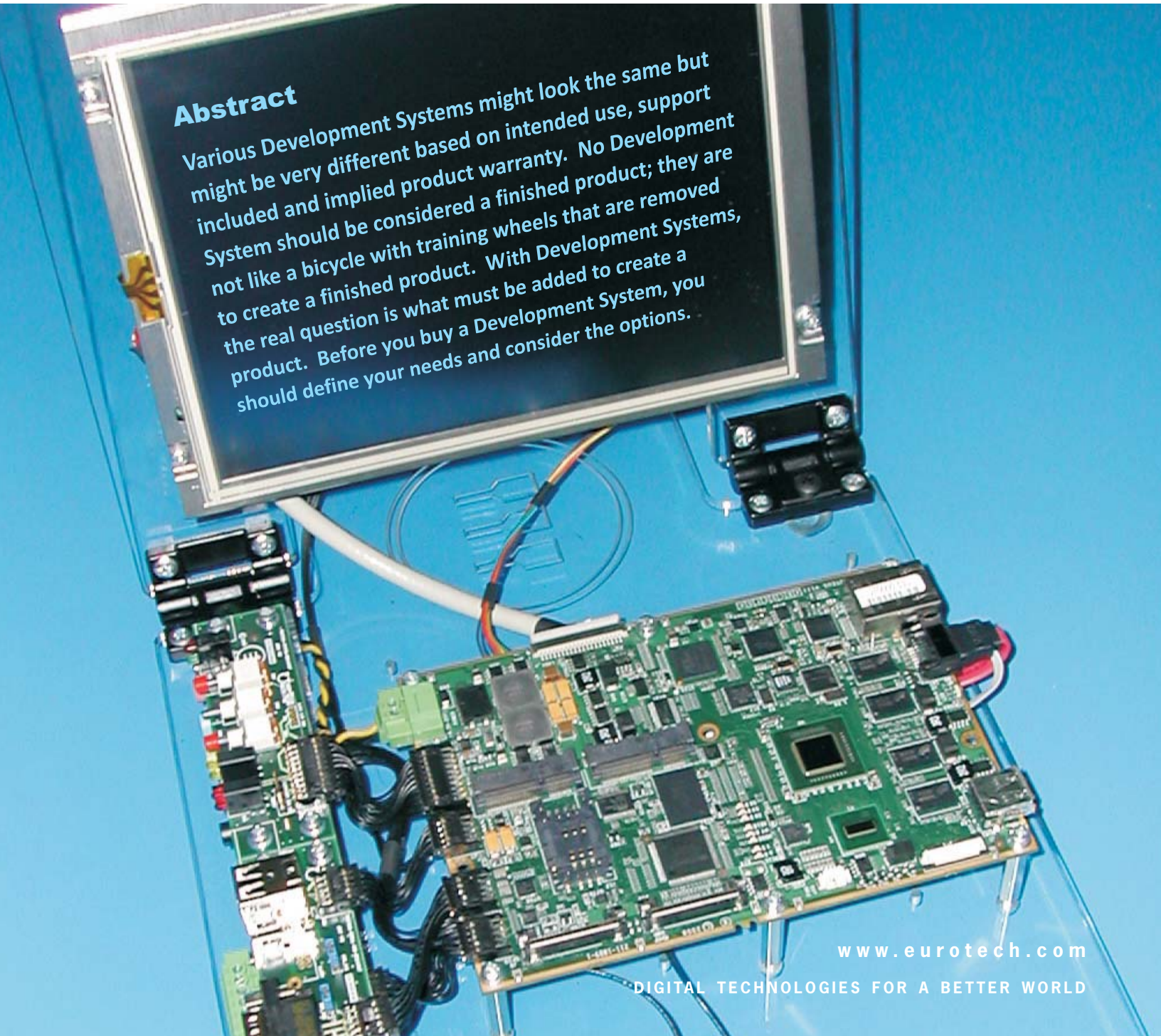
EUROTECH

Choosing an Embedded Development System

A Development Environment Should be measured against Objectives

Whitepaper

By: Lawrence Ricci



Abstract

Various Development Systems might look the same but might be very different based on intended use, support included and implied product warranty. No Development System should be considered a finished product; they are not like a bicycle with training wheels that are removed to create a finished product. With Development Systems, the real question is what must be added to create a product. Before you buy a Development System, you should define your needs and consider the options.

Silicon Developer Kits

These are boards designed by the silicon vendor and are typically sold early in the CPU lifecycle, either from the silicon supplier, or his representative organization. Most often, you can not buy this type of Development System without a NDA and perhaps a contract with the Silicon Supplier. In general, hardware is designed by the Silicon vendor, but software is often subcontracted out to an ISV (Independent Software Vendor) who unbundles source code and offers support under service contracts. Participation in Silicon Development Kit programs is in fact a major way these ISVs find business. Overall project support is typically supplied by the sourcing ISV staff or specially trained FAEs from the silicon vendor's rep or distributor organization. Since silicon revs and updates are frequent early in the cycle, factory support is often involved. Also, since the silicon supplier is trying for very large design wins, support is often free or low cost.

Silicon Developer Kits are best used by early adopters, companies that plan to develop their own intellectual property around the silicon/OS union. The good news is these platforms are available early in the silicon lifecycle. The bad news is that, because they are quick to market efforts and centered on the silicon, not an application, they seldom are an optimized design and often have outright errors.

The software ports associated with these, if any, are typically subcontracted rush jobs done on the cheap with only the initial silicon documentation and are, again, not optimized and tend to be filled with errors. The initial port for these is often done by a third party who sells services to later optimize the BSP, so there is little motivation to provide a perfect, optimized software build.

Figure 1- the *Intel Mainstone* is typical of Silicon supplier dev systems- notice, this was targeted at PDAs and smart phones, but look how many components!!!
This is not a refined bill of material.

The principle advantage of Silicon Developer Kits is that bill of material, firmware, and software BSP are available, as part of Development System price. They are also available early in the release cycle of the Silicon.

Such a Silicon Developer Kit is the right starting place for a company like Eurotech, who develops optimized designs for its products and its OEMs products. History shows that Eurotech (and probably anyone else) will spend two to four man years from the starting point of one of these Silicon Developer Kits to the point where efficient design suitable for volume manufacturing is placed in the market.

Educational and Hobbyist Development Kits

These are low cost boards typically supplied with 'community' support. Several firms specialize in these very low cost offers. Sometimes these designs are done by academic institutions and put in the public domain, sometimes they are done by silicon or board manufactures who want to sell products to the education market. Educational development systems are ubiquitous and cheep. They are purchased with a credit card off the web, they are given away by distributors, are donated to educational institutions to run classes and contests and are often rolled into the admission cost of technical conference and training programs. The designs, while typically constrained in features, can be quite good from a hardware point of view, but the software support shipped with the kit is typically pretty basic, a Linux root prompt or maybe a minimum build of Windows Embedded Compact Edition. The support structure for these systems is almost always community based: forums and newsgroups. Code contributions to these are typical 'community' contributions- some is excellent, some is terrible. None of it is screened in a certifiable manner for security vulnerability, backdoors, license and patent violations, and other IP restrictions. There is no single point for tough hardware/software problem resolution; the buyer must do this himself.

Every so often, these educational Development Systems are put into products, and with enough software work they can do a good job. The real problem is that since these educational Development Systems are cost engineered, often with prices based on concession silicon pricing and single run, high volume production, these boards many not offer the typical OEM a stable source of supply. Where OEM products typically demand a seven to ten year lifetime, most of these educational boards are products in a campaign of a few thousand, sold out, and then produced in the next, updated, 'better' version. This is quite different than the supply chain of a serious OEM who needs to buy a certain number of exactly the same thing every month for a decade.

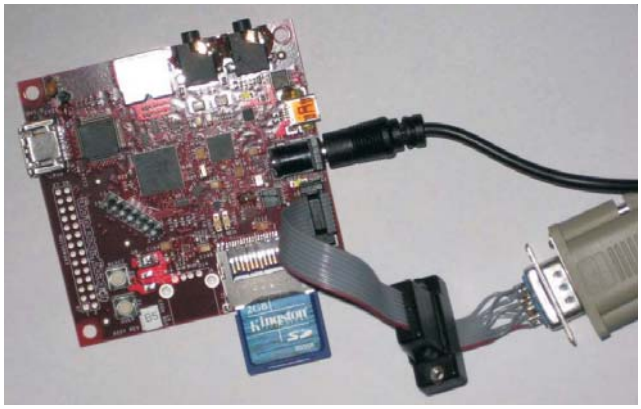


Figure 2- BeagleBoard is one of the suppliers serving the educational community. Code can be found at BeagleBoard.org. Photo credits to http://pmeerw.dyndns.org/blog/2008/Aug.

Perhaps the best thing a serious OEM can do before incorporating one of these designs it to ask the supplier for a fixed price for a five or ten year supply agreement. Likewise, since the software comes from the 'community' the OEM might want to consider some form of indemnification or at least investigate the particulars of the license agreements he is adopting. There are companies, like Black Duck Software, that will review open source code and advise on the various licenses and IP included.

Product Development Kits

These are Development Kits based on production grade hardware and software, used in real products in volume. Most often these Product Development Kits include industry specific features and interfaces, with support both at the driver and applications level. They are typically available from companies who hope that the customer will continue to purchase similar boards from them, or perhaps a license and design.

Figure 3- This Development System is based on a 'module' and 'carrier' board. Software drivers are in place, and the carrier board schematics are available to the customer to facilitate the design of his own product.

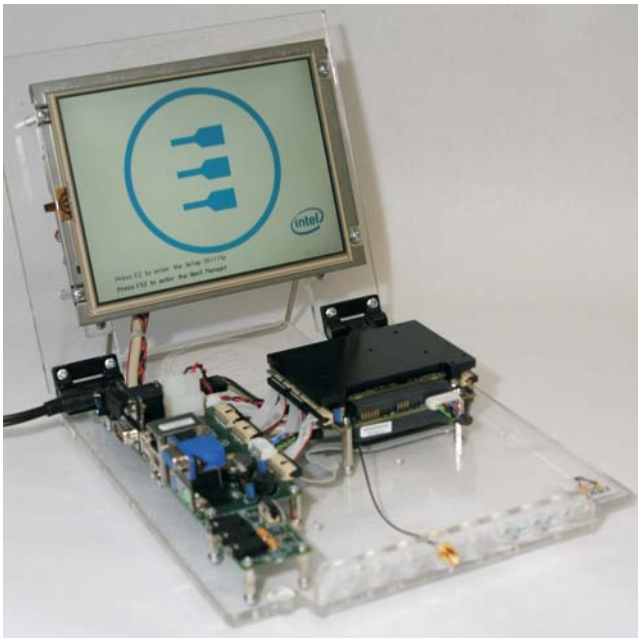


Figure 4- This Product Development System is for a PC/104 card. Even so, the system is wisely equipped with a substantial 'breakout board' to allow the engineer easy access to the many on-board features and signals he may want to employ in his product.

Support for a Product Development Kit is typically from the supplier of the board. This means that there is a single point for resolution for hardware/software support issues. This support may be free or may be part of a contract with the supplier.

Often, Product Development Kits are based on 'modules' with the hard to engineer CPU related components on a supplier's plug in board (Standard or non-standard connections). The OEM can then plug the suppliers module into his own 'carrier board' which contains the bulk of the products electronics, power supplies, etc. This gives the OEM the ability to engineer and manufacture the portions of his product that are industry and application specific.

Summary

Like bicycles, Development Systems come in many forms. Going from the Development System to a finished product is not a matter of removing training wheels, it requires value add from the engineer and his company. Carefully examine your starting point and destination. The Development System is an important first step.