



Figure 1 | Customized COM Express based embedded vision system for a medical application

Customized Embedded Vision Solutions

In the high volume consumer markets, we see more and more intelligent embedded vision in phones, cameras, cars and so on, but the focus of this article is on practical solutions for building customized embedded vision systems for machine vision and related industries, in the low to medium volume range – anything from tens to thousands per year.

So what is meant by “customized embedded vision solutions”? Embedded systems are characterized by several factors – the key ones being compactness, high reliability and long service life. To provide an overall “solution”, other areas of functionality as well as vision, would typically be integrated - for example machine control, communications, networking and other custom input/output (I/O) functions. It is worth mentioning that a key benefit with the custom integration of GigE / USB3 Vision is that a known good chipset can be used and integrated in such a way so as to maximise bus bandwidth. The result is a perfect fit for a specific application, well suited to the requirement of long service life and high reliability. A good example would be the embedded systems to run complex machines in the medical field – for example X-ray, CT scanners, and cancer treatment machines. These types of medical machines use a variety of imaging technologies and are designed to last many years. Not only that, the regulatory approval process may take years, so there is no desire to change anything once approved and the same embedded unit would be required for purchase for many years. In machine vision, the same can apply to areas such as pharmaceutical

packaging, quality control and food processing – areas where there is regulatory control and the requirement for high reliability. The alternative to a customized embedded vision system would typically be a standard or mini PC with various cards plugged in to provide all the necessary functionality. This approach is fine for some applications and certainly well suited to low volume, but where there are size constraints and long product life-time requirements, after a regulatory approval process, then the off-the-shelf PC route may not be appropriate. Even standards such as Mini-ITX, that preserves the same basic layout, still have attributes that change on a regular basis resulting in potential problems in type approved systems.

Formats and Operating Systems

Embedded formats such as PCI/104-Express and CompactPCI offer a good solution to some applications, but in a volume, size constrained, long-life application, the best solution is to integrate all the required functionality onto a single customized PCB, designed to be the appropriate size with connector positioning laid out in an ideal manner for the target application.



Figure 2 | ARM based embedded vision system for an industrial machine application

This custom PCB approach fits well when the volumes are of the order of one hundred systems per year and upwards. At this level up to several thousand per year, it generally makes sense to source the processor module as an off-the-shelf component, but using a mezzanine format designed for reliability and long-life product life such as COM Express. Processor modules in this form-factor typically have a product supply lifetime of five years; however it is a relatively simple task to just replace the processor module while keeping everything else identical. This provides a near identical fit, form and function. Of course at the design outset, all components need to be chosen in order to maximise lifetime, but by controlling the manufacturing, component obsolescence can be mitigated through careful supply chain management. Software-wise the environment to the application developer can be made to appear identical to a standard desktop machine running Windows or Linux. Even if the target processor is not an Intel device – for example the ARM family, then there are still options to make the embedded device appear the same as an Intel/Windows machine for development, and then ARM/Linux for deployment. Regarding the operating system (OS), the choice is often between Windows Embedded and Linux, but sometimes QNX and others. OS independent code can be useful here too – for example development under Windows but deployment under Linux to save costs in volume. Typically the operating system's file system is configured to be read only, since power is usually removed from the embedded system without any forewarning. For systems requiring read/write functionality, such as for recording, a separate read/write drive is typically used and may require the use of an integrated short-term power supply backup – only for a few seconds, to prevent file system corruption if power is removed part-way through a write cycle to disk.

Application examples

To provide some real-world examples, two embedded vision systems, designed and manufactured by Active Silicon are described below. Figure 1 shows a custom embedded system that uses GigE Vision and uses the COM Express format for the processor module. The unit runs off a single 12V supply and integrates various I/O functions required to control the medical machine into which it is fitted. This particular unit, which runs Windows Embedded, went into medium volume production from 2008 with an anticipated product life of 10-15 years for new sales and a further 10 years for service life, so a total anticipated lifetime of 25 years. The development cycle for this type of customized embedded system is typically six to nine months. Figure 2 illustrates an extended mezzanine approach used to achieve rapid time to market in this three sensor vision application, which is integrated into an industrial machine. The custom PCB defines the size and connector layout, while a number of off-the-shelf modules for I/O and motor control provide the required functionality. This rapid prototyping approach gives the customer systems to work with while for volume manufacture, several of the off-the-shelf modules can be designed into the main board. The processor in this example system is a 32 bit ARM based device running Linux, but development work may also progress on the unit under Windows on a standard PC when connected via a USB port.

www.activesilicon.com

Author | Colin Pearce, Managing Director, Active Silicon Ltd.

Article published in inVISION, Issue 2, April 2015, p. 34-35. (German). Distribution of the English translation by Active Silicon with kind permission of inVISION.