SERVOTECH
Award Winning Engineering Services

# EMBEDDED SOFTWARE


## The software development and testing process has the following steps:

- Develop the control algorithm in Matlab/Simulink Environment. For simulation purposes, develop a "plant" model.
- Matlab/Simulink code can be further extended with C/C++ functions using the S-functions interface.
- Simulate, debug, test, and evaluate the control system in the non real-time, Laptop's Matlab/Simulink environment.
- Auto-code the control algorithm for the Target ECU: generates C code.
- Build the C code for the target ECU and generate executable code: using the C Compiler tool.
- Download/Flash the executable code to ECU: using the Calibration and Tuning Software tool.
- Run the code on the ECU, tune, calibrate, test, and validate it while running on the ECU, using the Calibration and Tuning Software on the Laptop PC.

**Email:** info@ServotechInc.com | **Phone:** 312-376-8103 | www.ServotechInc.com

# SERVOTECH
**Award Winning Engineering Services**

# EMBEDDED SOFTWARE
**DEVELOPMENT TOOLS**


Embedded software is the vital component that is the "brains" in all of the devices used in our civilization, from microwave ovens, cars, tractors to airplanes. Embedded software is the program that runs on the embedded controller hardware, referred to as ECU (electronic control unit) or ECM (electronic control module), which is a printed circuit board (PCB) in one package that includes a microcontroller, IO interface, and often power amplifier circuits. The exact content and shape of the ECU vary depending on the application market, i.e. ECUs used in vehicle applications are different that the ECUs used on microwave ovens or medical equipment. The smarter the functionalities of the device, the more sophisticated the embedded software needs to be.

As systems become more and more complex, so do the complexity of embedded software. For instance, the embedded software on a microwave oven is much simpler that the embedded software on a self driving car. Equally important aspect of embedded software development is its testing and validation; how do we make sure it is reliable. Extensive simulation testing technologies are utilized in testing and validation, such as model in the loop (MIL), software in the loop (SIL), hardware in the loop (HIL) testing. Depending on what is at stake (lives or inconvenience), the extend to which embedded software is tested and validated varies in different applications.

## Embedded Software versus Non–Embedded Software

Embedded software is a type of computer software that is "embedded" in a device that we use.
The differences between "embedded software" and general purpose "software" are as follows:

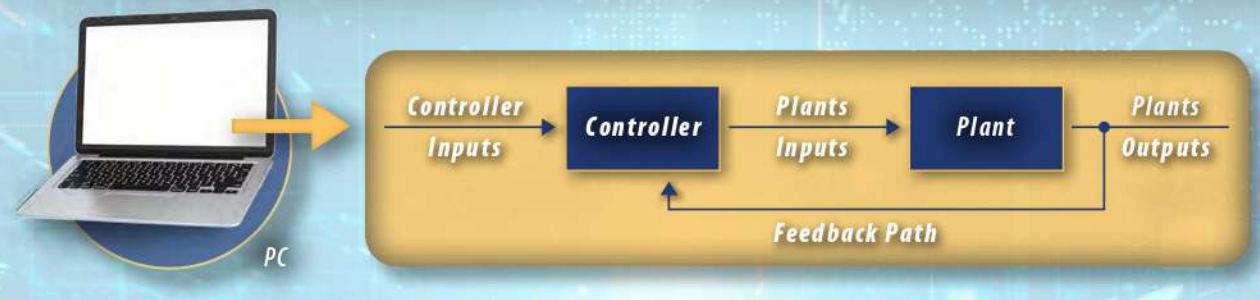
- Embedded software deals with real world IO in real-time. A few seconds or even milliseconds of delay can have serious consequences, including loss of life, in some embedded control applications. Whereas such delays in general software may simply result in inconvenience to the user.
- Embedded software is the "brain" of a device. The smarter the "brain" is, the smarter the device is.
- Embedded software reliability is paramount importance because lives may be riding on it, hence must be tested extensively using exhaustive testing techniques using MIL/SIL/HIL technologies. Non-embedded software generally does not have this strict requirement on reliability.
- As the markets demand more and more sophisticated devices, with more reliability and lower cost, the embedded software development task necessarily becomes more sophisticated, must be reliable, and development cost must be low.
- Security is a fundamental concern for embedded software and embedded computers, since virtually everything will shortly have IIoT connectivity, opening a box for many opportunities as well as security vulnerabilities.

**Email:** info@ServotechInc.com | **Phone:** 312-376-8103 | www.ServotechInc.com

# SERVOTECH
Award Winning Engineering Services

# EMBEDDED SOFTWARE
DEVELOPMENT, TESTING AND VALIDATION

## An embedded control software is developed in three main phases.




**PHASE 1:** DESKTOP ENGINEERING – non-real-time, concept, modeling, and simulation


**PHASE 2:** HIL TESTING – real-time


**PHASE 3:** ON MACHINE TESTING – Prototype machine testing

**Email:** info@ServotechInc.com | **Phone:** 312-376-8103 | www.ServotechInc.com

SERVOTECH
Award Winning Engineering Services

# EMBEDDED 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.***

---

## PHASE 1–

In Phase 1 of the development process, the control software for the electronic control unit (ECU) is created using software tools like Matlab, 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.

---

## PHASE 2–

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 Simalink. 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.

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.

---

## PHASE 3–

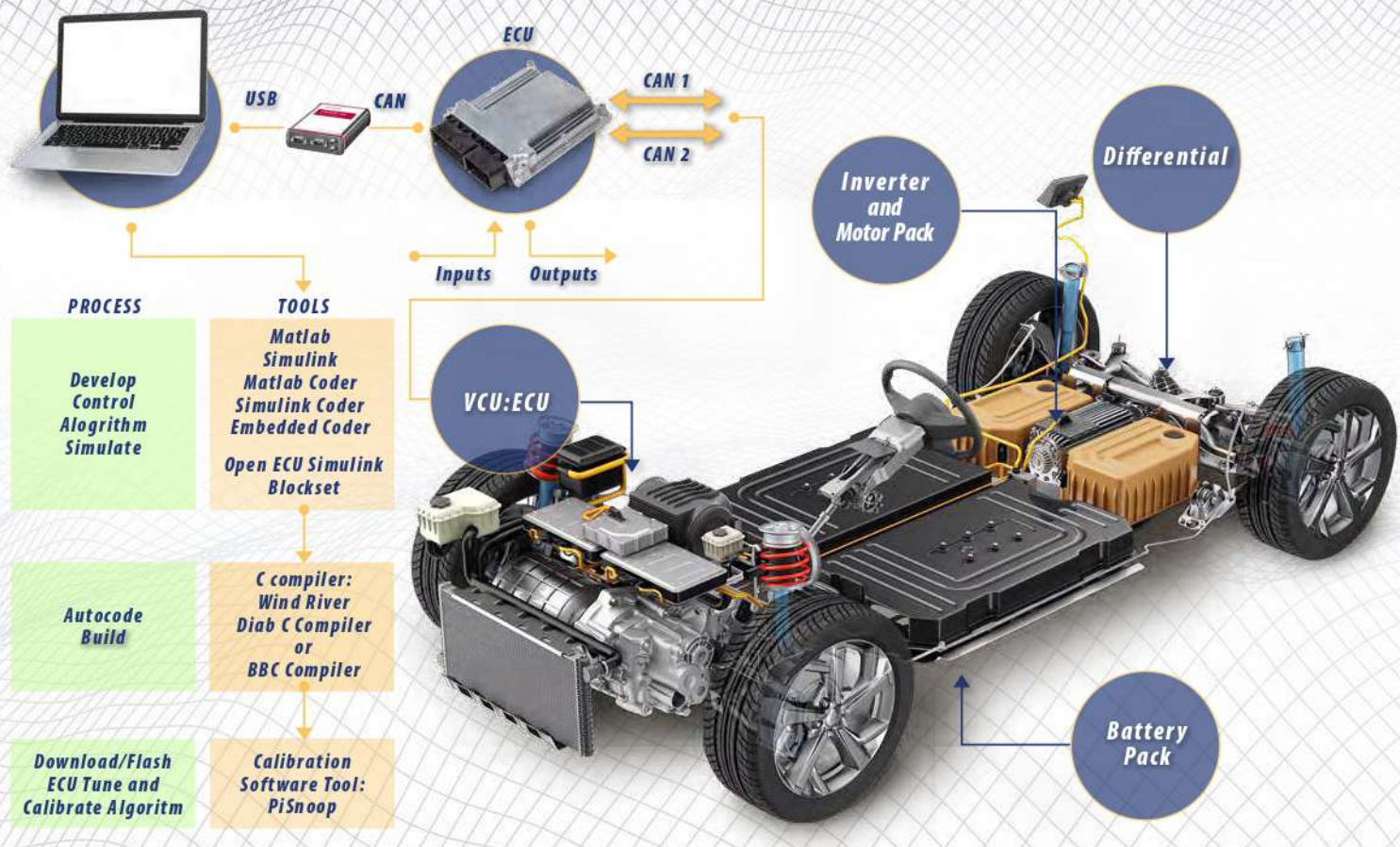
In Phase 3, The ECU is tested and tuned on an actual prototype machine. The I/O interfaces, sensors, actuators, and software logic are verified and calibrated. The control algorithm parameters are further tuned to optimize the dynamic performance. The machine's performance and reliability are compared 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.

***Email:*** *info@ServotechInc.com* | ***Phone:*** *312-376-8103* | *www.ServotechInc.com*

SERVOTECH
Award Winning Engineering Services

# EMBEDDED CONTROL SOFTWARE DEVELOPMENT




## The Embedded Software Development Tools

### Hardware

1. Laptop PC
2. USB/CAN Interface Module

### Software on the Laptop PC

1. Windows 10 (64 Bit)
2. Matlab
3. Simulink
4. Matlab Coder
5. Simulink Coder
6. Embedded Coder
7. Simulink Block Set for ECU Support
8. C Compiler (for target ECU i.e. Wind River)
9. Calibration and Tuning Software, i.e. CANape by Vector

**Email:** info@ServotechInc.com | **Phone:** 312-376-8103 | www.ServotechInc.com

SERVOTECH
Award Winning Engineering Services

# EMBEDDED CONTROL SOFTWARE DEVELOPMENT

## The embedded software development and testing process consists of several steps, which are as follows:

### 1. Develop the control algorithm in the Matlab/Simulink environment:

In this step, the control algorithm is designed and implemented using the Matlab/ Simulink software. The algorithm is typically developed using a graphical interface, allowing for easy visualization and manipulation of the control logic.

### 2. Simulate, debug, test, and evaluate the control system:

Once the control algorithm is developed, it is simulated in the non-real-time environment of a laptop using Matlab Simulink. This allows for thorough testing and evaluation of the control system's performance under different scenarios and inputs. Any issues or bugs in the algorithm can be identified and debugged at this stage.

### 3. Auto-code the control algorithm for the Target ECU:

After the control algorithm has been verified through simulation, it is auto-coded to generate C code. Auto-coding is the process of automatically translating the algorithm from the Matlab/Simulink environment into executable C code that can run on the target Electronic Control Unit (ECU).

### 4. Build the C code for the target ECU and generate executable code:

In this step, the generated C code is compiled using a C Compiler tool. The compiler translates the human-readable C code into machine-readable binary code that can be executed by the target ECU.

### 5. Download/Flash the executable code to ECU:

Once the C code is compiled and transformed into executable binary code, it needs to be downloaded or flashed onto the target ECU. This process involves transferring the code from the development environment to the ECU using specialized tools and protocols.

### 6. Run the code on the ECU, tune, calibrate, test, and validate it:

After the executable code is loaded onto the ECU, the control system is executed on the actual hardware. The system is tuned, calibrated, tested, and validated in real-time using Calibration and Tuning Software running on a laptop PC. This allows for fine-tuning of control parameters, configuration of parameters for different product models, performance evaluation, and validation of the control system's behavior under real-world conditions. HIL Testing is highly effective before testing on an actual vehicle.

***Overall, these steps ensure a systematic approach to developing and testing control systems, starting from algorithm development and simulation, to auto-coding, building, downloading, and finally validating the control system on the target ECU.***

**Email:** info@ServotechInc.com | **Phone:** 312-376-8103 | www.ServotechInc.com