

# Apex Embedded Systems

## Linux Library and Driver Brief Summary

### Contents

Apex Embedded Systems.....	1
Linux Library and Driver Brief Summary .....	1
Overview .....	2
Organization.....	2
Apex Immediate To Do Items .....	3
Apex Longer Term To Do Items.....	3

### Notice

The Apex Embedded Systems Linux Library and Driver code base is under continuous improvement, especially in this initial release. If you find a bug **please** let us know. If you would like to see something changed or improved, **please** let us know and we will consider it. We are looking to improve this code base and simplify its use where-ever possible. Contact via email at:

[mikeihm@apexembedded.com](mailto:mikeihm@apexembedded.com).

## Overview

The Apex Universal Library Driver Suite is an ongoing effort to upgrade and improve the code base to support our standard products.

Please note that this code base will be going through many updates and enhancements. It is a merge of several existing code bases which will help us produce better code and support, in the long run. During this initial release, since there is so much code, if you find a bug let us know and we will fix it immediately.

There are several goals we are working to achieve:

1. Provide a framework such that Apex has the ability to continue developing the library and driver with minimal impact on user applications. This also means that software components may be moved between application and kernel module as they mature.
2. Provide maximum feedback and diagnostics to customers in terms of where the library, driver or hardware might be having issues.

At the moment it is highly discouraged to use the driver without the library. The library provides the bulk of the logic for verifying configuration and reporting human readable error messages and so on.

## Organization

The aes folder is composed of a source tree (aul\_tree) and several application and driver folders. All driver related folders have a “drv\_” prefix. Most all the demo folders have a “demo\_” prefix.

There are two kernel-modules (drivers) available: “drv\_aul\_sim” and “drv\_aul\_x86”.

The “drv\_aul\_sim” is a driver with a very simple I/O simulator. The simulator emulates I/O by simply using an array of memory; it is not a robust emulator of the I/O space. The purpose of the simulator is primarily to test library and applications within an environment that does not have the

hardware available (i.e. desktop). Please read the README.txt file for more information on how to compile and install.

The “drv\_aul\_x86” is a driver intended for hardware running on an x86 platform. Please read the README.txt file for more information.

There is a GUI written in C# located in “aes/aul\_tree/\_Demo\_Csharp”. Please read the Quick\_Start\_Linux.pdf for more details.

Finally, there is a console based application in “aes/aul\_tree/AppTestDLL”. Moving into directory “aes/aul\_tree/\_Out\_Executable\_AulTest\_Debug” one can use “make clean” and “make all” to build this executable. This executable will produce reports and header files that are useful for user application development. The executable produced is “aul\_tree”.

There is a rough Doxygen configuration file that can be used to generate documentation, but it still needs work.

### **Apex Immediate To Do Items**

1. Complete Tracer-E and Summit support
2. Continue code review and testing, fixing any bugs that are found.
3. Complete API documentation. All STX104 API documentation can be found in the Windows Library/Driver PDF.
4. Provide users with Eclipse support.
5. Complete Doxygen API documentation.
6. Support for interrupts

### **Apex Longer Term To Do Items**

1. Provide support for ARM targets.