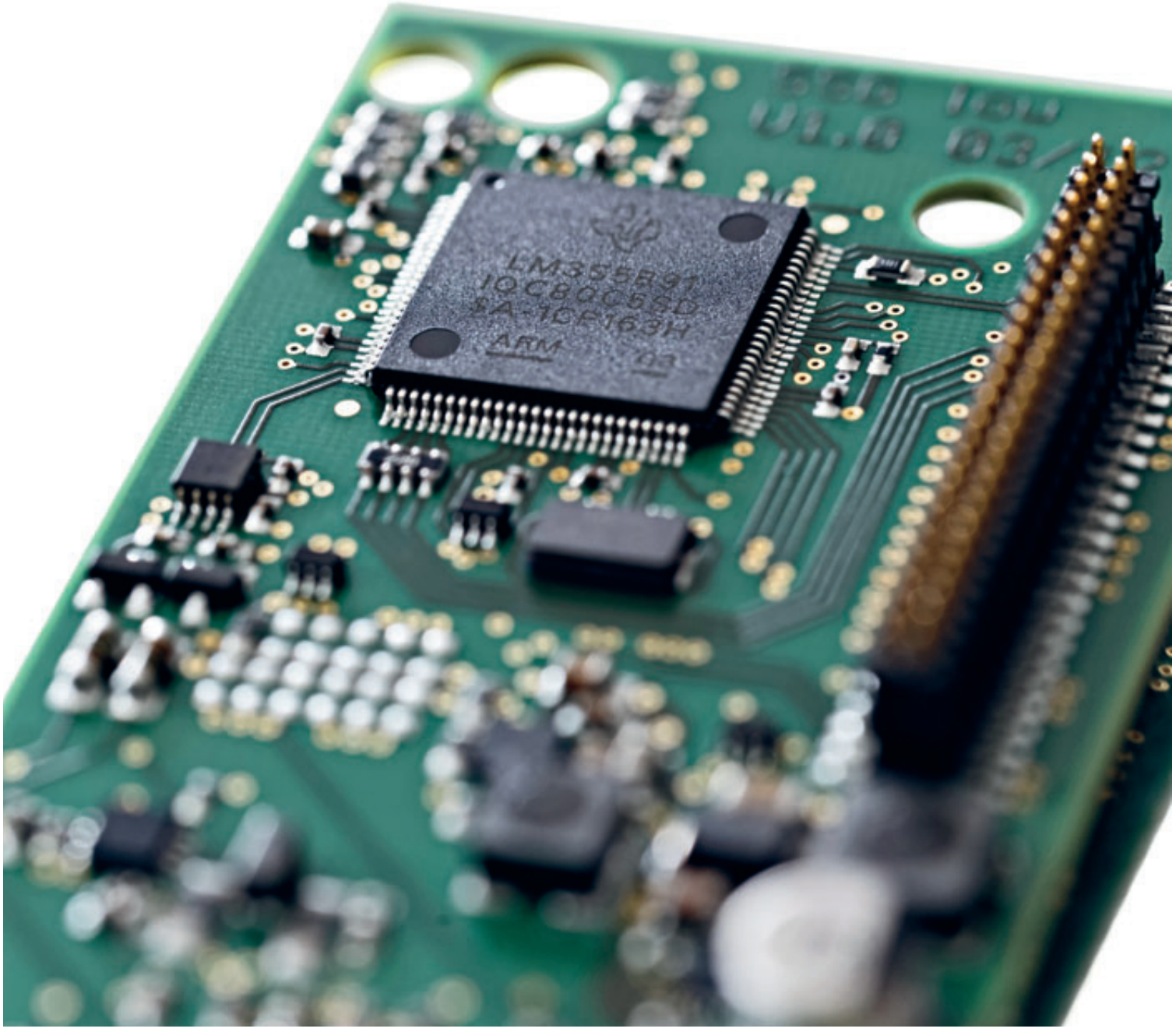
# Prototype Controller

In the automotive industry, the use of prototype controllers for developing new vehicles and products is standard practice. Porsche Engineering offers a requirements-based hardware and software platform that provides just the right solution, whatever the application.

*By Norbert Nentwig, Ulf Schlieben, Jens Müller, Andreas Müller  Photos by Jörg Eberl*



In order to remain ahead of the competition, it is important for a company to keep developing innovations for vehicles as quickly as possible, and to make them ready for series production. The implementation of this approach requires an efficient product development process and appropriate tool chains. In the future, with its modular prototype controllers, Porsche Engineering will be able to provide a flexible, economical, and robust solution for every application.

### Why does Porsche Engineering have its own prototype hardware platform?

Electronic controllers in vehicles ensure improved comfort and safety, and reduce the impact on the environment. In order to be able to apply all of these functions, more and more controllers are required, and the need for exchanging information increases.

During the development process, a large number of tests, simulations, and measurements are performed in the prototype phase, so that the desired functions can then be mapped in a number of iteration steps on the target hardware created by the supplier. To shorten this process and to be able to perform a verification of the simulation in the vehicle under real conditions as soon as possible, the development of cost-efficient prototype platform was started.

### What are the benefits of prototype controllers from Porsche Engineering?

> Cost-efficient hardware
> Complete solutions: "All from a single source"
> Flexible and open architecture through the use of standard components

### What does the modular structure of the platforms look like?

The basis of the controllers is the ARM Cortex processor series, which provides a large range of services and functions. The basic software, drivers, and interfaces are the same for all processors, and this simplifies the development process considerably. The areas of application are grouped according to performance, handling, safety, and environmental parameters.

The table below provides an overview of the various properties of the interface controllers (referred to below as ICs) of the types IC LOW, IC MID and IC HIGH. The following example uses the IC LOW to explain the properties and special features and thus clarify the hardware and software structures.

### Interface controller LOW

The IC LOW covers the lower performance spectrum of the controller platform, which includes topics such as sensor evaluation, sensor simulation, simulation of LIN nodes, logger functions, and simulation of diagnostic/tester functions. This also provides the option to implement customer-specific hardware extensions. A version with an on-board diagnostics (OBD) connector housing is currently finished and in use. ›

| FUNCTION | IC LOW | IC MID | IC HIGH |
|---|---|---|---|
| Basic IO | ■ | ■ | ■ |
| Individual software | ■ | ■ | ■ |
| Matlab/Simulink | — | ■ | ■ |
| HMI/Display | — | ■ | — |
| Ambient conditions | Interior | Interior | Engine compartment |
| Safety | | | IEC 61508 SIL-3<br>ISO 26262 ASIL-D |
| Application | Sensors, HW Gateway, Diagnostics | Logger, Network Gateway, HMI, Simulations | Transmission, Engine, Battery Management, Electric Vehicle Manager |

## Hardware description

The IC LOW consists of a motherboard and an extension board (see figure below). The motherboard contains the processor, the hardware inputs and outputs, and the voltage supply. The current extension board for the OBD connector housing (see figure at upper right on right-hand page) contains a Bluetooth module for communication and a micro SD card slot for mapping logging functions. The extension board is connected via a 50-pin connector and can be adjusted to meet specific customer needs. A version for implementation in an aluminum housing is currently under development.

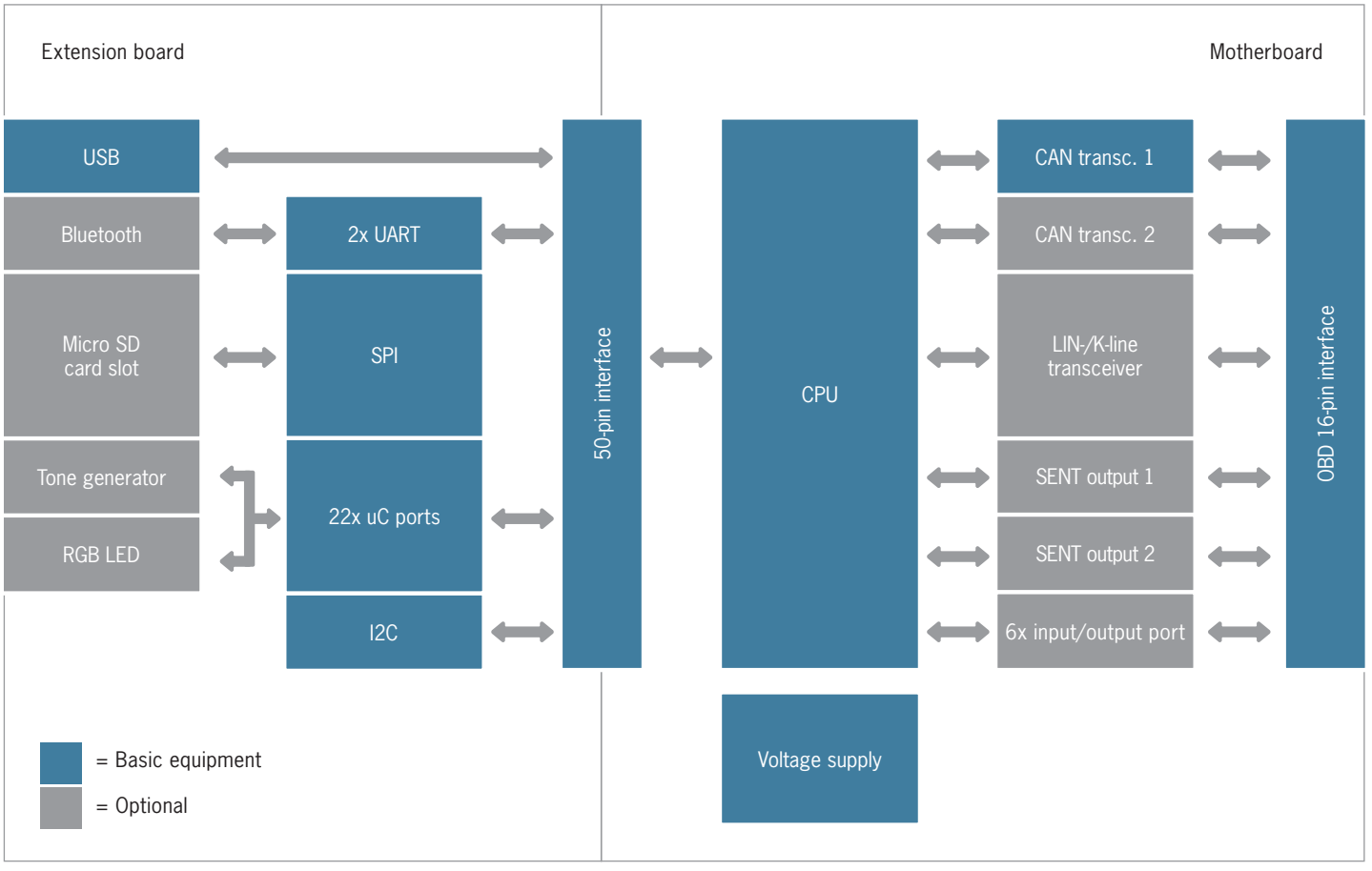
## Software description

Due to the complexity of modern microprocessors and the significantly increased demands on the software, flexible and modular software architecture is needed that can meet the following requirements:

> Structured software design
> Reusability of components
> Hardware-independent function design (separate behavior from the hardware architecture)
> Portability of the functions to different hardware platforms without changing code
> Use of manufacturer libraries

The software for the ARM Cortex M3 processor of the IC LOW has been developed based on the software library principle. Every hardware module has its own initializations and functions, so that the required hardware can be addressed as required. With this step, it is possible to take existing functions and use them for future developments with the same process architecture without any problems.

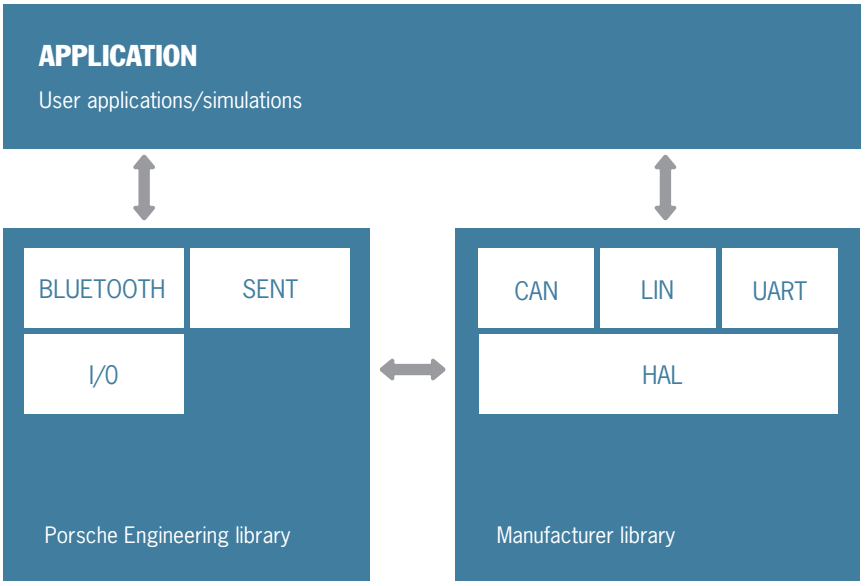
The modular hardware means that the motherboard, together with the voltage supply, CAN, LIN, SENT, and processor, can also be used in other hardware developments. Using the software library, basic functions can be implemented quickly, and the actual function development can be prioritized.




*Block diagram of the IC LOW, consisting of a motherboard and an extension board*

*OBD housing and interior*

## Software architecture




As each project accesses the library developed specifically for the IC LOW, changes are effective immediately across the board. This preliminary work makes it very easy to incorporate new projects.

Thanks to the variety of implementation options of the IC LOW, in parallel with the embedded software there is also a Windows interface for managing the individual software versions and documenting the devices already in use.

An integrated boot loader enables an older software version to be updated at any time. Other functions of the Windows interface include function tests, monitoring processes, and displaying operating statuses.

In its current development phase, five different projects have been implemented with the IC LOW. These include starting the fuel feed pump with a diagnostic command, activating/deactivating the dynamometer mode for all-wheel and rear-wheel drive, simulating a controller, and also incorporation into the "TargetSetupCenter" software. ■