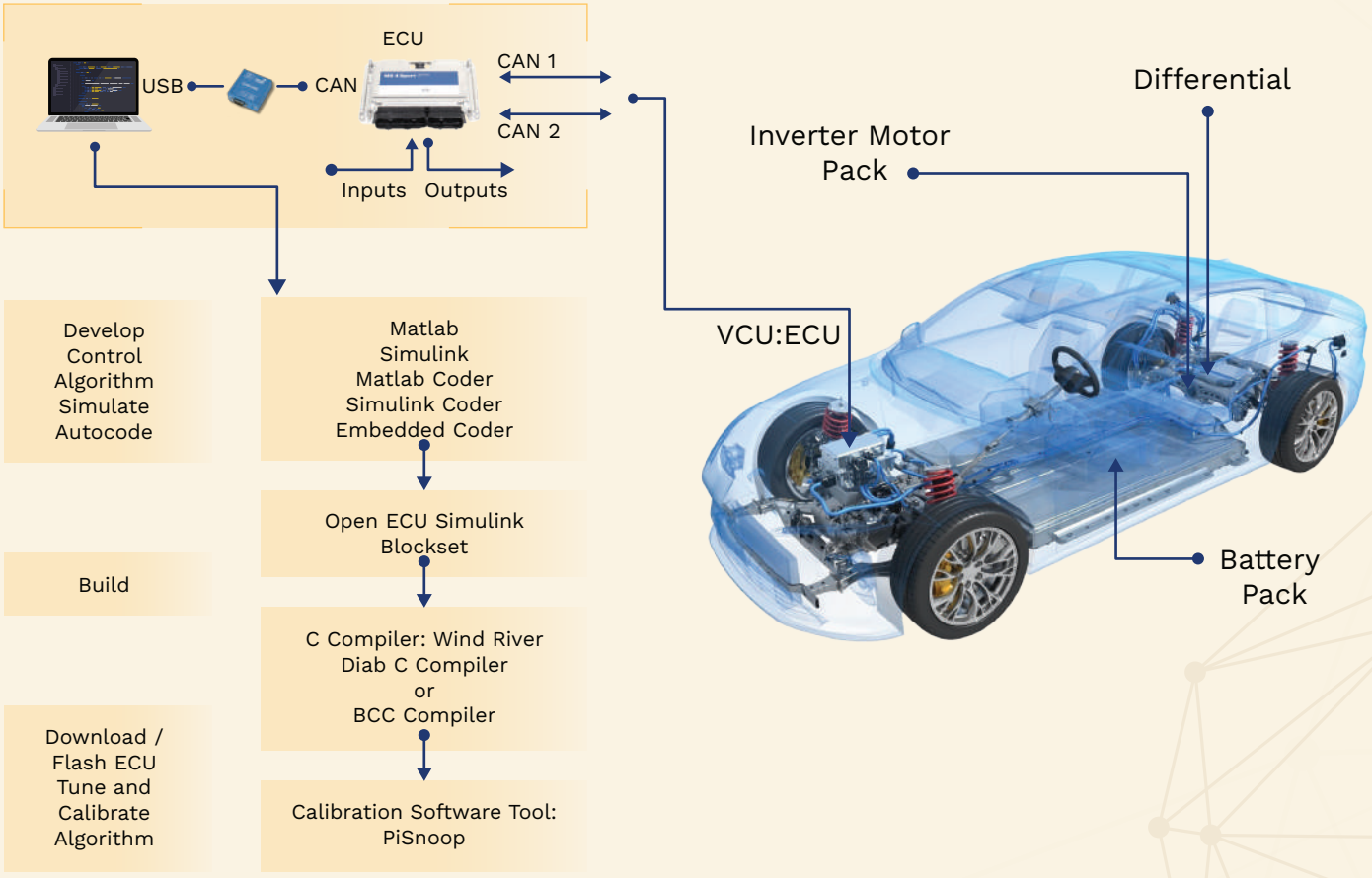
SERVOTECH

Embedded Control Software Development



The Embedded Software Development for the VCU (for that matter, for any ECU) are as follows:

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, i.e. Open ECU by Pi Innovo
8. C Compiler (for target ECU i.e. Wind River)
9. Calibration and Tuning Software, i.e. CANape by Vector, PiSnoop by Pi Innovo

info@ServotechInc.com
www.ServotechInc.com

SERVOTECH

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, 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.