES, please visit [www.analog.com/cces](http://www.analog.com/cces). For more on the USB-LAN EZ-Extender, please visit <http://www.analog.com/en/evaluation/bf-extenderul/eb.html>

The USB-LAN EZ-Extender BSP provides comprehensive software support for the USB-LAN EZ-Extender daughter board. Specifically, drivers and examples are included for the following components:

- SMSC LAN91c111 Ethernet controller
- Net2272 USB Controller

The BSP also provides comprehensive examples which demonstrate the services. TCP/IP examples based on USB-LAN EZ-Extender are available in the lwIP product which uses this BSP. USB Vendor Class based examples are present inside the BSP itself which demonstrates the data transfer over Bulk endpoints.

The CCES Help environment provides complete hardware and software documentation.

## License Checking

There are no license requirements for the USB-LAN EZ-Extender BSP.

## Installation Logging

The installer does not create a log file by default. If you encounter installation issues, you can generate an installation log file by running the installer from the command prompt.

Change to the directory containing downloaded installer executable and run the following from the command prompt:

```
ADI_USBLAN_EI1_EZExtender-Rel1.0.0.exe /v"/1*v c:\temp\installer.log"
```

## Support and Assistance

There are several options for contacting support:

- Submit your questions online at:
  - <http://www.analog.com/support>
- E-mail your Processor and DSP software and development tools questions from within CrossCore Embedded Studio:
  - Go to “Help->E-mail Support...”. This will create a new e-mail addressed to [processor.tools.support@analog.com](mailto:processor.tools.support@analog.com), and will automatically attach your CrossCore Embedded Studio version information (ProductInfo.html).
- E-mail your Processors and DSP applications and processor questions to:
  - [processor.support@analog.com](mailto:processor.support@analog.com) OR
  - [processor.china@analog.com](mailto:processor.china@analog.com) (Greater China support)
- Post your questions in the Processors and DSP online technical support community in Engineer Zone at:
  - <http://ez.analog.com/community/dsp>

## Software Requirements

To build the example projects included in the USB-LAN EZ-Extender BSP CrossCore Embedded Studio Release 1.0.1 or later is required. For TCP/IP examples using USB-LAN EZ-Extender lwIP (Lightweight TCP/IP Stack) Rel 1.0.1 or later is required.

## Test Configurations

The products and versions used to test the BSP are

1. CrossCore® Embedded Studio Release 1.0.1
2. lwIP for CorssCore Embedded Studio Rel 1.0.1
3. USB-LAN EZ-EXTENDER Card Revision 2.1

### Adding a Driver to a Project

When adding "Net2272 USB Device Controller Driver" add-in, the IDE will add the sources for the driver to the CCES Project folders, starting at "system". There will be a folder specific to the driver(s) or service(s) you have added under this folder.

When adding SMSC LAN91c111 Ethernet module the IDE will add the required include paths and pre-built binaries to your project.

### **Creating a project which includes a USB-LAN EZ-Extender drivers**

In order to create a project you should follow the instructions provided in the CrossCore® Embedded Studio help. As part of the project creation, the page "Add-in selection" contains a list of all the available add-ins for the project that you are creating based on the installed products and the project's chosen processor and type. You can see the drivers in support of the USB-LAN EZ-Extender under the "Device Drivers and System Services" category. Within this category you will see "USB LAN E11 Extender Board Drivers" which contains the drivers for "Net2272" and you will also see "Etherent" which contains the SMSC LAN91c111 driver binary.

The USB-LAN EZ-Extender add-in generates a call to `adi_initComponents()`. For more information on `adi_initComponents()`, please refer to the CCES help section:

CrossCore Embedded Studio Release 1.0.1 > Graphical Development Environment > System Configuration

### **Adding a driver to an existing project**

Every CrossCore Embedded Studio project contains a System Configuration file called `system.svc` which is located in the root of the project. The file is the IDE's interface for managing the various pre-written software components used in the "system" implemented by a project. Double-clicking any `system.svc` file in a navigation view opens that file in the System Configuration Utility which allows you to see the add-ins that you currently have in your project. Click on "Add..." and select the "USB-LAN E11 Extender Board Drivers" add-in which is under the "Device Drivers and System Services". For adding USB driver select "NET2272 USB Device Controller driver" under "USB". For adding "LAN91c111" Ethernet binaries select "SMSC LAN91c111 Etherent Module".

#### **Notes:**

- If the IDE detects that `adi_initComponents()` is not yet present in `main()`, it prompts you to add it and offers to insert it for you.

### **Configuration**

There are no USB-LAN EZ-Extender driver configuration options available in the IDE.

## Interrupts

CrossCore Embedded Studio provides a coherent interrupt management mechanism which allows for the same interface to be used in RTOS and non-RTOS applications. This means that interrupt service routines in all applications must be written in C and use the `adi_int` interface. Any thread-safety requirements or interactions with tasks are handled by the `adi_int` interface. For more information on the `adi_int` API, in CrossCore Embedded Studio go to Help > Search and enter `adi_int`.

Examples of the usage of this interrupt management mechanism are the System Services and Device Drivers provided with Crosscore Embedded Studio. By using the `adi_int` interface, the same services and drivers can be used in all applications regardless of whether an operating system is used.

## Sketches and Examples

### Sketches

None

### Examples

The USB-LAN EZ-Extender BSP provides the following examples.

1. USB Vendor bulk example for ADSP-BF533 and ADSP-BF561.

Note: TCP/IP examples based on LAN91c111 driver are provided in lwIP (Lightweight TCP/IP Protocol Stack) Release 1.0.1 or later.

### Location

In order to locate the USB-LAN EZ-Extender examples you can use the following:

- Open CrossCore Embedded Studio's Example Browser which can be found in CrossCore Embedded Studio under Help. Select in the Product section "USB-LAN EZ-Extender" for a full list of examples.
- Import projects located in your USB-LAN EZ-Extender BSP installation folder under the example directory in product installation  
(`<USB_LAN_EZ_Extender_installation_root>\BF_USB-LAN_EZ-Extender\Blackfin\Examples`).

## Documentation

API documentation for the drivers included in the USB-LAN EZ-Extender BSP can be found in CCES Help under *Blackfin® USB-LAN EZ-Extender® Board Support Package 1.0.0*.

General information on the driver model can be found in CCES help under

CrossCore® Embedded Studio 1.0.1 > System Runtime Documentation > System Services and Device Drivers

## **MISRA-C Support**

MISRA C is a software development standard for the C programming language developed by the Motor Industry Software Reliability Association (MISRA). Its aims are to facilitate code safety, portability, and reliability in the context of embedded systems, specifically those systems programmed in ANSI C. The compiler detects violations of the MISRA rules at compile-time, link-time, and run-time.

## **System Services and Device Driver Thread Safety**

All system services and device drivers (SSLDD) use mutexes and semaphores to ensure thread-safety. If an RTOS is present then the SSLDD will use the RTOS mutex and semaphores. If an RTOS is not present then the SSLDD will use a non-RTOS implementation of mutexes and semaphores (spin locks).

### **Known issues:**

TAR-49871: On ADSP-BF561 platform VendorBulkDemoBF561 example may crash if heap is placed on L1. By default heap is placed on L2 memory.