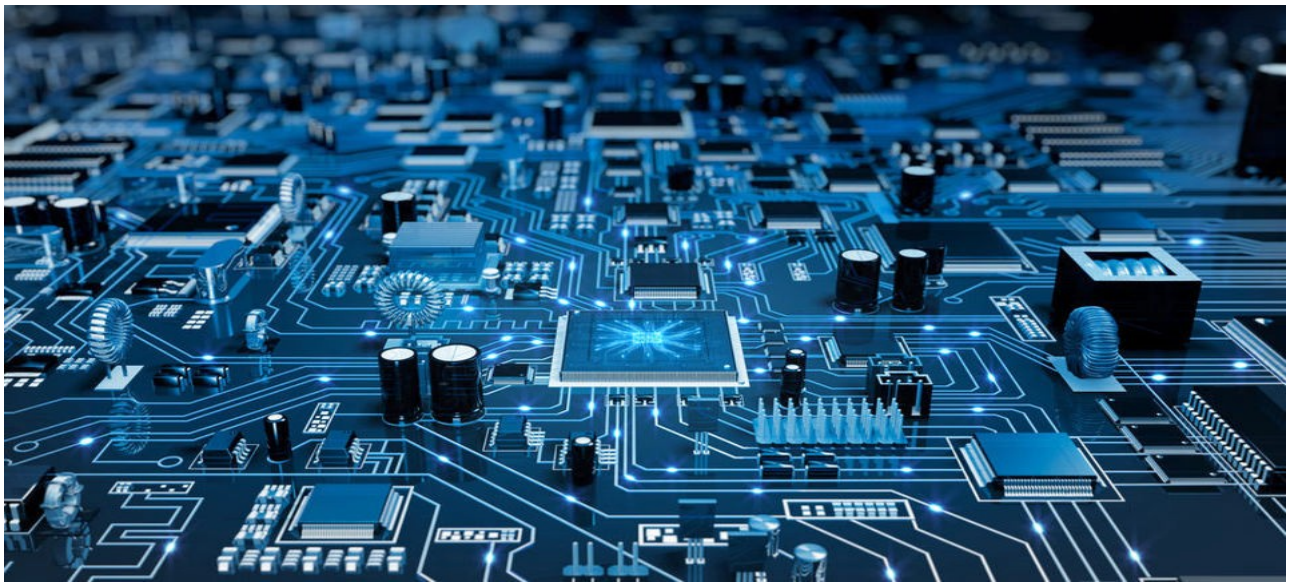
AimValley

Embedded Software
Technology Brief

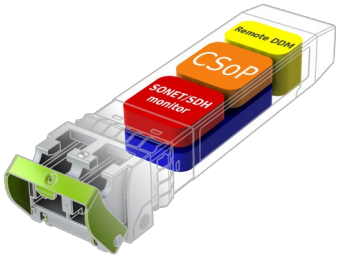


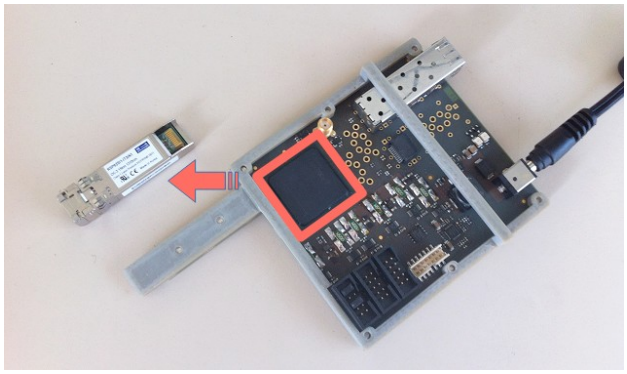
Optimized Hardware through Embedded Software!

Embedded Software is all around us, running in the majority of electrical appliances in house holds, businesses and industries. Often referred to as “firmware”, it is software tailored to specific hardware and the fixed-function, or set of functions, it needs to perform.
Hardware platforms running embedded software range from small controllers with limited processing and memory to multi-board, multi-processor systems running sophisticated control and monitoring algorithms.

Some examples of embedded software:

- Bluetooth controller software in EarPods
- Tire pressure monitoring software running in modern cars’ wheels
- Management and control software running on switches and routers
- Control software running inside (smart)SFPs.





Utrechtseweg 38
1213 TV Hilversum
The Netherlands

phone +31 35 689 1900
sales@aimvalley.com
www.aimvalley.com

AimValley

Embedded Software
Technology Brief

Why use embedded software?

Embedded software is used for implementing complex functions which cannot or should not be realized in hardware, be it in board design, in programmable logic or in ASIC logic.

The strength of embedded software is the ability to perform a multitude of different operations on a heterogeneous set of data and inputs, especially when the program flow branches out into many different conditions, each requiring different actions to be taken. Such cases are best implemented in software.

The basis for a good embedded solution is determined in the design phase of the product where the choice of feature/requirement versus implementation is made. The key here is to find the sweet spot between performance, power and heat dissipation, maintainability and flexibility. Electronic circuitry and implementations that use application specific integrated circuits (ASICs) are very power efficient and fast when a fixed set of operations can be performed on a large, homogeneous data set.

However there is a downside in using hardware and that is flexibility and development cost (NRE). AimValley has expert knowledge in hardware, ASIC/FPGA and software design and can help find the best fit for any embedded design

Our recommendation for using embedded software:

Cost: combinations of complex electronic and mechanical designs can be simplified or replaced through the use of embedded processors running software programs.

Efficiency: the right combination of hardware, ASIC/FPGA and software optimizes the embedded solution in terms of real-estate, power consumption (heat) and performance.

Complexity: when complex processing is required, embedded software is better suited.

Some complex processing examples:

- Management interfaces for configuration and upgrades. Applying configuration items widens the scope of a system's deployment. Allowing upgrades adds the possibility to get the hardware platform to market quickly with a limited feature set and to add features later through upgrades.
- Monitoring/reporting. Fault handling and collecting & reporting of performance counters is a typical general purpose computing task. If a given application requires fault handling and/or performance counters this will be done by embedded software
- Protocols. Due to their complex nature, (networking) protocols are best implemented in software. Especially when multiple interdependent protocols are running on a single system it is preferable to run these in software unless there are very stringent performance requirements.

Flexibility & Future Proof: whereas hard wired electronics can only be changed prior to manufacturing, embedded software and software in general can be changed, upgraded and improved through version upgrades.

As can be seen from the points and examples provided, both embedded software implementations and hardware/FPGA implementations have their own strong points. Embedded software is fit to run large sets of conditional instructions on varied datasets where hardware and FPGA/ASIC implementations are better at performing linear, high-speed operations on large uniform datasets.

Note that hardware is specifically built for one specific data set and is costly to change. An FPGA is somewhat more flexible as it is programmable hardware that can be adapted for various data sets. Depending on cost, complexity and efficiency, software might be the better solution.

Time critical steps can best be handled in hardware/FPGA. Work can be handed-off to software for the less time critical parts.

AimValley
Utrechtseweg 38
1213 TV Hilversum
The Netherlands
phone +31 35 689 1900
sales@aimvalley.com
www.aimvalley.com

AimValley

Embedded Software
Technology Brief

Finding the optimal point to move from hardware to embedded software creates high speed, feature-rich solutions in an optimal system size.

Embedded System Architectures

Given the wide variety of platforms and systems running embedded software, a few main architectures can be distinguished:

Control loop

The simplest of embedded software architectures is a control loop or bare metal software system. This software runs directly on an embedded processor, without any form of operating system. The program runs a top level loop which determines the program flow, in combination with interrupts being handled. The program itself is responsible for management of all processor resources such as non-volatile memory and RAM.

Preemptive/Cooperative multitasking

In this type of embedded system a small operating system layer provides the facilities to run multiple tasks in “parallel” on a single processor and the control structures to facilitate communication and coordination between tasks.
This architecture is normally employed on platforms without Memory Management Unit(MMU). This means that any code can (over)write the data of any other task. This calls for carefully developed software as an error can bring down the entire system.

Multithreading

In this type of embedded system, a sophisticated, feature rich operating system is used. This may be a common operating system with real-time extensions to improve real-time behavior, or a specific full-featured Real-time operating system.

This type of system claims more resources than the other architectures discussed. However, it offers advantages in hardware support/drivers, software packages that can be integrated, closed or open source, and system/debugging facilities. An MMU is required, separating user space and kernel space processes, creating a more reliable system with portable code.

The application consists of one or more processes running one or more threads. Development takes place in languages like C, C++. Since additional features can be added easily, also interpreted languages like Java, python and PHP are encountered on these systems.

Some more characteristics of these architectures are listed below.

Architecture	OS	Resources	Processor	Language	Connectivity	Management
Control Loop	None	RAM: 100B - 10kB NVM: 1kB - ~512kB	8/16/32 bit controller	Assembly, C	Proprietary, if any	Proprietary, if any
Pre-emptive multitasking	(Free)RTOS, OSE, LynxOS	RAM: 10kB - >512 kB NVM: 512kB - >2MB	16/32 bit controller, FPGA softcore	Assembly, C, C++	Ethernet, IP	Lightweight SNMP, HTTP
Multi-threading	Linux, VxWorks	RAM: 10MB - >1GB NVM:64MB - >2GB	32/62 controller, powerPC and ARM cortex A series	C, C++, Java, Python, PHP	Ethernet, IP	SNMP, HTTP, REST, Netconf, CLI

AimValley
Utrechtseweg 38
1213 TV Hilversum
The Netherlands
phone +31 35 689 1900
sales@aimvalley.com
www.aimvalley.com

AimValley

Embedded Software
Technology Brief

AimValley Expertise
Aimvalley is a provider of Carrier class Telecom and Datacom systems employing embedded software solutions tailored to each specific platform and application:
- from large, multiprocessor applications to small bare metal applications.
- BSP packages to support custom hardware platforms.
- high quality, thoroughly tested, robust software components.
- experience with a wide variety of interfaces and protocols.
- realizing a short time-to-market through re-use and portable code.

AimValley proven track record
Nearly all software projects done by AimValley involve embedded software. This has allowed us to build up a large collection of mature and high-quality components that can be used to create a head start for a new project.

AimValley embedded software contributes to the success of products developed on an OEM basis and is used in the following AimValley products:
- EX240 Carrier class ethernet switch
- BX1000 SONET/SDH Optical Access Router
- SmartSFP product family of products.

Why AimValley?
AimValley is a reliable provider of Embedded Software solutions since 2003, delivering solutions for:
- High speed data processing applications
- Complex FPGA-based accelerated systems
- High speed, low power hardware equipment
- Robust embedded software
- Early adopter of Acceleration Technology

AimValley understands the full complexities as well as the subtle nuances of designing great edge solutions. We excel in building complex systems that are part of your product in the fields of Industry 4.0, Big Data, Healthcare and Transportation markets. Our combined skills represent all the important aspects required for the development of end-to-end systems.

Our customers enjoy the benefits of working with a strong team with more than 2 000 years engineering experience. AimValley is a trusted partner of Tier 1 customers in Telecom and Industrial markets and has shipped more than 100 000 products.

Quality Focus
- Outstanding track record of on-time delivery
- Best in Class Designs – Time, Budget & Quality
- ISO9001, ISO140001, Ecovadis Platinum CSR

AimValley EX240

AimValley BX1000

Utrechtseweg 38
1213 TV Hilversum
The Netherlands

phone +31 35 689 1900
sales@aimvalley.com
www.aimvalley.com