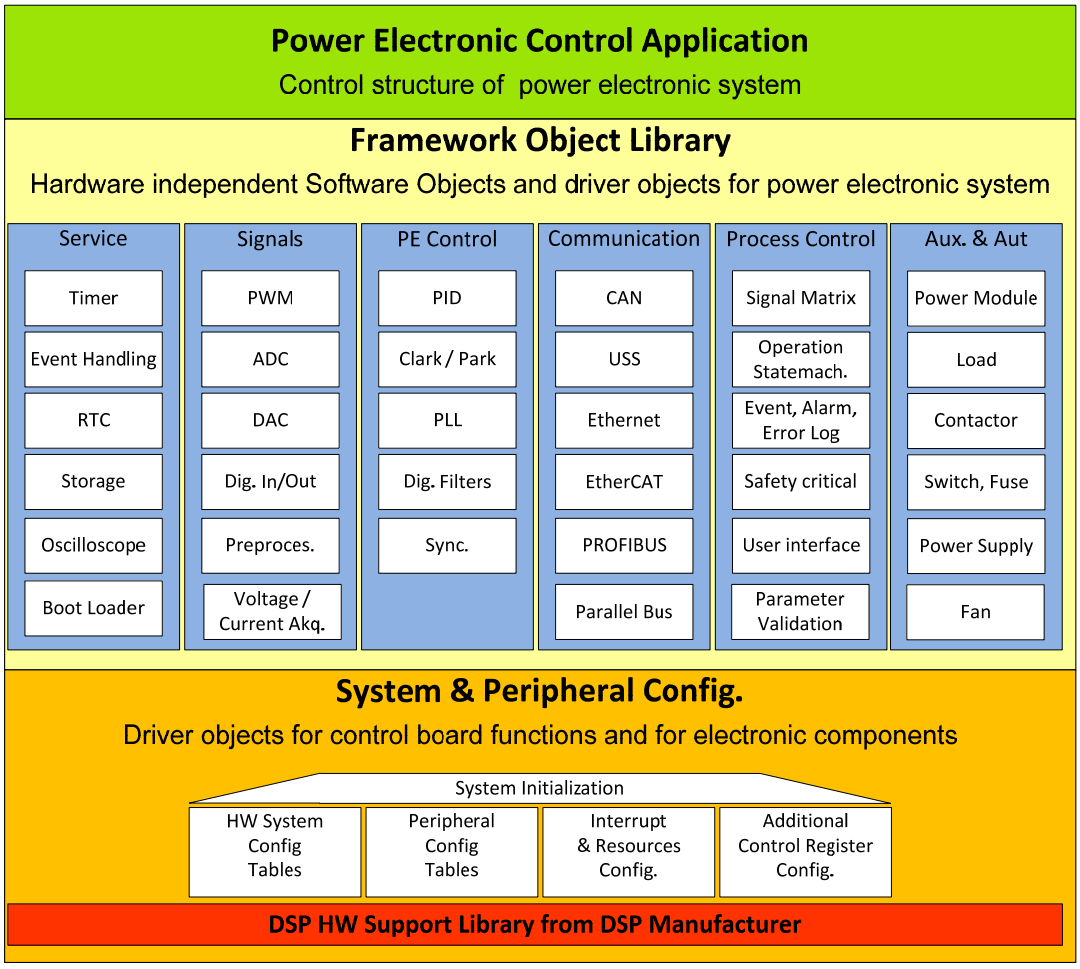
Software Framework for DSP Control Board

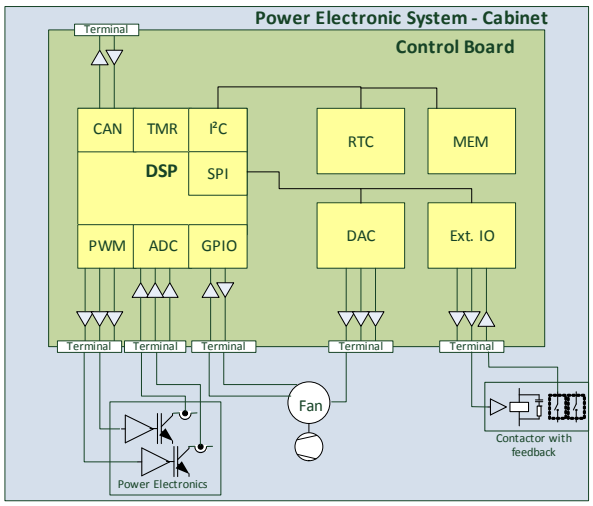


The object oriented real-time software framework has been designed to process control tasks and to interact between the application software and the power electronic system. It provides access to data communication interfaces and the I/O peripherals.

There are access layers for a modular software approach, which ensures system and spare parts availability, even when components will have become obsolete and hence will be out of production. Then, with new controllers or new peripherals devices, a successor control board can be designed and the framework can be adapted easily to reuse of the existing application code.

Main Feature List

- Hardware / peripheral access through hardware abstraction layer
- Board Specific Objects
- Peripheral Object Library
- Digital I/O Objects
- External hardware access objects (RTC, EEPROM, MEM etc.)
- Configuration of ADC and PWM
- Integrated oscilloscope and monitoring
- Data communication interfaces and protocols
- System management and safety
- Interrupt system configuration, task scheduler
- IGBT fault detection and system protection
- Configuration and parameter updates
- Power Control Object Library
- Developments system installation procedure (guide and installer)



The Framework layers provide a flexible driver concept to access hardware and system signals on a generic engineering base. Signal processing, power electronic and data communication hardware has been encapsulated into software objects.

Power Electronic Control Application:

This section contains the application specific software objects, using the framework classes and objects. The system developer works on this layer, which contains state machines, signal matrix, power electronic control and signal processing. Signals can be named according to the system drawing and are scaled in floating point format in physical dimensions (e.g. A, V). The application class provides a method, which is synchronously called by the PWM and ADC peripherals of the DSP to process real-time critical control algorithms. Further, e.g. timer based methods are provided for extensive control and data processing and data communication. The application layer allows the implementation of fully user definable classes and objects, to implement any application or product specific software modules.

Framework Object Library:

This section contains control algorithms and function blocks, which do not directly interact with hardware. It covers very common control classes for the operation of power electronic converters. Examples are:

- PID controller
- Clarke / Park transformations,
- Space vector modulation
- Digital filters
- Phase locked loop (PLL)
- V/f control for drives
- Main contactor K1
- Link Voltage VDClink
- Output current IWRL1

The system programmer can easily derive as many objects as needed from the framework classes.

System & Peripheral Config:

This section refers on control board signals, regardless of the entire electronic circuits.
It contains objects for standard access to electronic components on the control card. All hardware or device specific requirements are encapsulated in the entire software objects. Examples are:

- DSP internal peripherals (ADC, PWM, Dig I/O, Memory Access, SPI, CAN, Interrupts, Timers etc.)
- External components (SPI based I/O, real time clock, analog multiplexer, LEDs)

Long Term Product Availability, Obsolescence

The software framework architecture gives the product developers a powerful toolset to ensure long time availability of product spare parts. As the market availability of electronic chips like DSPs is limited, spare control cards can be easily built with future generations of DSPs, while the application specific part of the software can be kept, and the circuit framework layer will need to be updated. The concept implies that future DSP or hardware components will provide at least the same or most likely higher processing and peripheral performance.