

Modular Devices

Open Source		
Object-Oriented Hardware	Object-Oriented Firmware	Object-Oriented Software
<p>Mixture of processors, electronics, and mechanics for interacting with, sensing, and controlling things in the physical world.</p>	<p>Object oriented firmware/software server objects run on the hardware objects and provide the interface necessary to interact with and control the device.</p>	<p>Object oriented firmware/software client objects may be run on other modular devices or on host computers connected to one or more modular devices.</p>
<p>Each hardware object can be used independently or combined like software objects to create more complex systems.</p>	<p>Each firmware/software server object adds a set of methods, parameters, fields, and callbacks to the device.</p>	<p>The client software may be written in almost any language capable of writing strings over the communication channel to the modular device.</p>
<p>The hardware objects can be implemented in a variety of form factors.</p>	<p>Every device may consist of several layers of firmware/software server objects.</p>	<p>Only one small client software driver should be necessary in each language to talk to all firmware and hardware objects.</p>
<p>Some form factors might be designed to be as small and inexpensive as possible, while others might be larger with more features or capacity, but every object has its own processor, with one or more communication channels, and can be used on its own or combined with any other hardware object, regardless of form factor.</p>	<p>Each layer extends or modifies the layer beneath it, like inheritance and composition in traditional object oriented software. Lower layers may provide general features specific to the particular hardware it is running on, while higher layers may add features specific to the particular task the user may want to run or hardware connected to the device.</p>	<p>When the client software object initializes, it asks the modular devices about its methods, parameters, fields, and callbacks, and then uses that information to create the client object. So new or modified modular devices can be connected to a client object and controlled without the user needing to change or update the client object code.</p>
<p>Hardware objects can scale both up, by combining many of them together when large complicated systems are needed, and scale down, so only a single hardware object is necessary when needs are small and simple, minimizing cost and component count.</p>	<p>Firmware/software server objects are written to maximize code reuse and separation of concerns.</p>	<p>Once the client has connected to the server, other software may interact with the client, calling methods on the local client object that automatically call the remote methods on the modular device over the communication channel.</p>