

KoliadaES is both a paradigm for embedded systems development and an implementation.

Implementations available; ARM, PPC, MSP430, 8051, i86, iMX, Atmega

- ❖ Features for footprint ratio
 - ❖ Other systems may or may not offer the same features, but KoliadaES provides the broadest reach for the smallest footprint. There is no embedded OS on the market (for small embedded systems) that provides a formal embedded system object model (SOM).
 - ❖ All KoliadaES features are deployed as SOM classes each of which can be included or omitted at developer discretion. This provides complete developer control over the system abstraction footprint, which may be scaled to the exact needs of the application.
 - ❖ SOM allows the system to manage the class objects without prior knowledge of those objects or their classes. Thus application classes may be derived from system classes allowing the management of the class objects by KoliadaES without prior knowledge of the application class or the object. KoliadaES is the only embedded system (other than those using Linux or similar) that allows for binary component deployment of SOM classes.
 - ❖ Binary deployment allows system classes to be deployed as binary components that may be mixed and matched according to the specific SOC they are running on. This is a form of dynamic linking for embedded systems and allows compile once deploy many for homogenous architectures. This can save a lot of hassle for developers managing multiple platforms with varying sub-systems on the SOC.

KoliadaES is the only embedded system that supports a VM for the deployment of 'write once run everywhere' class components in heterogeneous embedded system wired/wireless networks. This allows the same component to be run, for example, on an Atmel based device as is also run on a PIC or an EnergyMicro based device. No change in the component or its data object, run it anywhere (including your PC) and deploy to any supported system without change.

These components may be shared, cached, moved, lost, recovered and run by any supported device.

This is important for failure recovery across heterogeneous systems, robust data collection and distributed embedded systems. The most important aspect is that for other offerings, some of these features require embedded Linux. Using KoliadaES significantly reduces the hardware complexity and footprint.

KoliadaES does all this and more with significantly less footprint and more speed. For example, one client tested KoliadaES vs Linux on an iMX (Freescale) system. KoliadaES = 256k footprint and sub one second boot, Linux = 2Mb footprint and sub one minute boot.

Another client also compared KoliadaES to third party based code (TI Simplicity). KoliadaES used 40% less power than the same functions using TI Simplicity on the same hardware (CC2530 - 8051 based).