

calvOS	calvOS System Development	0.1
	General System Architecture	Draft

- calvOS Project -
calvOS System Development
General System Architecture

Intended Audience:	Users / Developers
Author(s):	Carlos Calvillo
File Name:	calvOS System Architecture.odt
Last save on:	05/01/09 - 02:30:33 PM
Total No. of Pages:	7
Document Maturity:	Draft

Author(s): Carlos Calvillo	Version 0.1
http://calvos.sourceforge.net	Page 1 of 7

calvOS	calvOS System Development	0.1
	General System Architecture	Draft

Revision History

Version	Date/ Maturity	Author(s)	Revision	Version Comment
0.1	28/04/09 Draft	Carlos Calvillo		Initial Draft for System Architecture.

Author(s): Carlos Calvillo	Version 0.1
http://calvos.sourceforge.net	Page 2 of 7

calvOS	calvOS System Development	0.1
	General System Architecture	Draft

Table of Contents

1	General Description.....	4
2	Terms Used.....	4
3	Project Description.....	4
4	Project System Architecture.....	5
4.1	GUI Layer.....	6
4.2	Configuration Database Layer.....	6
4.3	Dynamic Data Layer.....	6
4.4	Static Data Layer.....	6
4.5	Application Layer.....	6
4.6	Build Layer.....	7
5	Project Development Items.....	7
5.1	calvOS OS core.....	7
5.2	calvOS Platform Features.....	7

Author(s): Carlos Calvillo	Version 0.1
http://calvos.sourceforge.net	Page 3 of 7

calvOS	calvOS System Development	0.1
	General System Architecture	Draft

1 General Description

The aim of this document is to present the Higher-Level System Architecture for calvOS project.

2 Terms Used

Terms used in this document.

Term	Description
SW	Software
HW	Hardware
OS	Operating System
MCU	Micro-controller Unit
I/O	Input/Output
GUI	Graphic User Interface
db	Database
Config	Configuration

3 Project Description

"calvOS" project is intended to be an Open-Source Operating System for small to medium size embedded applications, our initial steps walk around 8-bit and 16-bit micro-controller families. Our purpose is to offer a widely configurable Operating System allowing users to customize calvOS to match their specific application needs, they will be able to configure calvOS from a minimal non-preemptive kernel with no additional functionalities other than the ones required for it up to a preemptive system (if micro-controller supports it) with additional features like I/O control, power modes management, memory management, serial communication, etc.

One of our biggest goals is to provide end-users with a Graphical User Interface to fully-configure and prepare calvOS system to start developing their own application layers. With this GUI our intention is that end-users with minimal or null knowledge regarding operating systems can build more interesting and complex applications by getting the advantages of using an Operating System.

Our Initial development line is to cover 8-bit Atmel AVR micro controller's family with GCC-AVR compiler. All our design is focused to allow calvOS system to be easily ported to other compilers and micro controllers and future development plans involve the use of the GUI to select the target MCU and compiler as they get natively supported by the calvOS Development Team.

Future micro-controller architectures planned to be covered by us are:

- 8-bit PIC
- 16-bit s12

Author(s): Carlos Calvillo	Version 0.1
http://calvos.sourceforge.net	Page 4 of 7

calvOS	calvOS System Development	0.1
	General System Architecture	Draft

- 32-bit AVR32

4 Project System Architecture

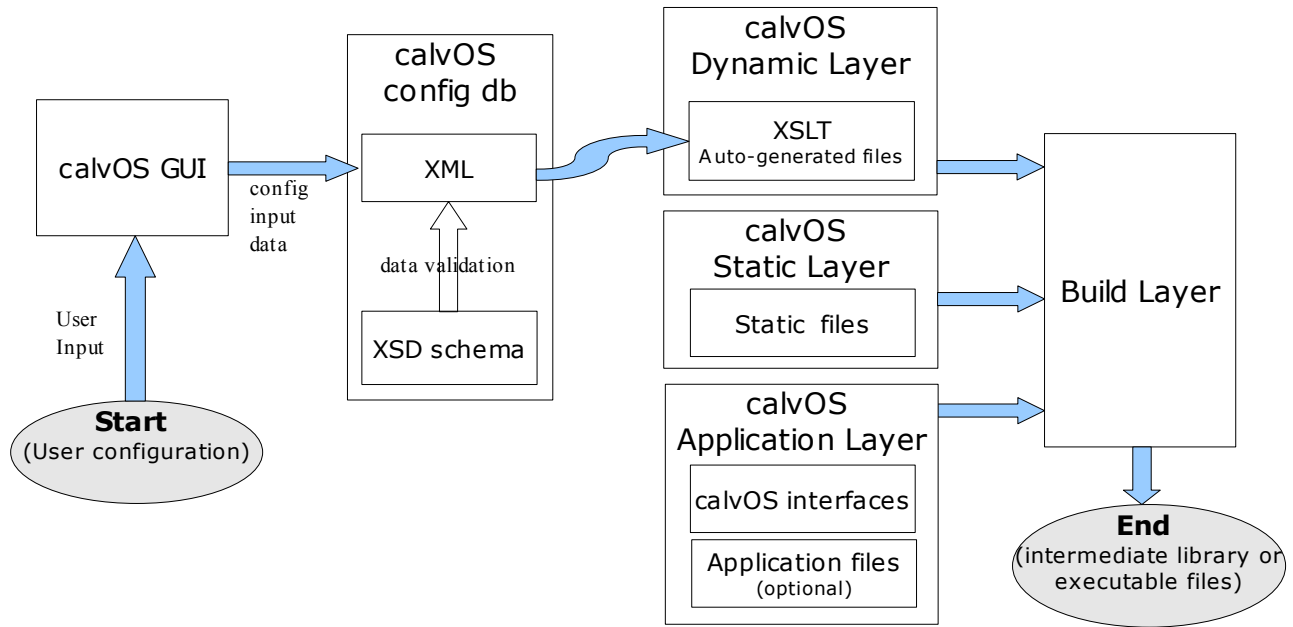
In order to achieve the goals of our project we have divided it into the following components:

- **calvOS GUI:** This is the front-end of our project.
 - This GUI must allow end-users to customize all the configurable parameters of our project.
- **Configuration database:** Is the place where all the settings entered by the user will be saved.
 - XML and XSD technologies will be used to validate and store the parameters configured by the user through the GUI.
- **Dynamic Layer:** This involves all the required actions to prepare the customized calvOS packet.
 - XSLT technology will be used to transform the configuration parameters stored in the XML files into source code and other required files used by calvOS build process.
- **Static Layer:** Involves all the source code and source data that is static and is used in conjunction with the dynamic layer to build the calvOS system.
- **Application Layer:** Is the place where all the calvOS source code interfaces to the applications are given.
 - If desired, in this layer the end-user can include their application source files in order to build them together with the calvOS system.
- **Build Layer:** All the required actions (like call the compiler, linker commands, etc.) to build the calvOS system and generate the outputs.
 - At this moment we haven't decide the technology to be used by this layer. Some options are: perl, bash scripting...
 - Output files could be two options:
 - Intermediate libraries and corresponding interfaces to be used in the application development environment.
 - Executable files including or not the application files (if application decide to use Application Layer to include their source code).

These components and a brief idea of their interactions are showed in the following Drawing 1.

Author(s): Carlos Calvillo	Version 0.1
http://calvos.sourceforge.net	Page 5 of 7

calvOS	calvOS System Development	0.1
	General System Architecture	Draft



Drawing 1: calvOS System Architecture

4.1 GUI Layer

TBD

4.2 Configuration Database Layer

TBD

4.3 Dynamic Data Layer

TBD

4.4 Static Data Layer

TBD

4.5 Application Layer

TBD

4.6 Build Layer

TBD

Author(s): Carlos Calvillo	Version 0.1
http://calvos.sourceforge.net	Page 6 of 7

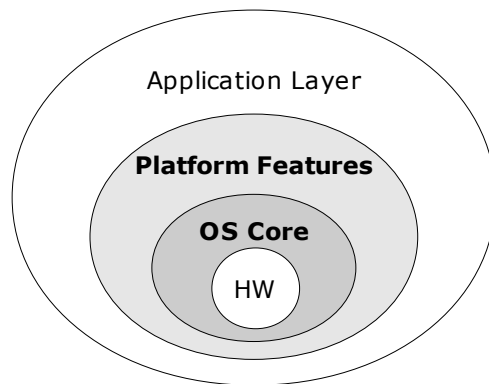
calvOS	calvOS System Development	0.1
	General System Architecture	Draft

5 Project Development Items

Section 4 Project System Architecture shows the architecture of the system but we can go some step deeper and divide the project in two main items (each one following the same architecture showed in section 4):

- calvOS OS core
- calvOS Platform Features.

Next Drawing 2 shows the interactions of these items.



Drawing 2: Development Items Interaction

5.1 calvOS OS core

With "Operating System Core" we made reference to all the stuff involving just the Operating System. This is for example: the scheduler, task dispatcher, operating system hardware abstraction layer, etc.

5.2 calvOS Platform Features

This includes all the features that are not direct part of the operating system but provides handling of the most basic functionalities of a micro controller. For example: memory management, I/O controller, Power Modes Management, Serial Communication, etc..

Author(s): Carlos Calvillo	Version 0.1
http://calvos.sourceforge.net	Page 7 of 7