Integrating Image Processing into Embedded Systems

The Key for New Product Developments

Embedded computer systems are interesting for a variety of applications and enable the development of new products. Many users can also employ embedded systems as a more efficient approach to their existing systems which can be a portal to savings.

Those interested in using embedded image processing technology in their systems must answer one important question from the start: How will the camera be integrated into the system? In industrial image processing, camera integration is typically handled by connecting camera and PC (or IPC) via a GigE or USB interface — in essence a plug-and-play solution. Add in a manufacturer's Software Development Kit (SDK) and accessing the camera becomes child's play. Does the same principle apply to an embedded system? The short answer: yes! The process remains the same for a Single Board Computer (SBC): popular SBCs such as Raspberry Pi and Odroid, as well as industrial-strength modules like Thoradex and Advantech offer comparable interfaces (USB/Ethernet). While SBCs with x86-based processors are on the market, ARM-based processors are more common. A growing number of camera manufacturers offer their SDKs in an ARM version, allowing users to integrate a camera using the same familiar method as on a Windows PC. Ideally the SDK would offer the same functions and APIs (Application Programming Interfaces) for both platforms. This allows programming code to be reused, keeping integration time and effort to a minimum.
Components of an industrial image processing system

Components of an embedded image processing system

**Specialized embedded systems**
Embedded systems can also be based on much more specialized hardware. Many of these systems use SoMs (Systems on Module), which are ultra-compact board computer modules comprised only of a processor (typically an SoC or System on Chip), RAM and other key components.
These SoMs are mounted on a circuit board that provides the necessary connections with the outer world. Because they are relatively affordable, these circuit boards allow for the systems to be adjusted individually for specific requirements. Even so, the use of a standard SoM helps keep total system costs low. This configuration can also be outfitted with a standard interface port (such as USB). In that case, the user achieves the same benefits as offered by an SBC. The USB interface is not always optimal from a weight, space or energy standpoint. In that case, a direct camera/processor connection is a potential alternative.
**Special image transfer**
Stable and robust: Image data transfer on a LVDS-basis or via the CSI2 standard is well suited for the direct connection between the camera and SoC. For both methods, however, there is no specific standardized hardware, meaning no specified connections or cables. As a result, these must be defined separately for the connection of a camera onto the circuit board.

CSI2, a standard from the mobile electronics industry, defines the signal transfer and a software protocol standard. Some SoCs offer CSI interfaces, although they don't work uniformly, and there is no generic driver. As such, CSI2 doesn't represent an immediate "out of the box" solution.

At the same time, there is also no standard SW protocol or standard driver for LVDS, in spite of its wide availability as a procedure for high-speed data transfer. Several manufacturers try to improve the situation through cameras with LVDS outputs based on proprietary protocols and processing chips with the matching drivers. The benefit is a complete solution with low integration costs. However, the user is tied to a specific set of hardware.

Other camera manufacturers offer a documented, LVDS-based camera output for hardware integration. This allows a driver or read-out routine to be created, or existing drivers can be adapted for the user's system. Existing reference systems or driver templates are a major help during development.

**Camera configuration**
Another important question for camera integration is the camera configuration. Control signals between the SoC and the camera can be passed via bus systems such as CAN, SPI or I2C. No standard has yet been defined for this function. The camera manufacturer determines which image processing parameters can be controlled and how, and whether, for example, GenICam will be supported. When the right driver is in place, the software can access the camera configuration directly and adjust parameters. Good documentation of the software protocol helps take advantage of the full range of camera configuration options. Basler supports access to cameras via I²C (as part of the BCON for LVDS interface) from within Basler's pylon SDK and offers standardized APIs (in C++) for configuration programming.
Basler dart board level cameras with USB 3.0 and LVDS-based BCON interfaces

Potential interfaces for embedded image processing: BCON for LVDS and USB 3.0

Summary
Embedded image processing can be the ideal solution for specific applications. Embedded applications can be developed for SBCs featuring GigE or USB connections to the camera. This hardware is popular, available in many price, performance and quality ranges and suitable for a variety of application cases. LVDS and CSI2-based camera-to-SoC connections offer a direct interface for image data transfer, but without the benefit of a comprehensive industry standard. As a result, both methods involve adjustment and integration time and costs when developing simple integration solutions. Companies offer drivers, development kits and other tools to help make integration easier for users. In the future, the development and roll-out of standards would be a welcome step towards simpler integration of cameras involving a direct connection to an SoC. Generic drivers and standardized data APIs can also make it easier to develop specialized image processing solutions. As such, the integration of image processing technology can be made as simple as the prevalent (I)PC-based industrial image processing, even for the smallest and leanest embedded systems.
Autor(en)

Dr. Thomas Rademacher, Product Market Manager - Factory & Traffic

Kontaktieren