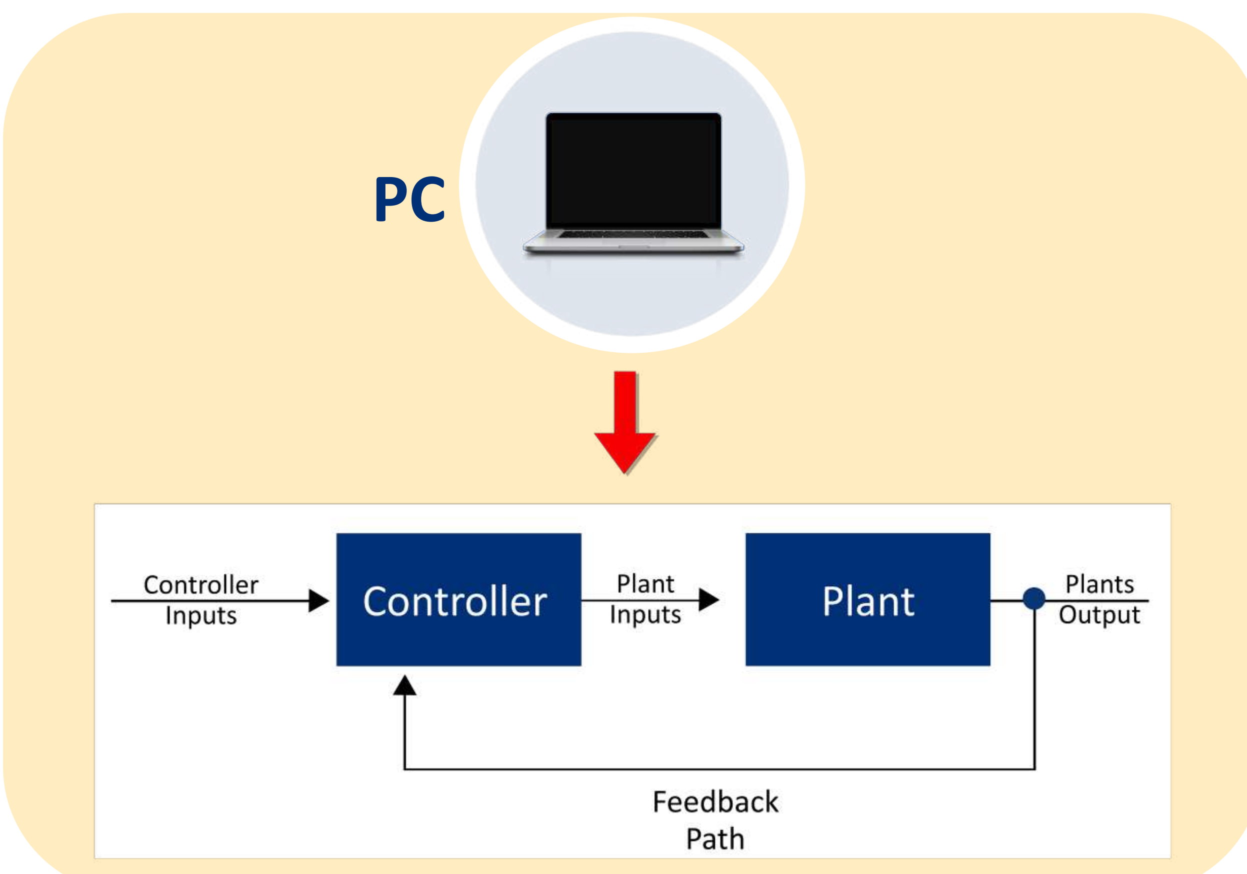
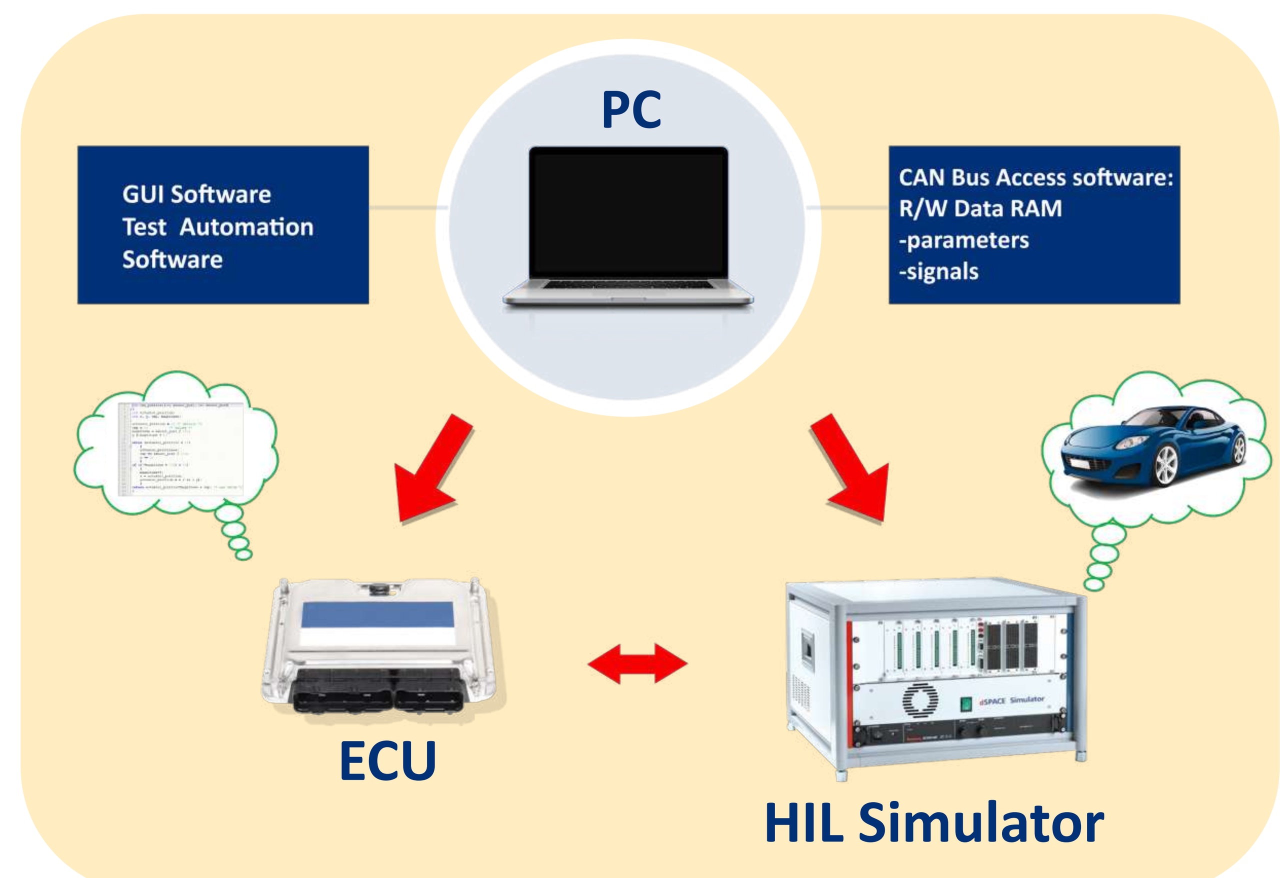
Embedded ECU Software :

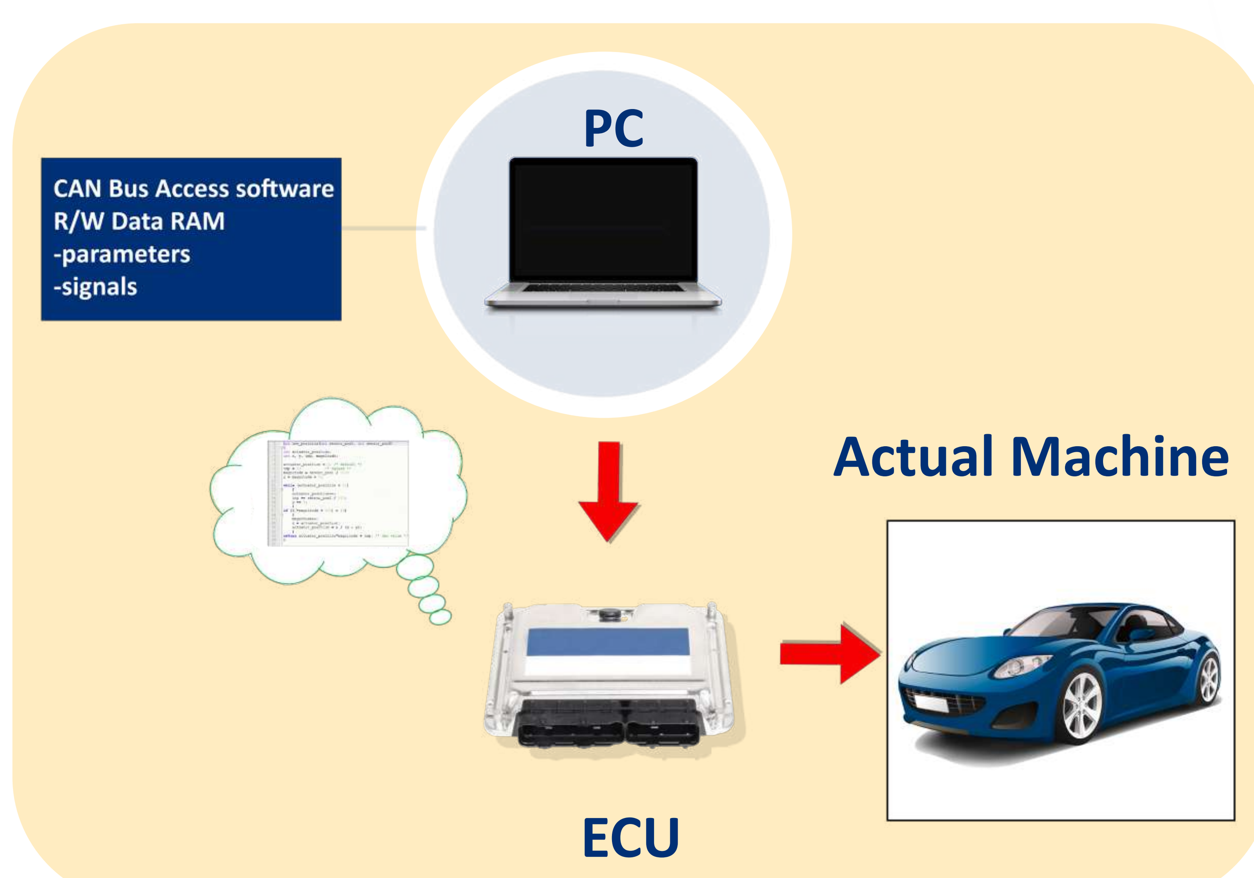
Development, Testing and Validation

Phase 1: Desktop Engineering



Phase 2: HIL Testing





Phase 3: On Machine Testing

An embedded control software is developed in three main phases. A controls engineer may be involved in one or all of these stages .

- Phase 1: Desktop engineering: non-real-time, concept, modeling, and simulation
- Phase 2: Hardware-In-the-Loop (HIL) testing : real- time
- Phase 3: Prototype machine testing

Embedded ECU Software :
Development, Testing and Validation

An Embedded controller is a rugged computer hardware , which may include power amplification circuit, with an application-specific software. The embedded controller hardware, also called Electronic Control Unit (ECU) or Electronic Control Module (ECM), is rather standard provided by various suppliers to Original Equipment Manufacturers (OEM). The ECU software, however, is always application-specific and custom developed for each application by an OEM.

Three phases of embedded control software development

In Phase 1 of the development process, the control software for the electronic control unit (ECU) is created using graphical software tools like Simulink and Stateflow. The software is simulated and analyzed on a non-real-time desktop environment. A detailed dynamic model of the machine called the "plant model," is used for accurate simulations. The software is developed in different layers with defined interfaces between them, including a hardware I/O layer, core logic layer, and application layer. The software contains both the logic, such as control algorithms, and parameters, such as gains for the control algorithm.

In Phase 2, the control software is tested on a target ECU using a Hardware-in-the-Loop (HIL) system. The software is auto-generated into C-code from the graphical diagrams. The embedded control software runs on the ECU in real-time, as it will on the actual machine ECU. The HIL system simulates the behavior of the actual system and provides real-time input and output signals to the ECU. The plant model is simplified compared to Phase 1 to run in real-time. The HIL system hardware emulates the actual machine's signals. HIL testing allows for the identification and resolution of problems before testing on the actual machine. It is cost-effective and safer than testing on a prototype machine, especially for life-critical applications. The HIL system provides repeatability and allows for testing under extreme conditions that cannot be easily created in the real world. HIL testing is an intermediate step between pure software simulation and pure hardware testing.

In **Phase 3**, the ECU is tested and tuned on an actual prototype machine. The I/O interfaces, sensors, actuators, and software logic are verified and calibrated. The control algorithm parameters are further tuned to optimize the dynamic performance. The machine's performance and reliability are compared to benchmark results, and the results are documented for production release. Matlab/Simulink files can be auto-code generated and flashed into the ECU using a laptop PC and a USB/CAN bus interface module. The laptop PC connects to the ECU's CAN bus, and MCD software tools implement communication protocols to read/write data memory locations of the ECU.

Various software tools like CANalyzer and CANape are used for HIL testing, enabling monitoring, test condition definition, data collection, and result analysis. These tools facilitate memory read/write access, test plan execution, and automation of test and validation processes.

HIL testing can be performed by independent engineering teams or companies, providing objective and exhaustive testing. It offers cost savings, better safety, and test repeatability. HIL testing by independent organizations is especially important for OEMs in order to assure impartial testing and validation of their products.