

# JAPL - Java Abstract Peripheral Library

In the Embedded Systems world 'peripherals' usually refers to microprocessor and microcontroller I/O interface subsystems like 8 bit I/O ports, UARTS, Timers, Counters etc. With Microprocessor based systems these peripherals are usually attached to the microprocessor's parallel expansion bus and with microcontrollers peripherals can be built into the microcontroller itself, attached to the microcontroller's parallel expansion bus (if it has one) or interfaced to one of the microcontroller's serial expansion buses. Examples of serial expansion buses include SPI, I2C and 1-Wire.

The purpose of the JAPL is to allow Embedded Java code to be compiled against a standard library of Abstract Peripherals. The incentive for creating a library like this is that even though most microcontroller and computer system peripherals are very similar, the specific details of how these peripherals are accessed is quite diverse.

For example, the 8 bit wide I/O port is a very common peripheral but the way that the data direction for each bit is configured, and the manner in which the port itself is accessed, is usually different across systems. Even Java based embedded systems like TINI and JStamp encapsulate their I/O ports in proprietary classes. JAPL solves this problem for the application developer by defining one generic interface for each chip/peripheral type. One way to view this is that each JAPL peripheral type is designed to do for the class of peripherals it is associated with what JDBC does for databases.

Embedlets and JAPL peripherals are distinct technologies and while Embedlets use JAPL peripherals in order to measure and control quantities in the physical world, JAPL peripherals have no dependencies on Embedlets. Put another way, Embedlets live inside of an Embedlet Container and internally they contain no connections to entities outside of the container aside from the JAPL peripherals that they have references to.

© 2002 Apache Forrest