Agenda of this course

- Make you understand what are the importance of HIL Testing
- What kind of Scenarios included in HIL Testing
- What are the components Involved
- How the Tools dealt in HIL Testing
- Overall Understanding with HIL Testing
- Interview Q&A in HIL Testing
- Few other topics related to HIL Testing

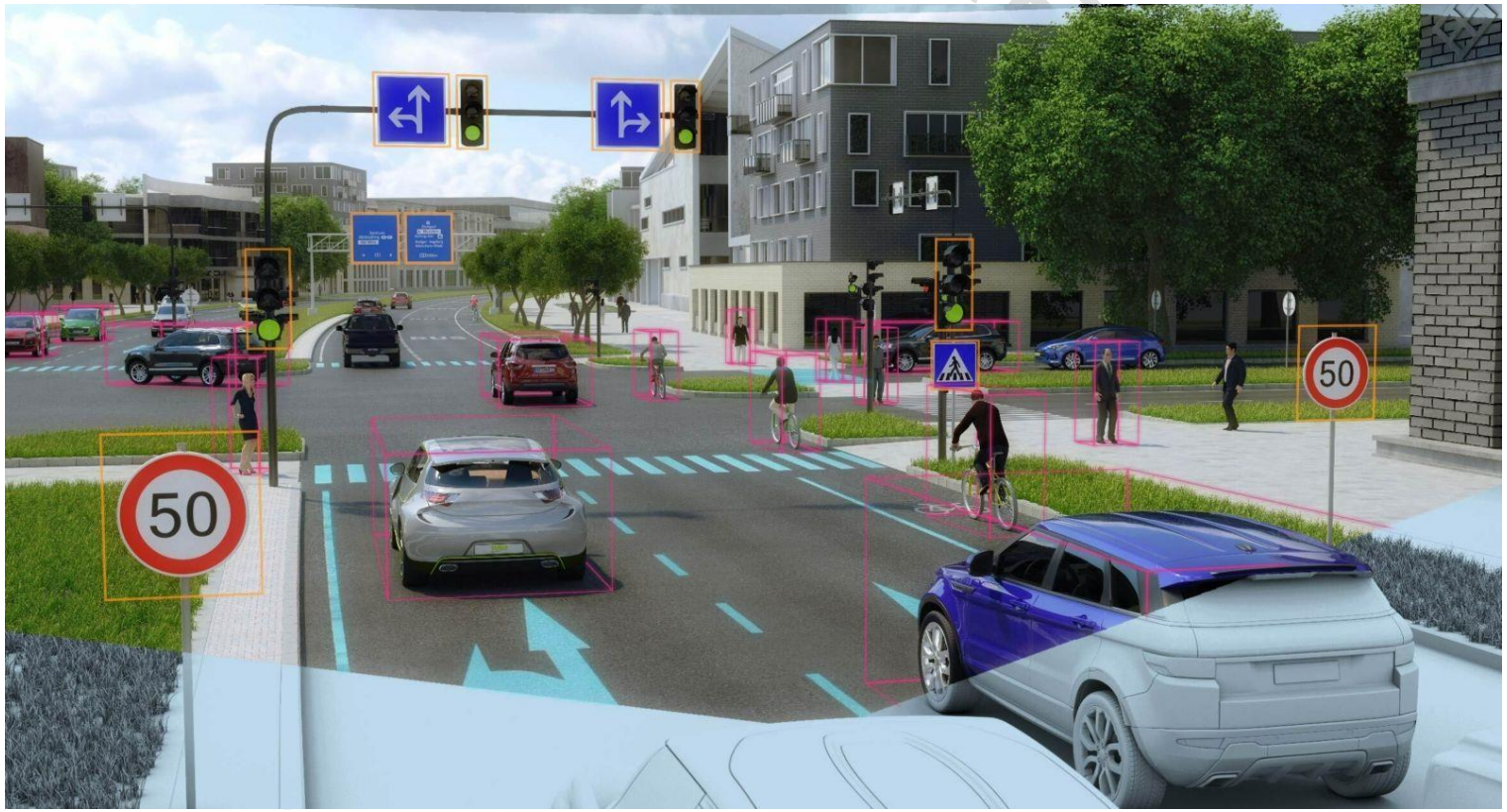
Hardware **I**n the **L**oop

Requirements from the Customer

Requirement 1 from customer

The Ego Car has to stops when Software detects pedestrian or object detects on its way prioritize to the sign board

- Brake action should **activate** to stop the vehicle to avoid accident



Requirement 2 from customer

Car should holds for 4 seconds in its position when it detects the road inclination upwards and driver releases complete accelerator pedal

Understand here !!

What is the advantages of with and without HIL testing

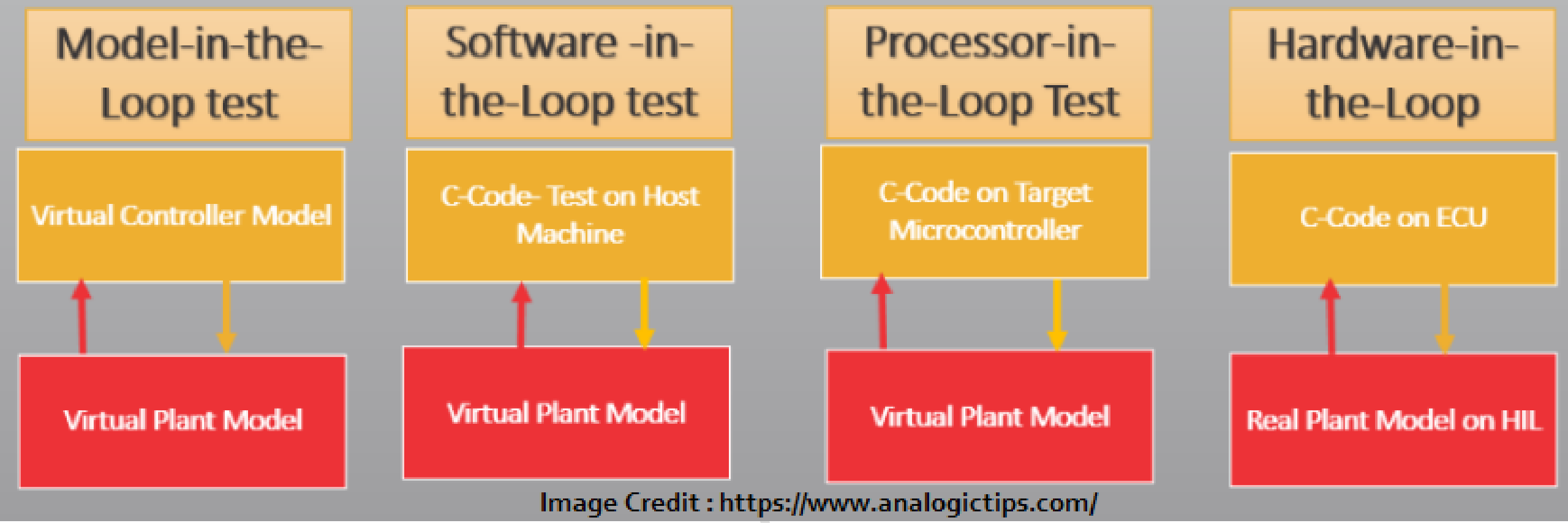


Image Credit : https://www.analogictips.com/

Hardware-in-the-loop simulation

- Hardware-in-the-loop (HIL) simulation is a technique that is used in the development and test of complex real-time embedded systems.
- HIL simulation provides an effective platform by adding the complexity of the plant under control to the test platform.
- The complexity of the plant under control is included in test and development by adding a mathematical representation of all related dynamic systems.
- These mathematical representations are referred to as the "plant simulation". The embedded system to be tested interacts with this plant simulation.
- In Otherwords, HIL testing is a technique where real signals from a controller are connected to a test system that simulates reality, making the controller to work as its assembled product

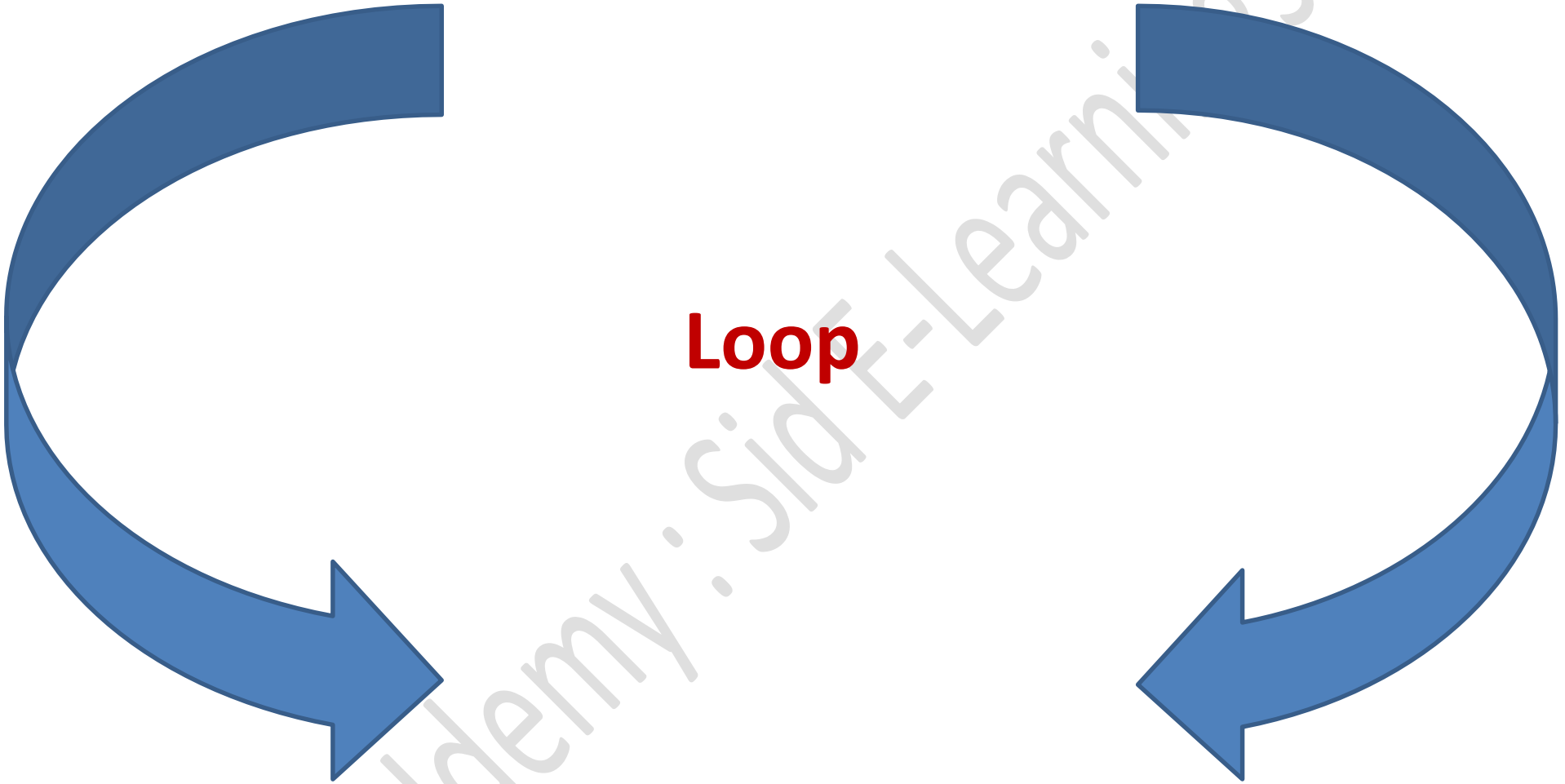

ADAC


ADAC


DYN44


DYN44

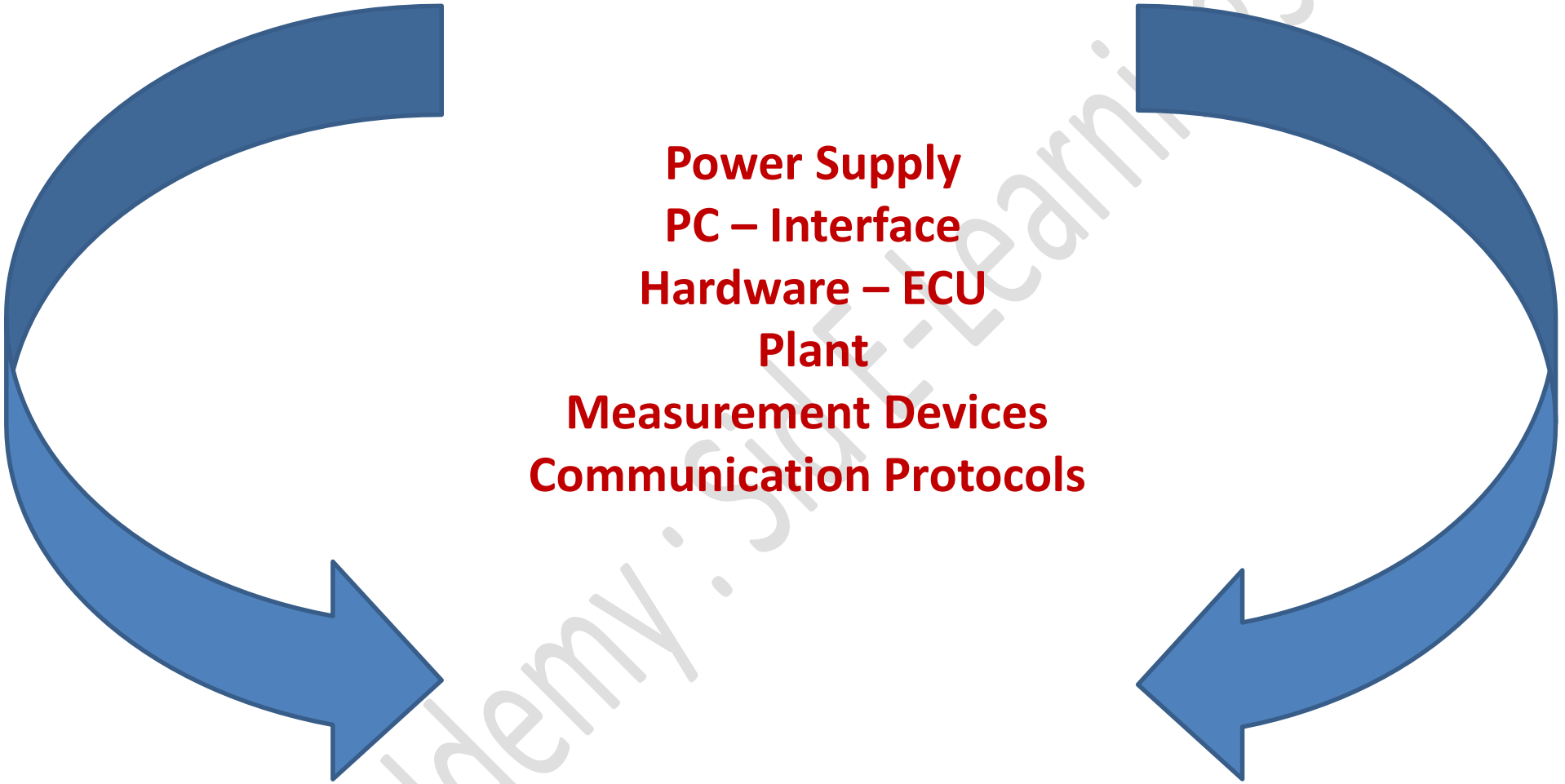
Hardware in the Loop



Hardware In the Loop

Hardware

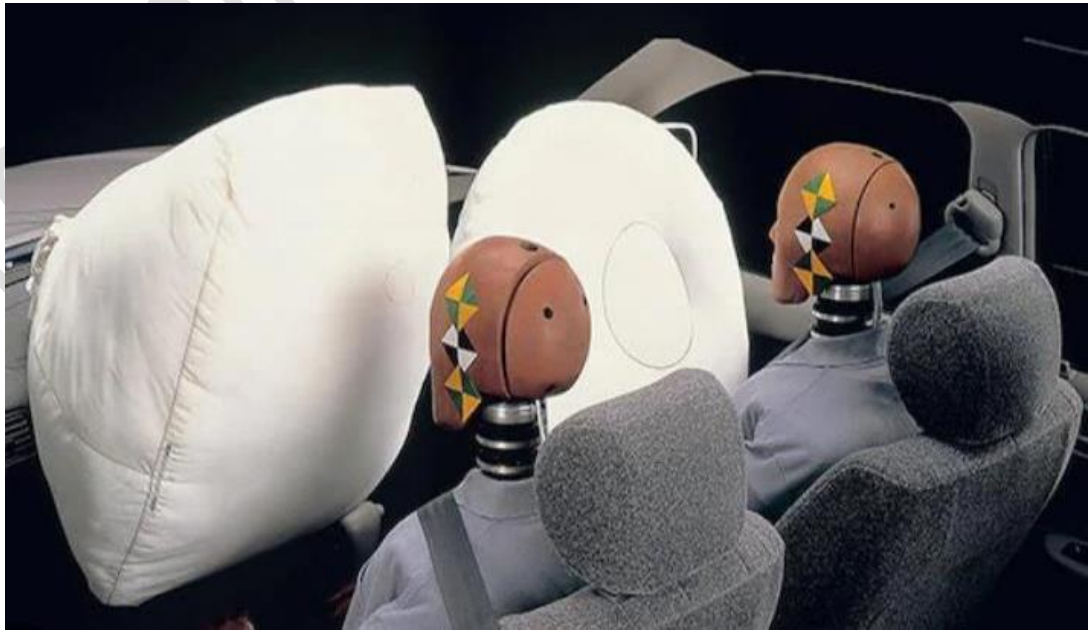
What Loop Contains ?



Power Supply
PC – Interface
Hardware – ECU
Plant
Measurement Devices
Communication Protocols

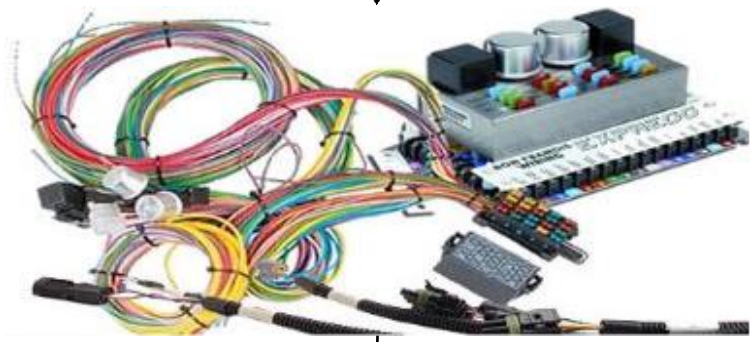
To be Continued


OFF
INT
ON
REAR

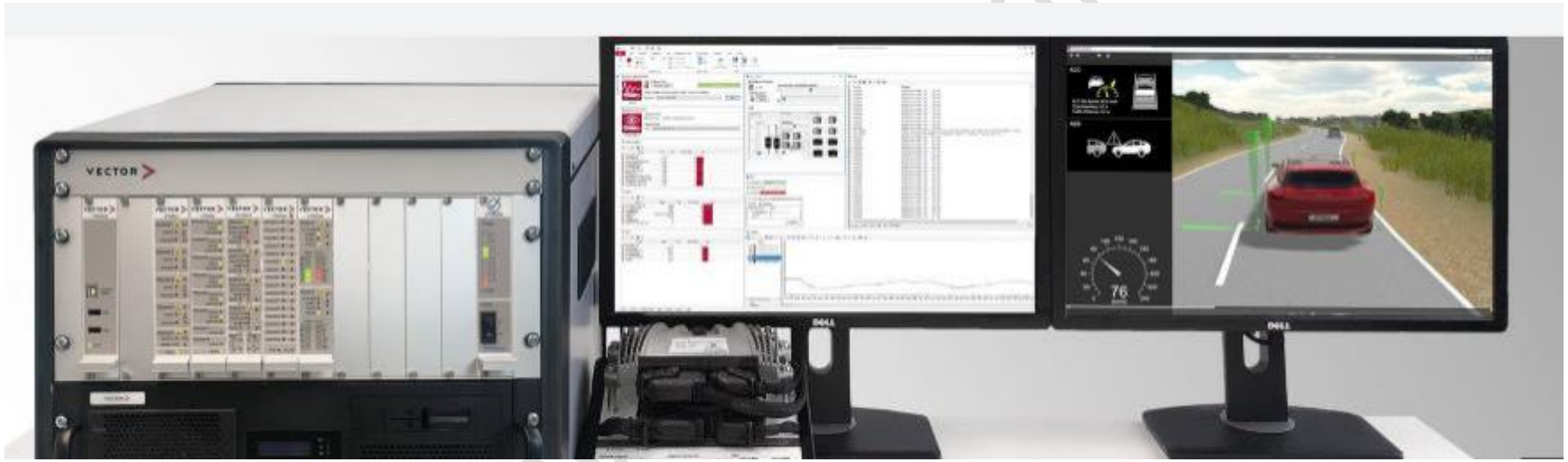


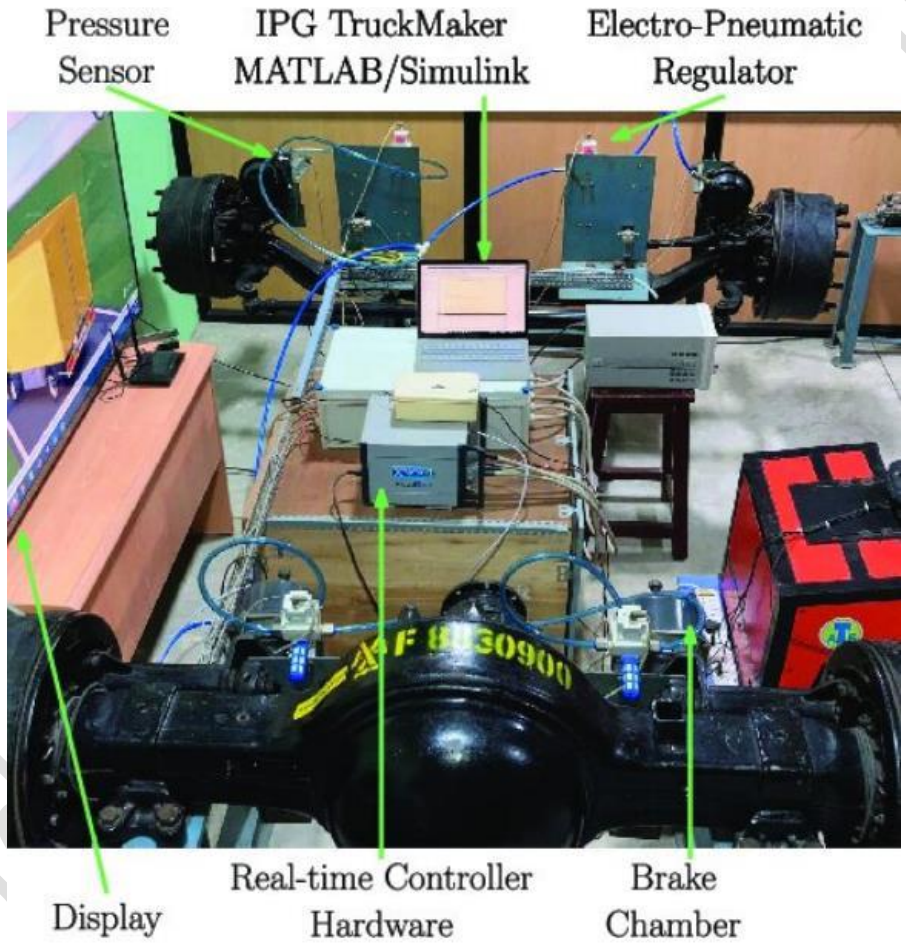
Plant

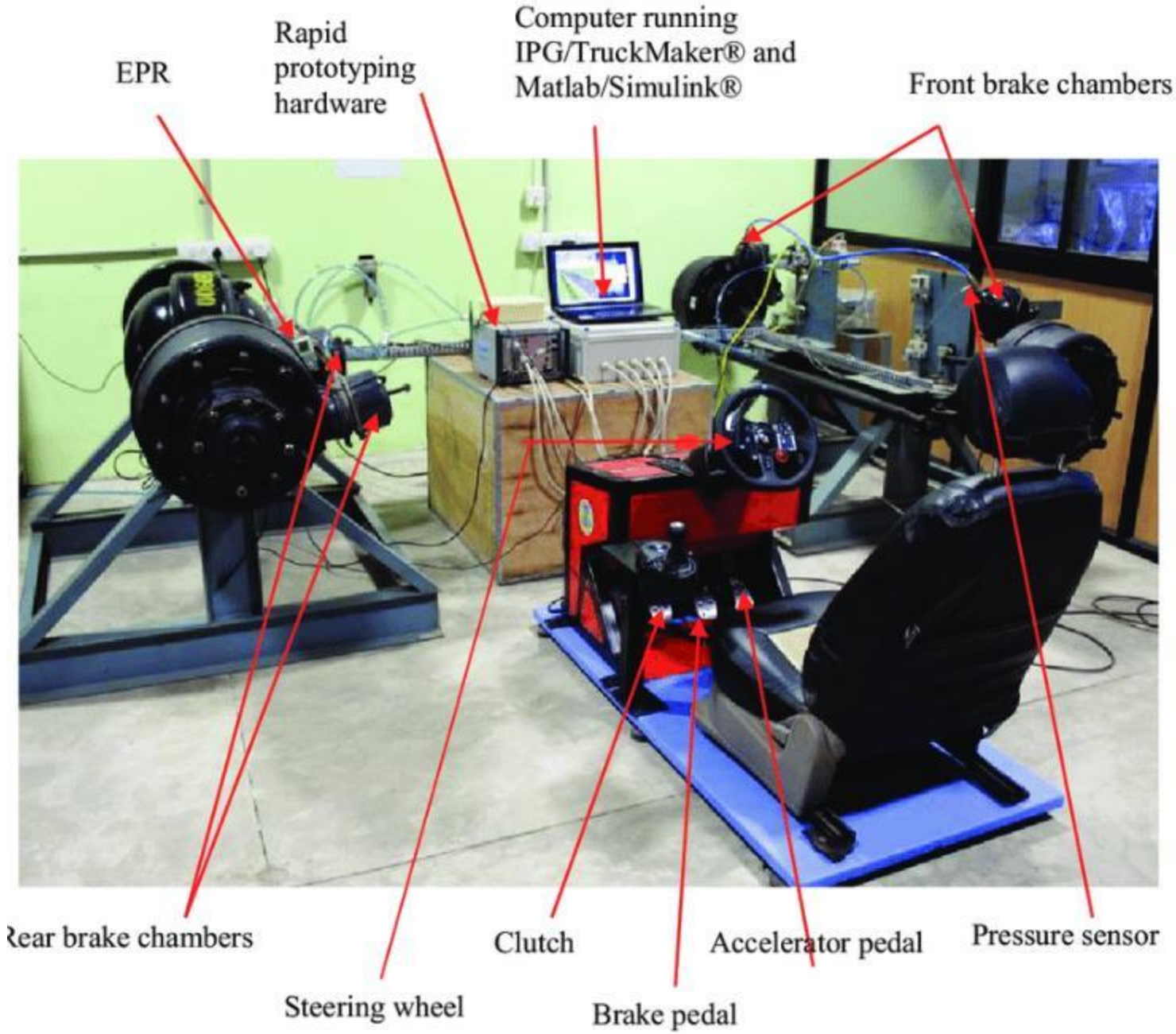
(CAN, LIN, Flexray or OEM Specific)




DH-447-DF
DELL




Pressure Sensor
IPG TruckMaker MATLAB/Simulink
Electro-Pneumatic Regulator
Display
Real-time Controller Hardware
Brake Chamber


EPR
Rapid prototyping hardware
Computer running IPG/TruckMaker® and Matlab/Simulink®
Front brake chambers
Rear brake chambers
Steering wheel
Clutch
Brake pedal
Accelerator pedal
Pressure sensor

Part - 2

Introduction

Hardware-in-the-loop simulation

- **Hardware-in-the-loop** (**HIL**) **simulation** is a technique that is used in the **development** and **test** of complex real-time embedded systems.
- HIL simulation provides an effective platform by adding the **complexity of the plant** under control to the test platform.
- The complexity of the plant under control is included in test and development by adding a **mathematical representation** of all related dynamic systems.
- These mathematical representations are referred to as the **"plant simulation".** The embedded system to be tested interacts with this plant simulation.
- In Otherwords, HIL testing is a technique where real signals from a controller are connected to a test system that **simulates reality**, **making the controller to work as its assembled product**

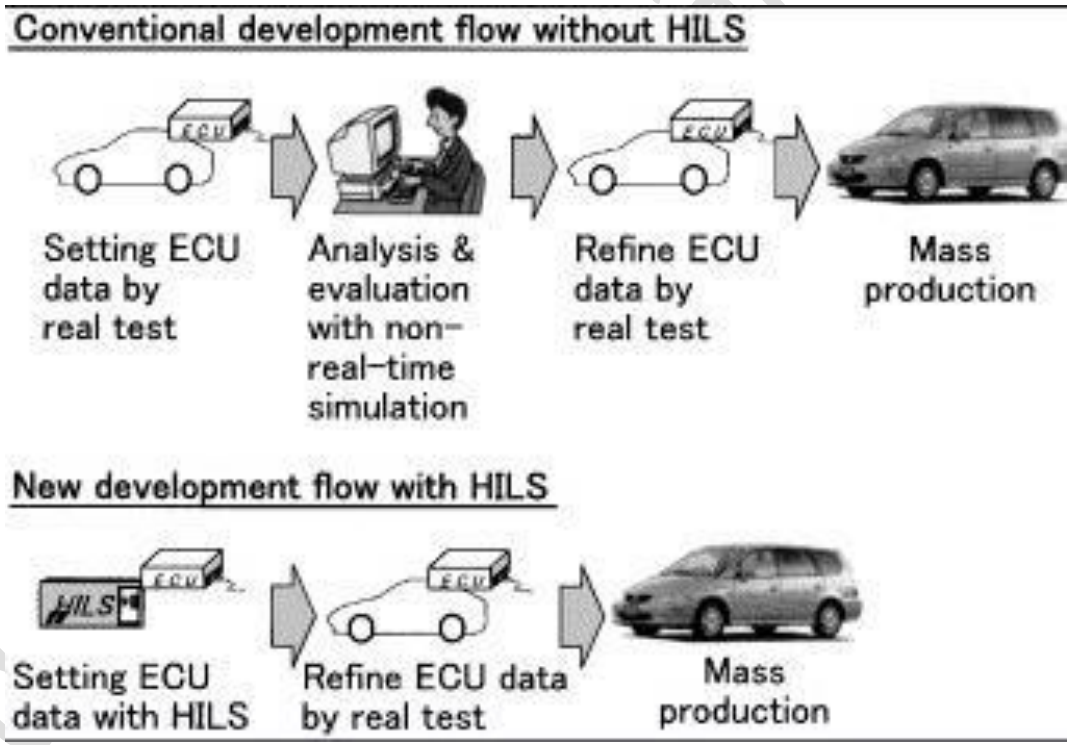
Why HIL

Supplier / OEM has developing software for braking component (Say : ABS),

After development is it recommended to validate in real car without HIL ?

Why HIL

After development is it recommended to validate in real car ?



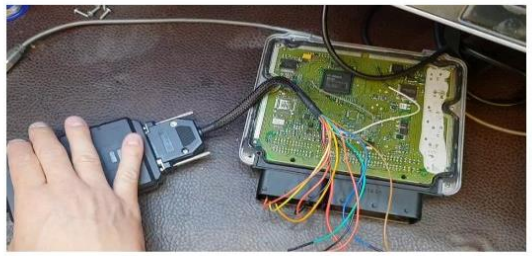
Why HIL

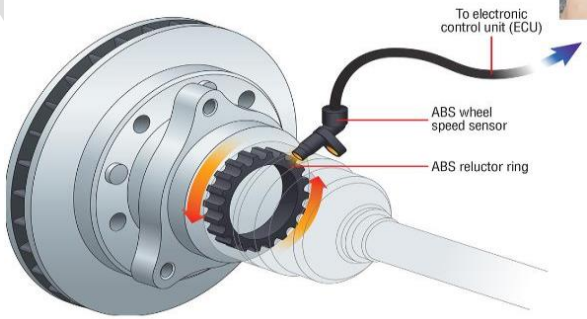
Supplier / OEM has developing software for braking component (Say : ABS),

After development is it recommended to validate in real car ? No

Basic Input / Output understanding in HIL

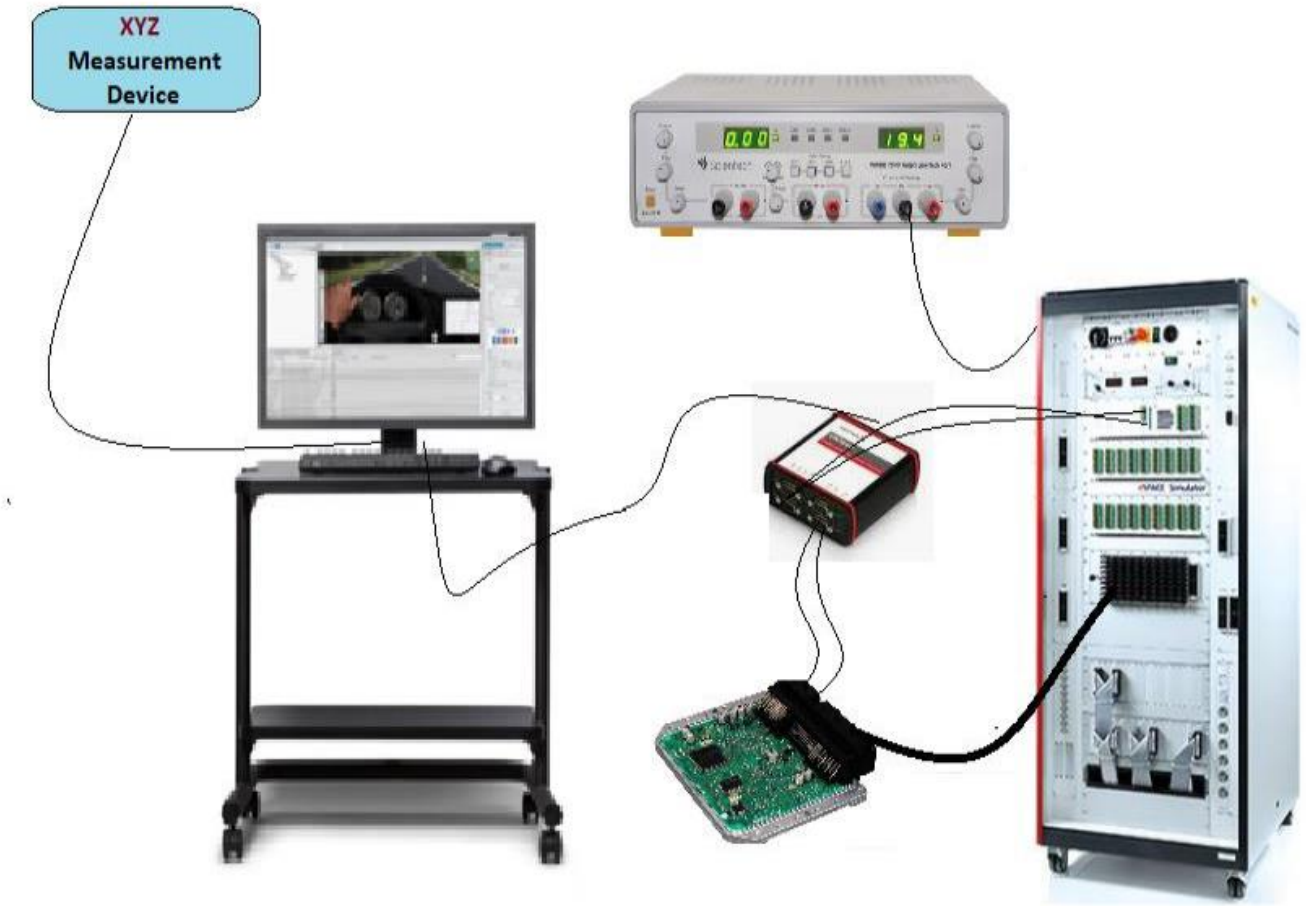
- Consider any Embedded System they work with **Real time Inputs** and also Control **output device or peripheral**.
- In general the input comes from the **Sensors** and **Output Devices** which the System Control are called **Actuators**.
- The course of Action over the Actuator is done based on Sensor input by the Control Algo





Components in HiL Testing

Components & HIL Set up



1. ECU [Electronic Control Unit]
2. I/O Interface
3. Supported Software's / Tools
4. PC
5. Plant
6. Sensors & Actuators
7. Communication Protocols [CAN, LIN, Eth, Flexray, MOST]
8. Battery Simulation [Power Supply] or Programmable Power Supply [PPS]
9. Wiring Harness for Power Supply

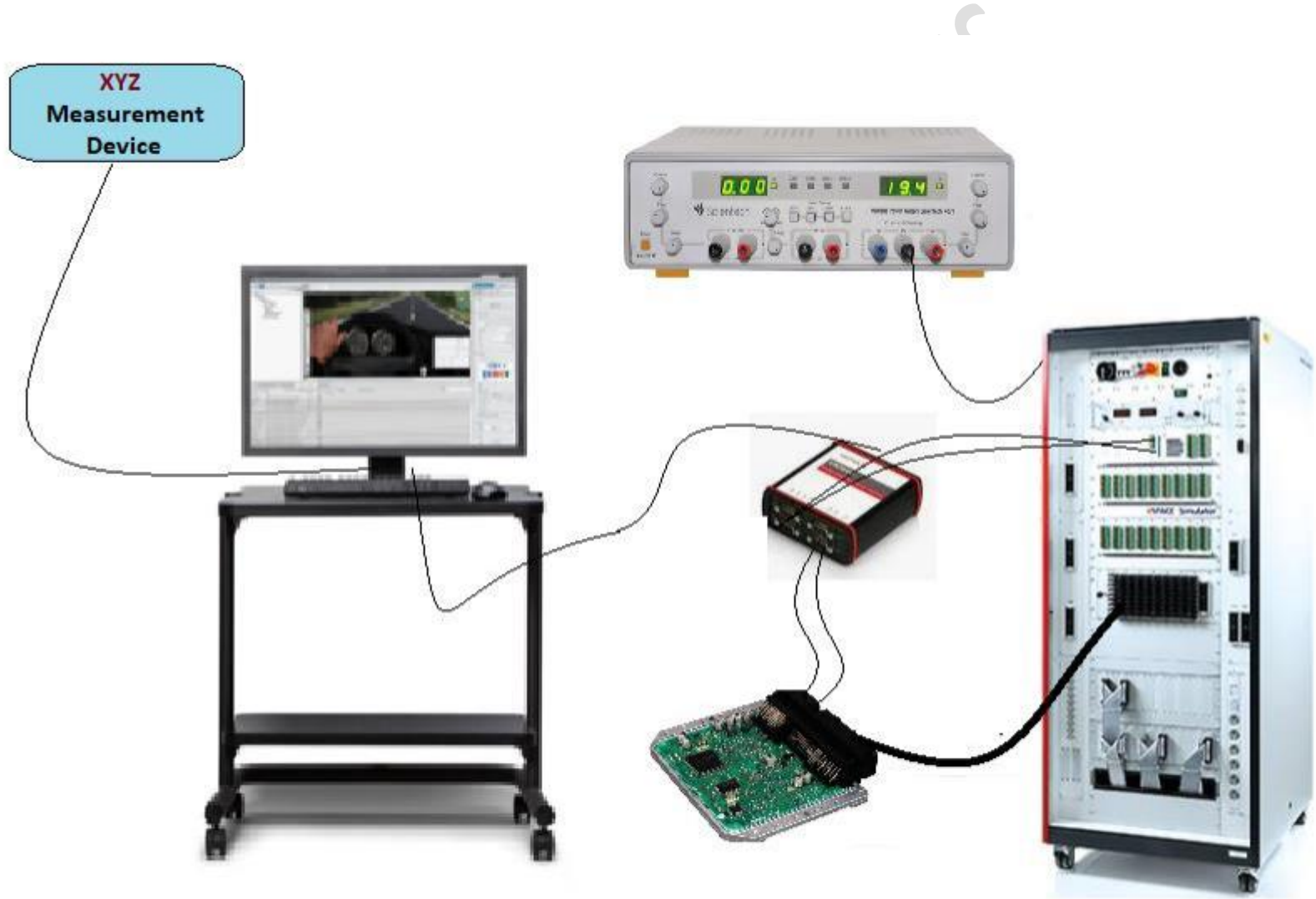
Next Tutorial

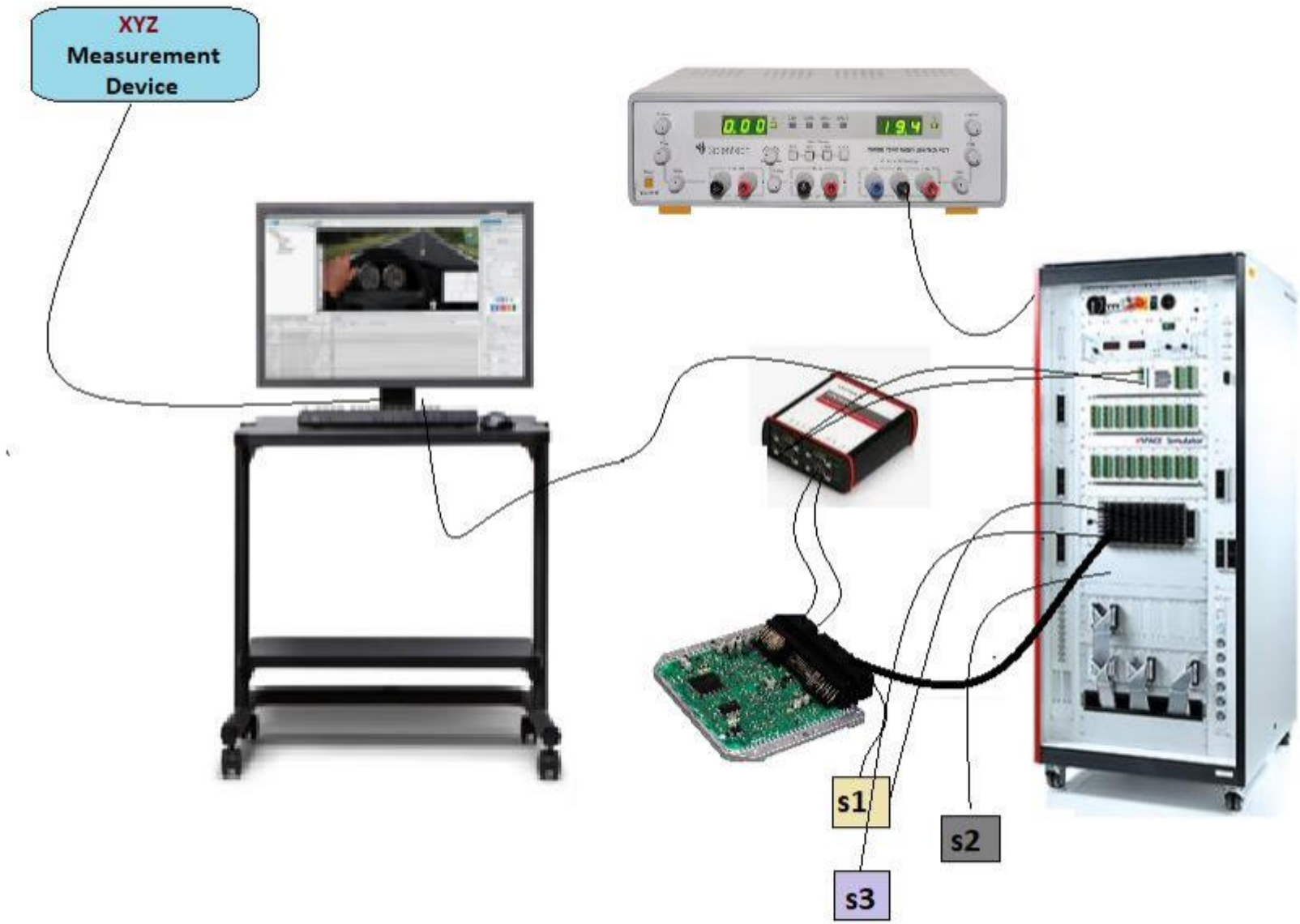
Part – 3

Components in the HIL

Components in the HIL

Part - 3


XYZ
Measurement
Device

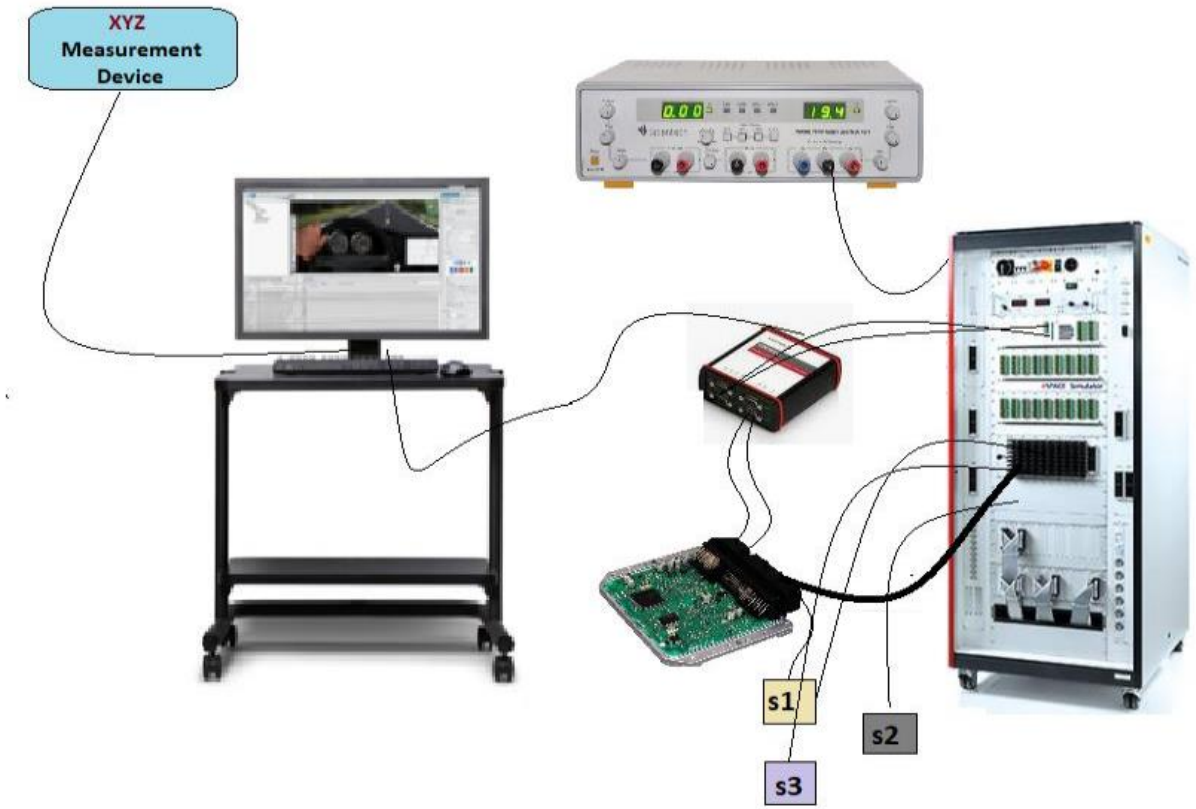

XYZ
Measurement
Device
s1
s2
s3

Next Video Tutorial : Part - 4

Complete workflow in the HIL

Complete workflow in the HIL

Part - 4



- Make **Hardware connections ready**
- Make **fault free** system
- **Input** given in **Plant simulation by Plant Model**
- Plant model sends the input to **ECU**
- ECU Process the signal and respond as output by actuating **actuators**
- The whole communication happens **in CAN Protocol**
- For all these **HIL Components power supply** is required with respect to individual components need

Next Tutorial

Part – 5

HIL Components

All about ECU

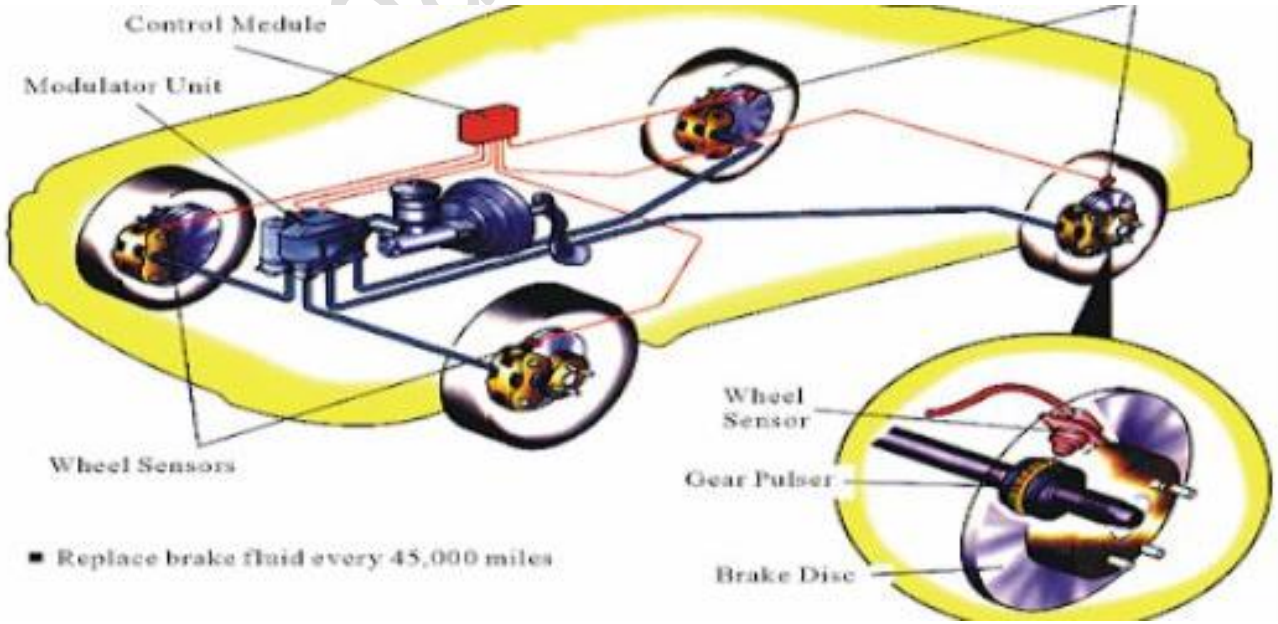
HIL Testing

Part – 5

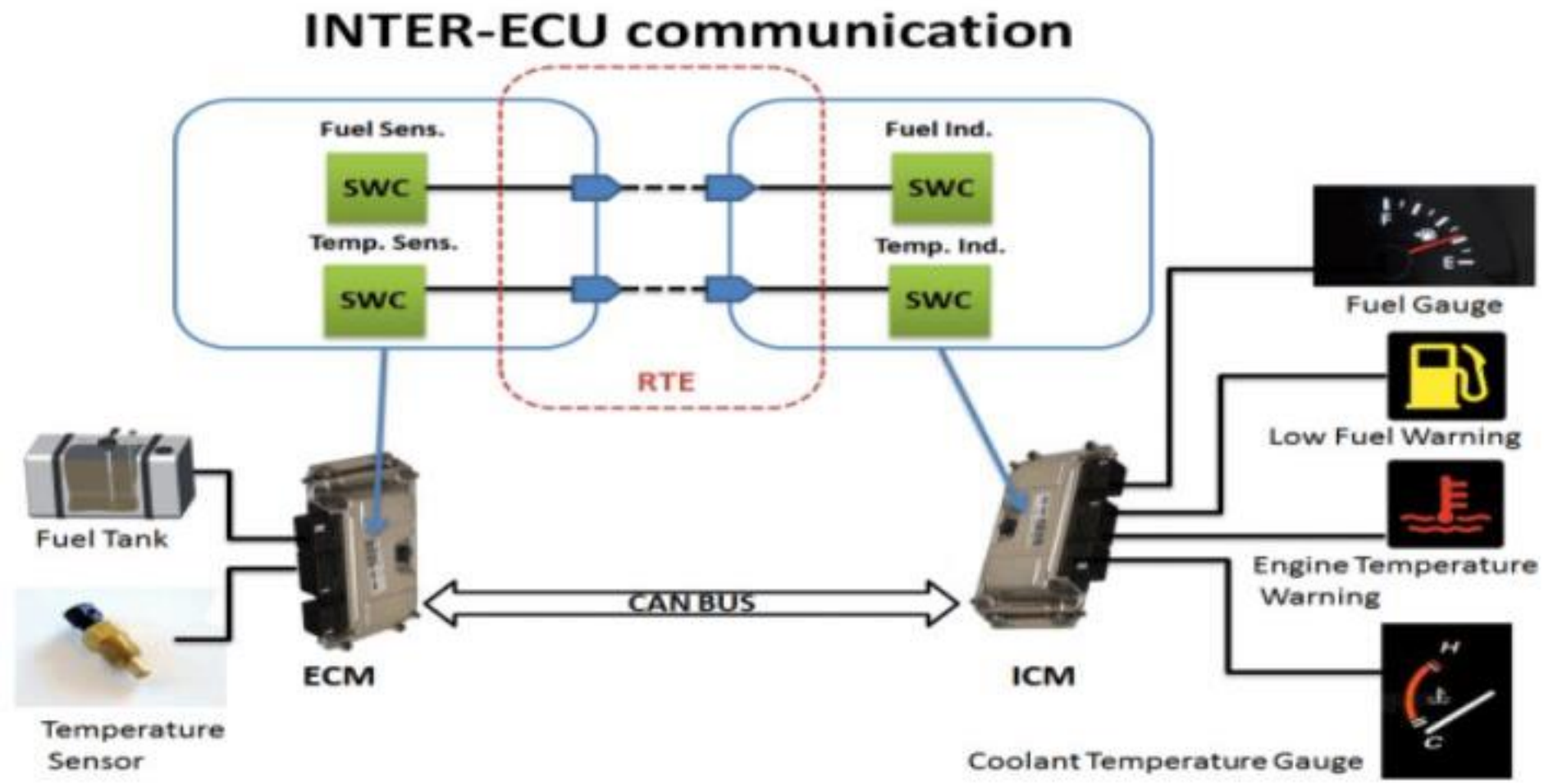
HIL Components

ECU

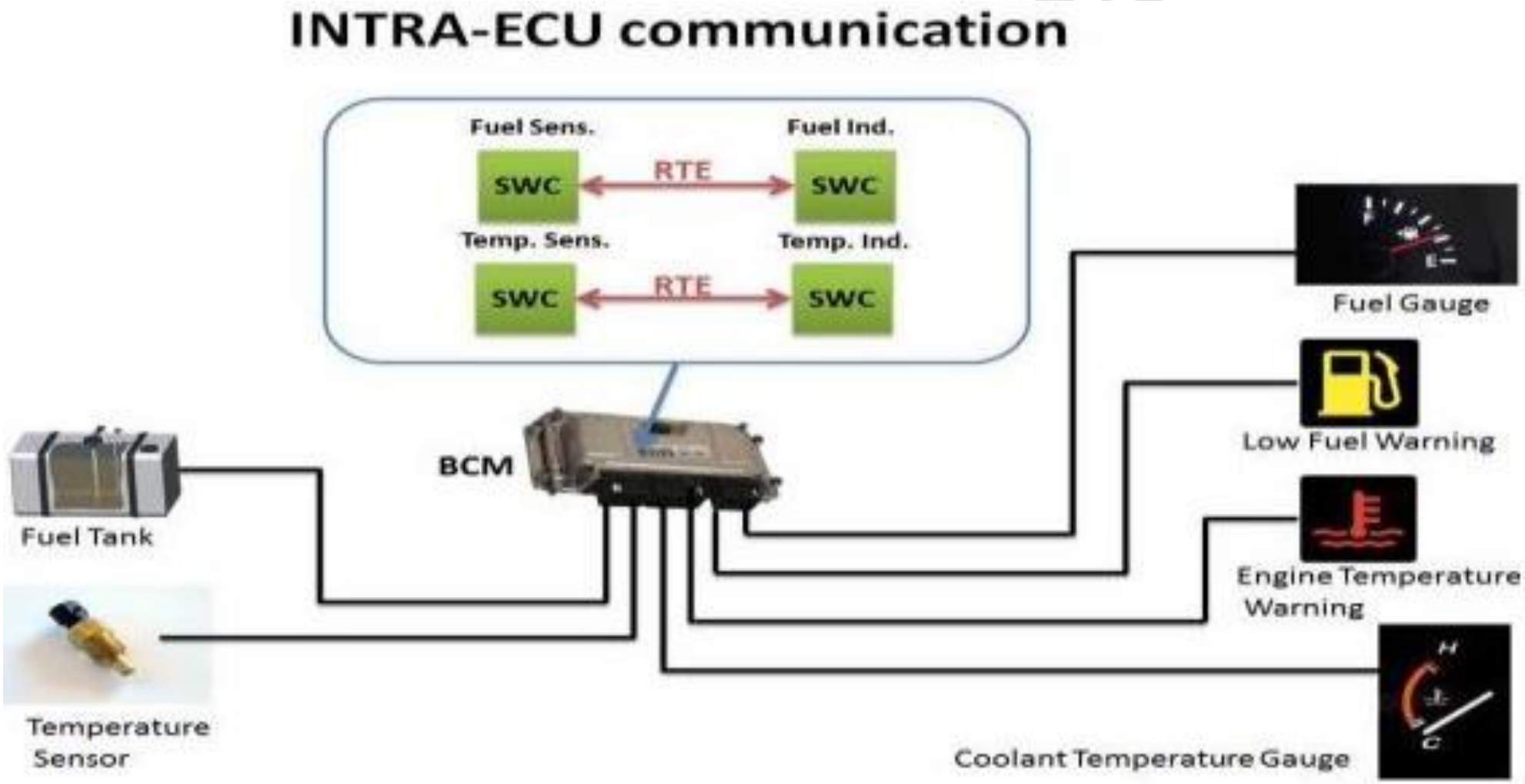
Engine Control Unit / Electronic Control Unit



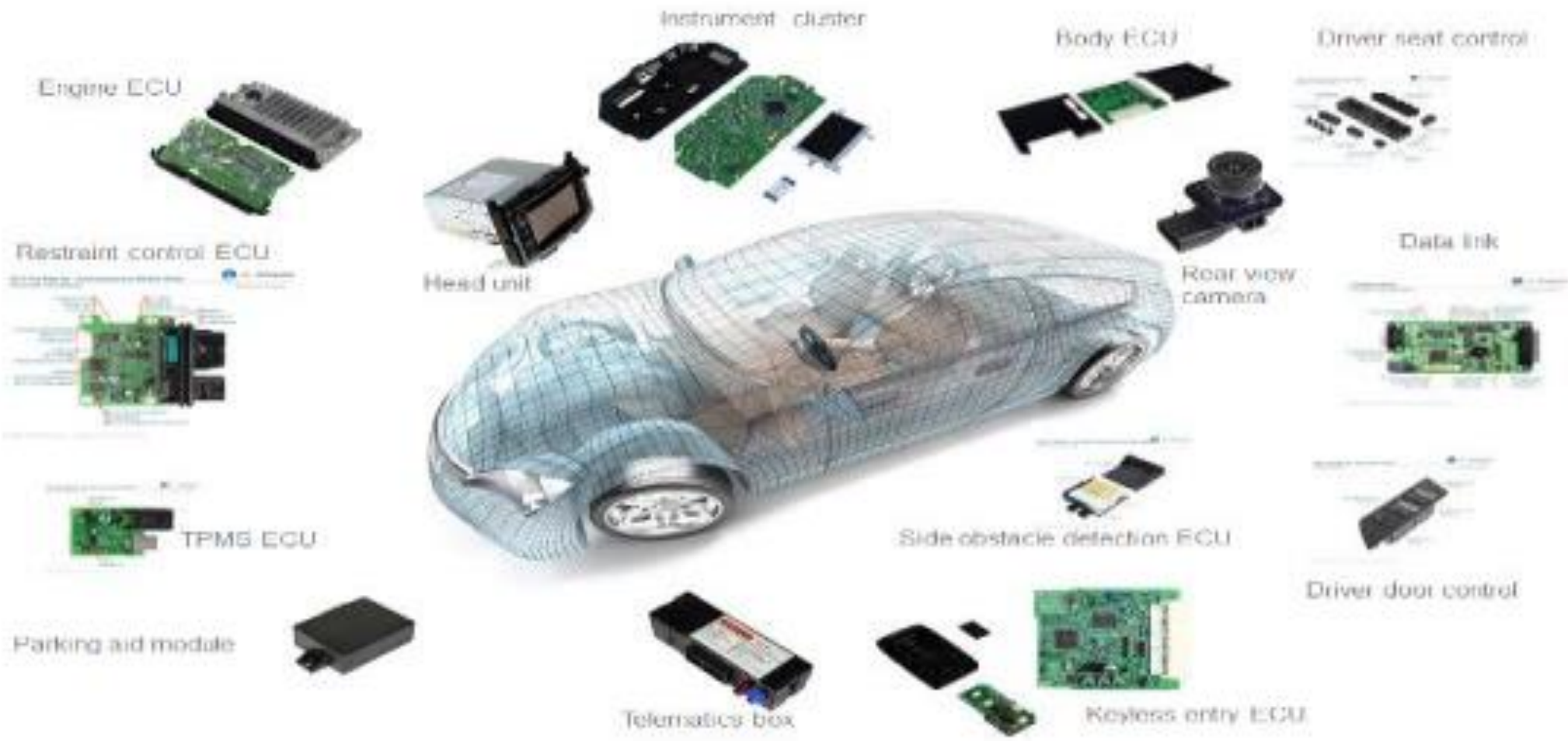
Inter – ECU Communication



Intra – ECU Communication



Various ECUs in CAR



Various ECUs in Car


Check Surroundings For Safety

Start with ECU

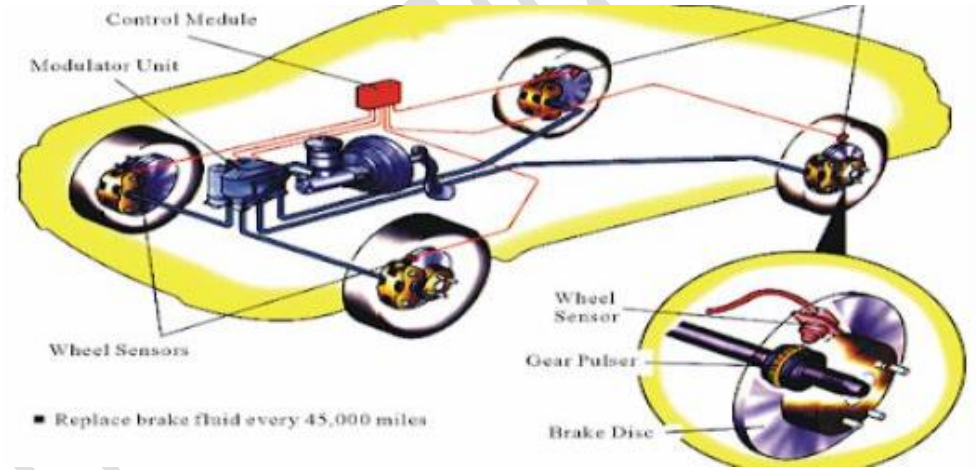
1. Develop the Software (Code)
2. **Testing**
   - Role of ECU in Testing
   - Flashing the Software into ECU
   - Validate the functionalities
3. Types of ECU
4. ECU Architecture
5. How ECU Interacts with Plant Simulator

Start with ECU

1. Develop the Software (Code)
2. Testing
   - **Role of ECU in Testing**
   - Flashing the Software into ECU
   - Validate the functionalities
3. Types of ECU
4. ECU Architecture
5. How ECU Interacts with Plant Simulator

Role of ECU in Testing

- ECU is the component which makes car to work **embed with software**.
- So the software developed can be flashed into the ECU and **testable in HIL Environment**.
- Literally it is called as **Device Under Test (DUT)**



Start with ECU

1. Develop the Software (Code)
2. Testing
   - **Role of ECU in Testing**
   - **Flashing the Software into ECU**
   - Validate the functionalities
3. Types of ECU
4. ECU Architecture
5. How ECU Interacts with Plant Simulator

Flashing the Software into ECU

1. Develop the Software (Code)
2. Testing
   - **Role of ECU in Testing**
   - **Flashing the Software into ECU**
   - Validate the functionalities
3. Types of ECU
4. ECU Architecture
5. How ECU Interacts with Plant Simulator

Flashing the Software into ECU

- ECU Flashing is the **process of updating** the software that runs in Cars ECU.
- Flashing allows user to change various maps and settings in Car ECU and significantly improve the performance of the engine.
- Re-flashing is a tuning technique that is also often called as **flashing, remapping, or flash tuning**. Like any computer, your car's ECM has software with different settings or parameters, and these can be changed to alter the **performance and driving limits** of a vehicle.
- **Re-flashing or remapping** an engine computer is essentially just the process of replacing the existing software in a vehicle controller with new software.

Types of Flashing

- In General, We have two ways of flashing,
  - » **Flashing Via External flash Drivers**
  - » **Flashing Via Application Protocols**

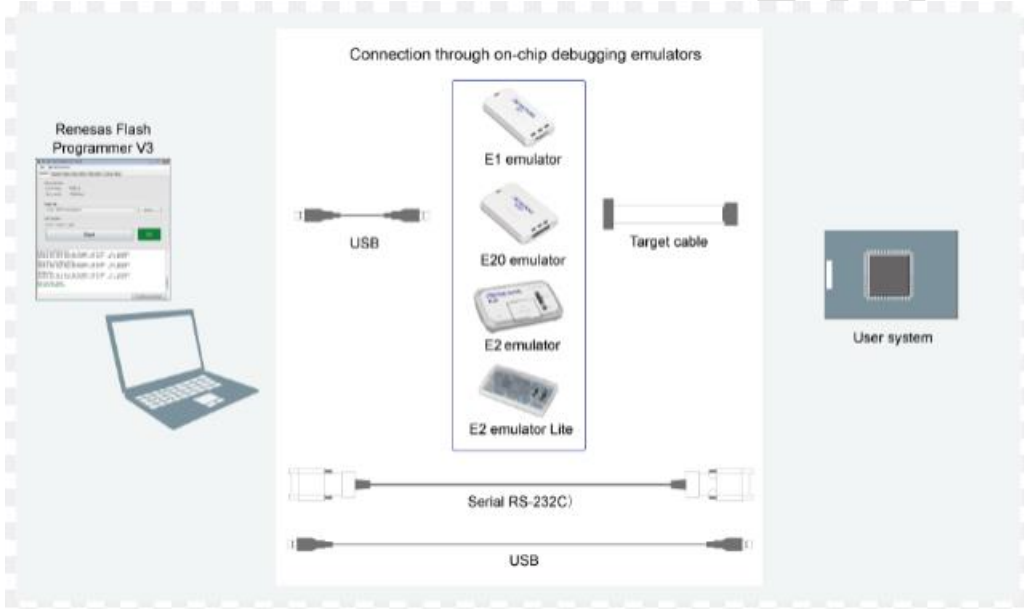
- The ECU flash can be segmented into two main areas:
  - **Program flash** where the **entire software code resides**, and is responsible for controlling logic, It is called as **PFlash**

    Ex: Entire Car Software
  - **Data flash** which is the area where the variables (such as **constants, maps and curves**) reside and are referred to by the software when needed. It is called as **DFlash**

    Ex : Re-flash for Maximum Speed

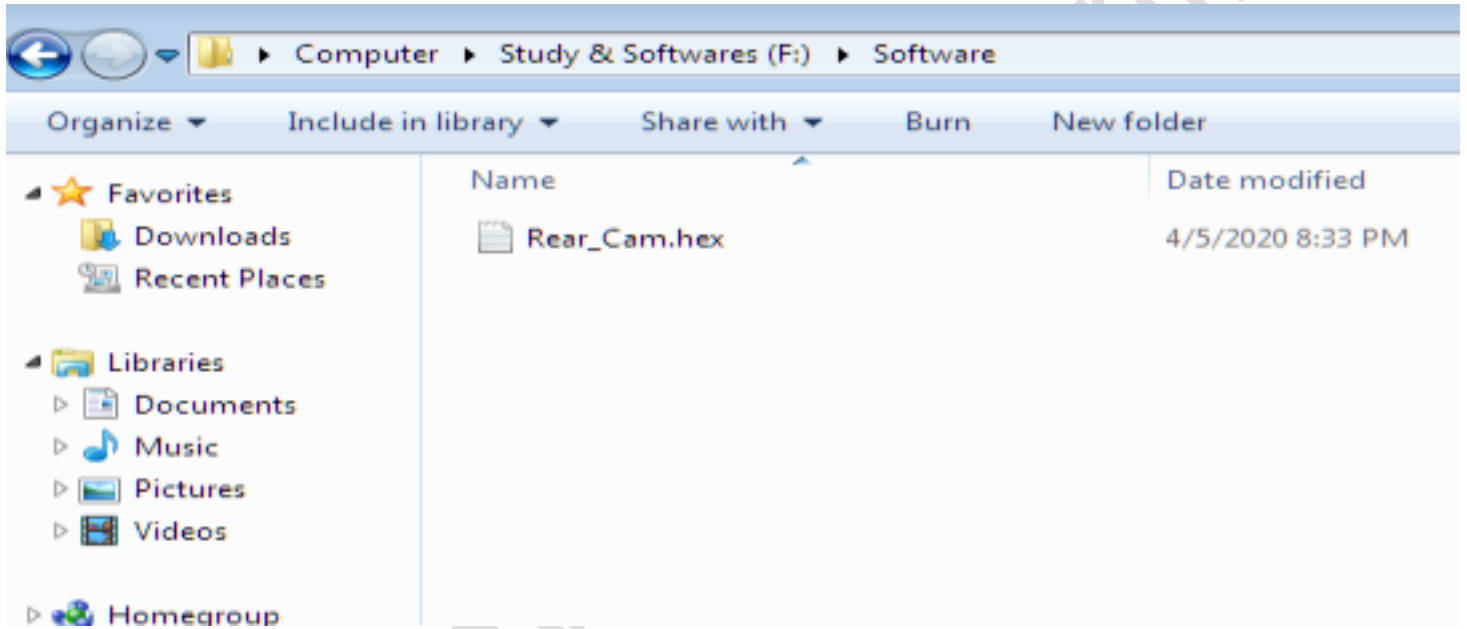
Flashing the Software into ECU

Flashing the Software into ECU

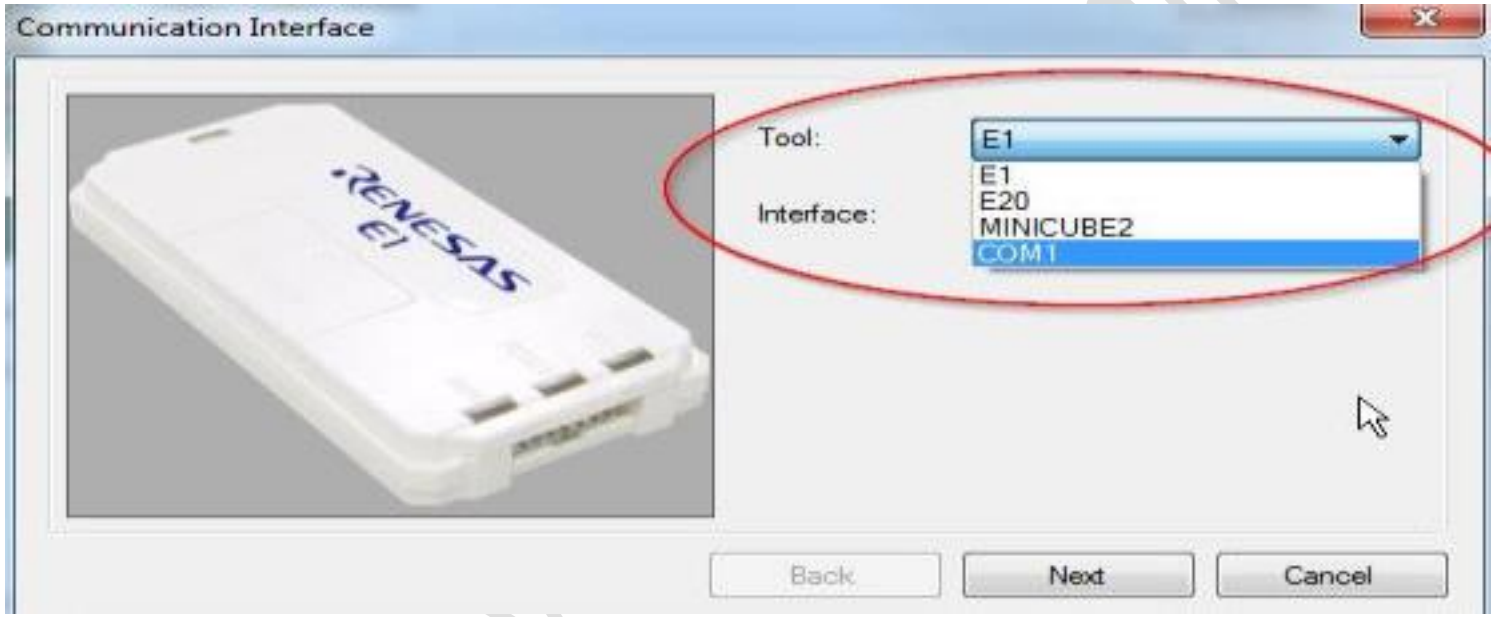


Flashing the Software into ECU

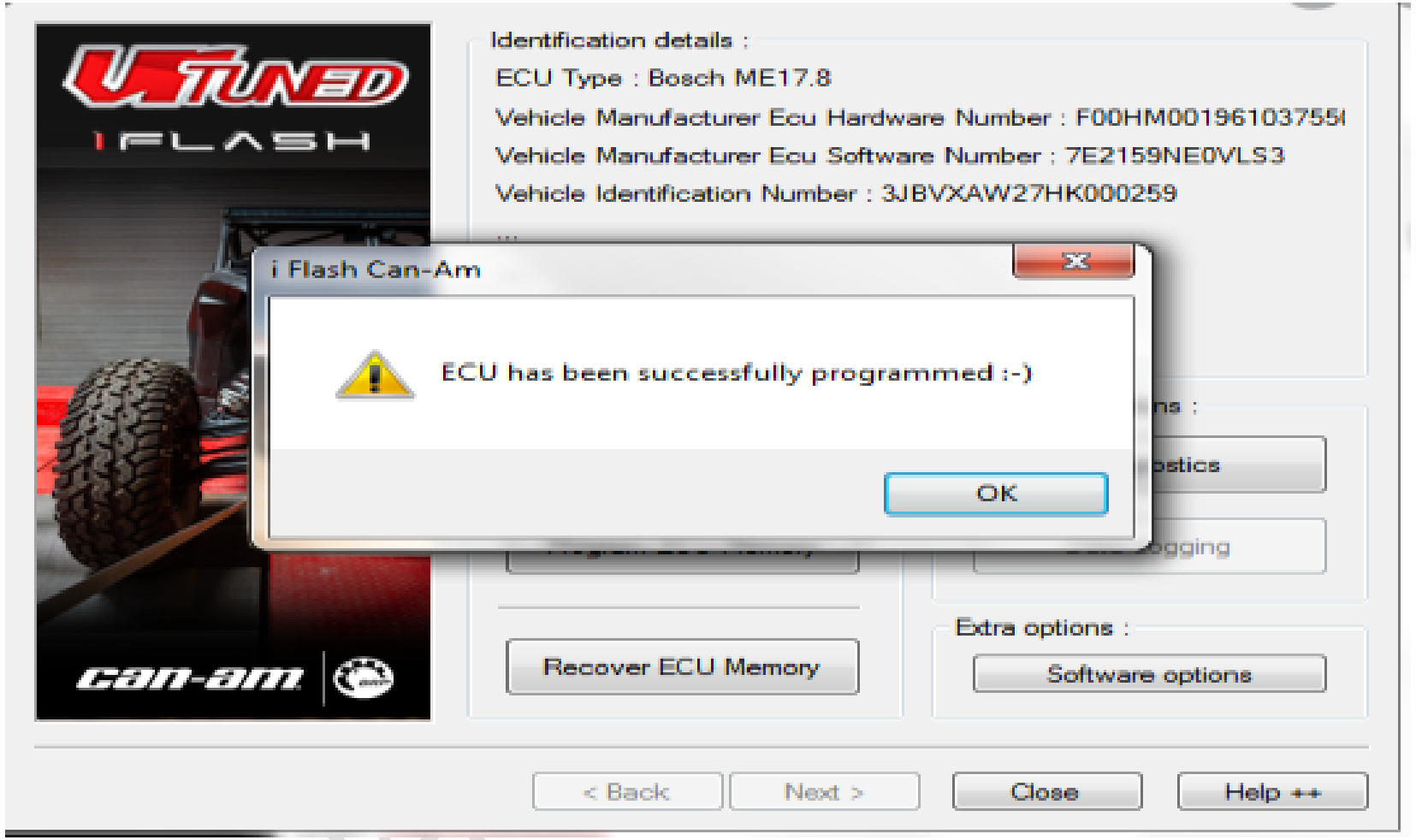
Flashing the Software into ECU



Flashing the Software into ECU



Flashing the Software into ECU



Types of ECU

There are two types of ECU

- Emulator ECU
- Attached ECU

Emulator ECU

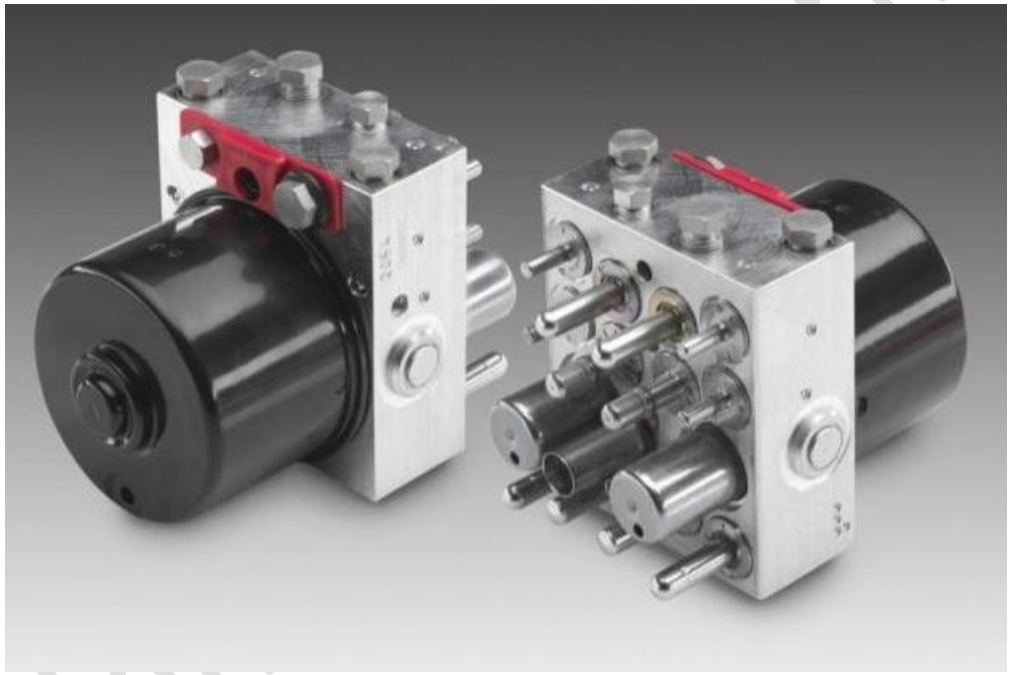
- Purely Integrated with PCB, ICs and other electronic components
- Approximately **80% price of conventional ECU**
- Recommended for less stable software

Attached ECU

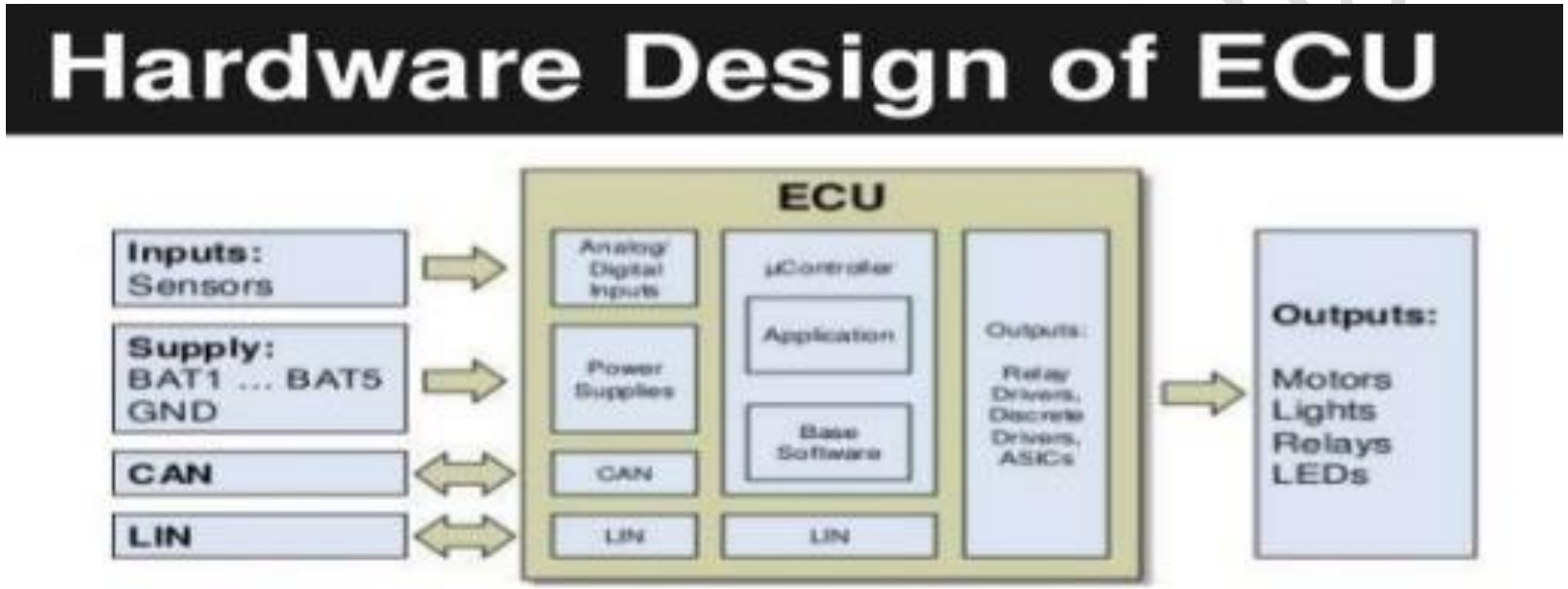
- Highly integrated VLSI Technology of Controller which embeds and integrates the hydraulic component with Electronic components
- Approximately **150% price of Conventional ECU**
- Recommended for highly stable software & the same hydraulic components integrate in Real Cars

Emulator ECU

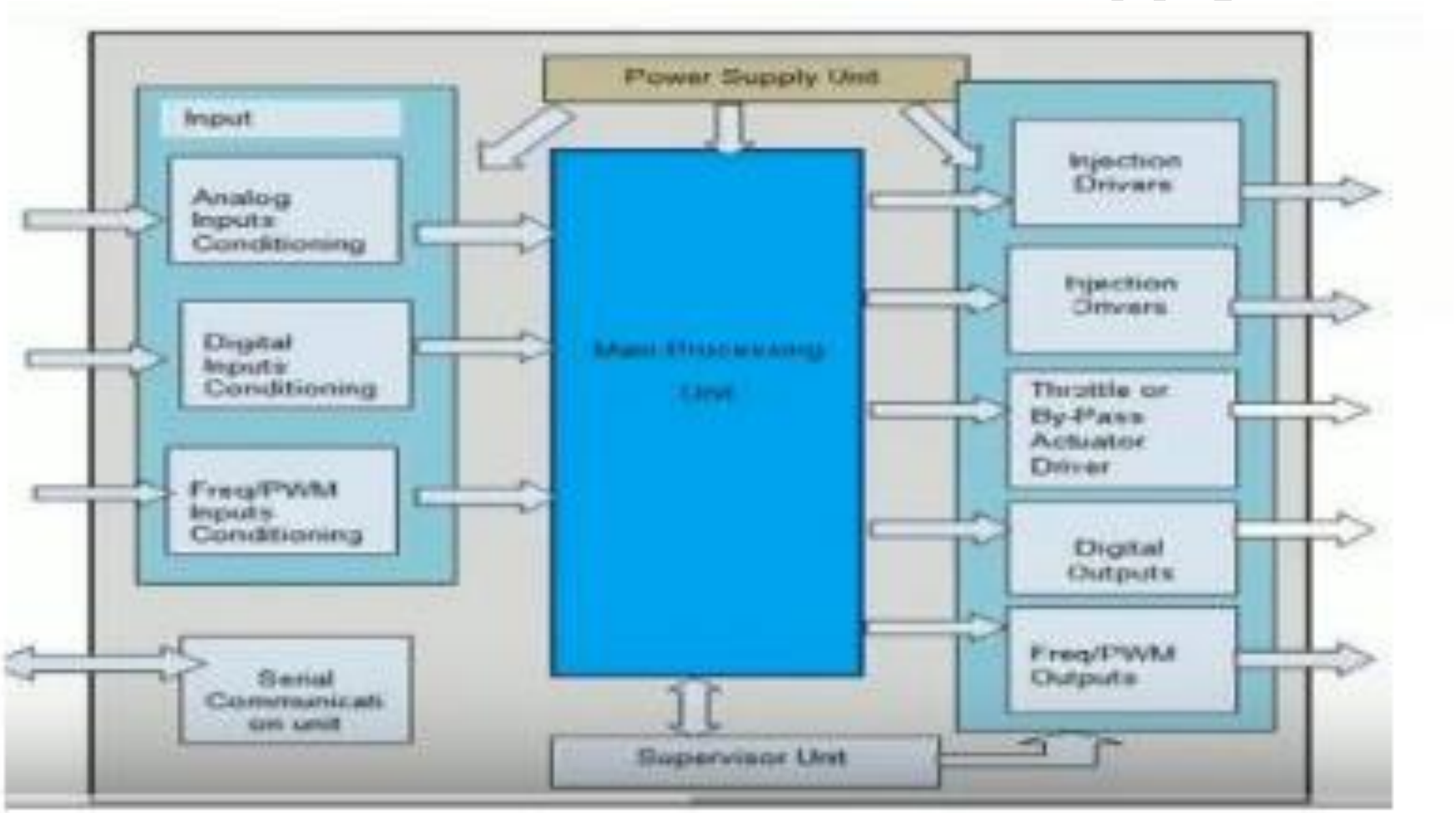
Attached ECU



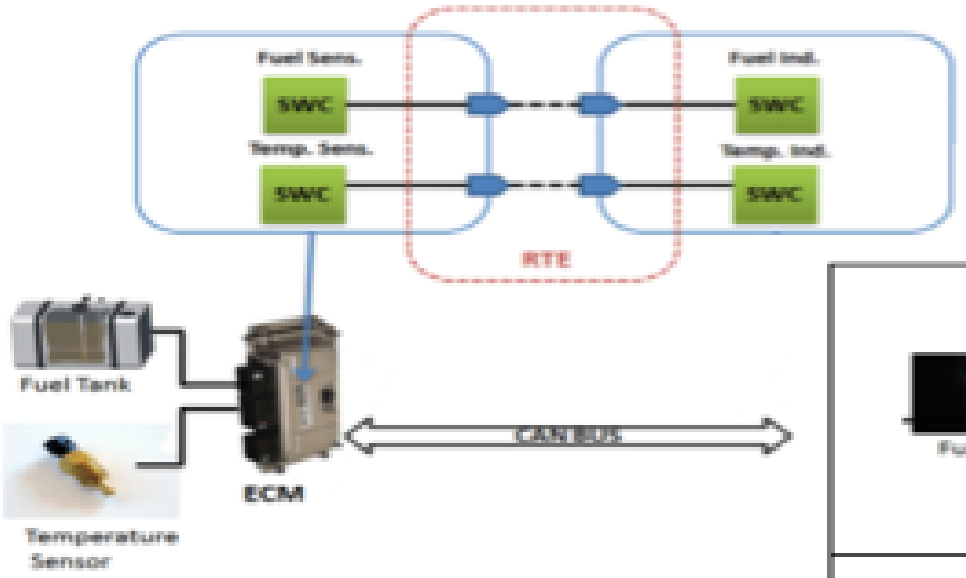
Hardware Design of ECU



Internal Architecture of ECU



ECU <-> Plant Simulator / Model



Next Tutorial

Plant Simulator

Next Tutorial

Part – 6

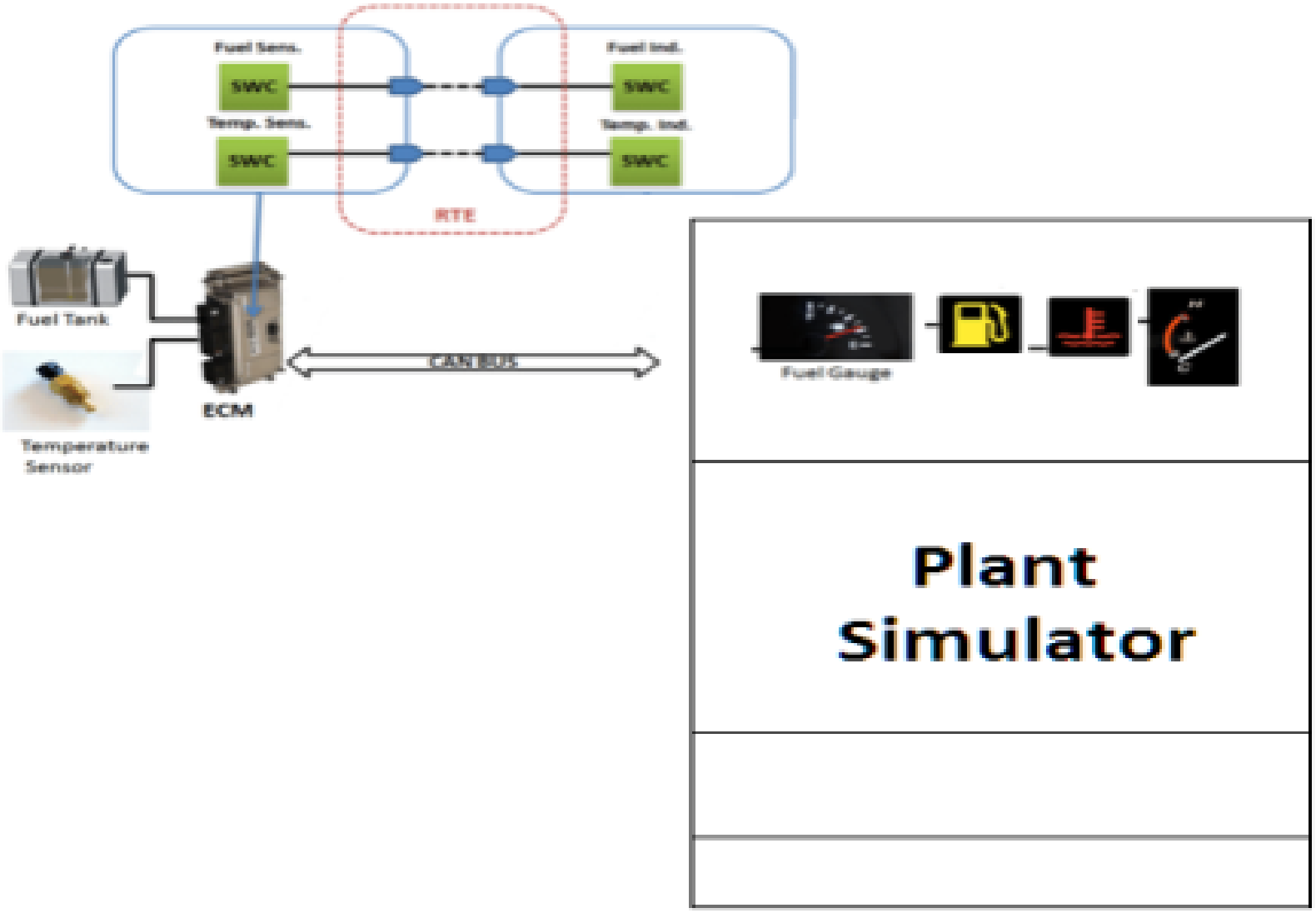
Plant Simulator & Plant Model

HIL Testing

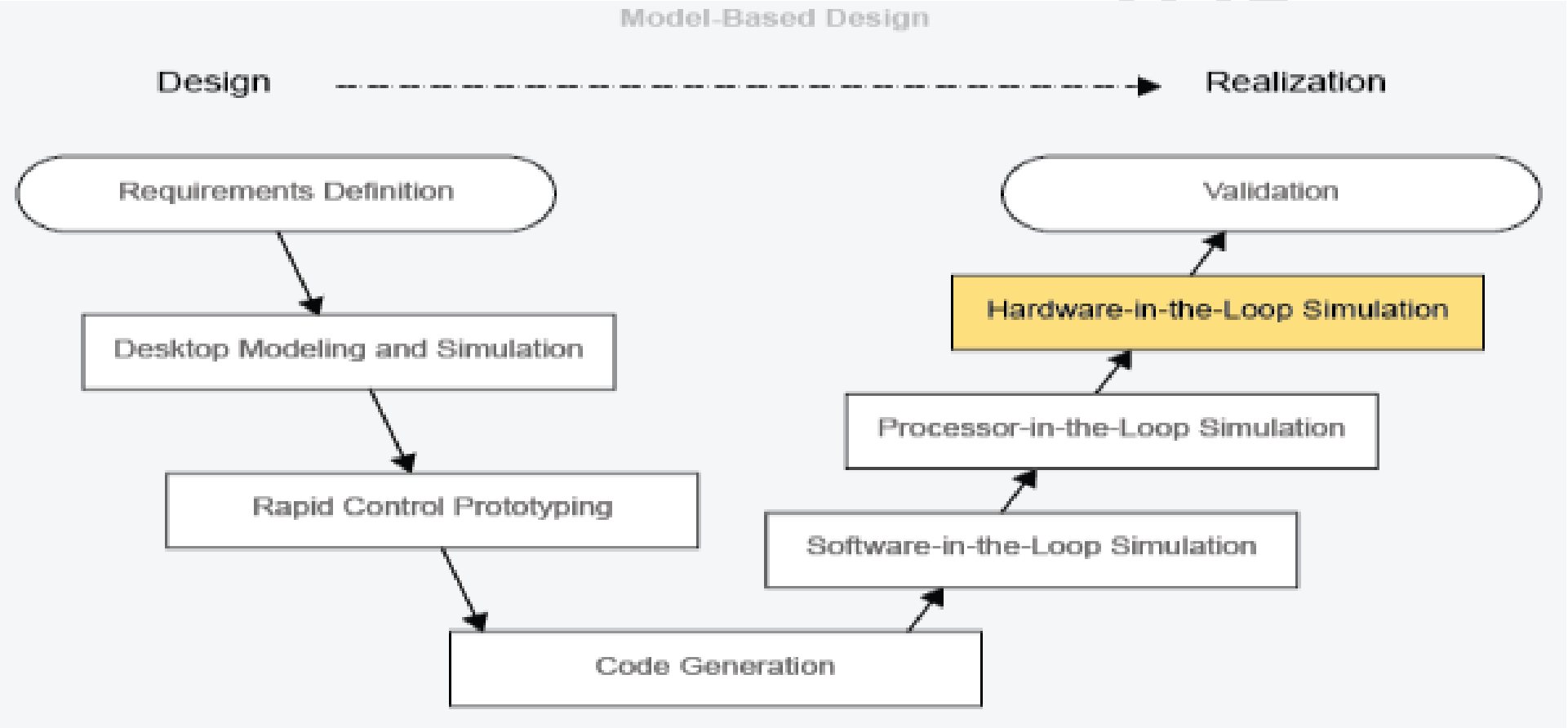
Part – 6

Plant Simulator & Plant Model

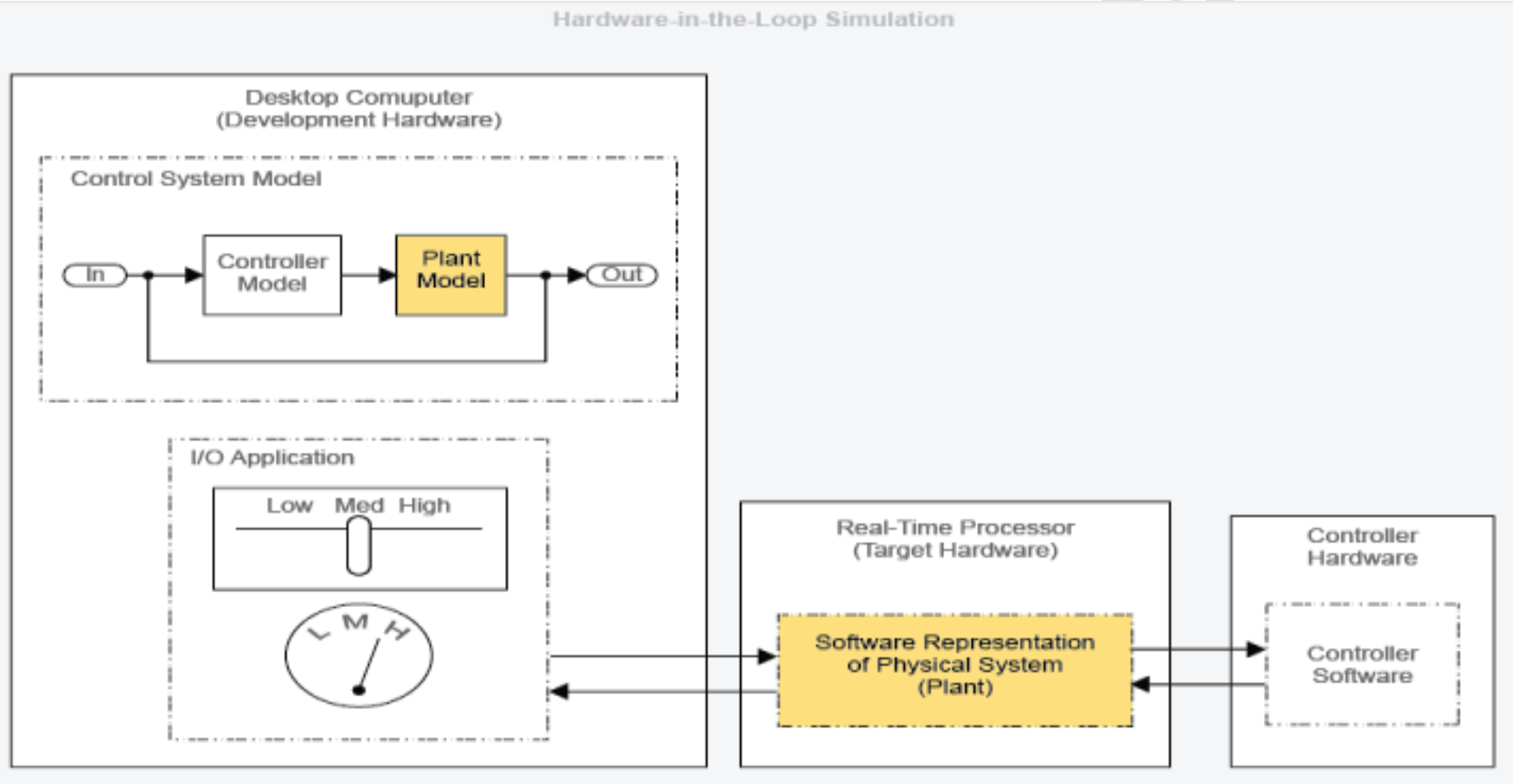
ECU <-> Plant Simulator / Model



ECU <-> Plant Simulator / Model

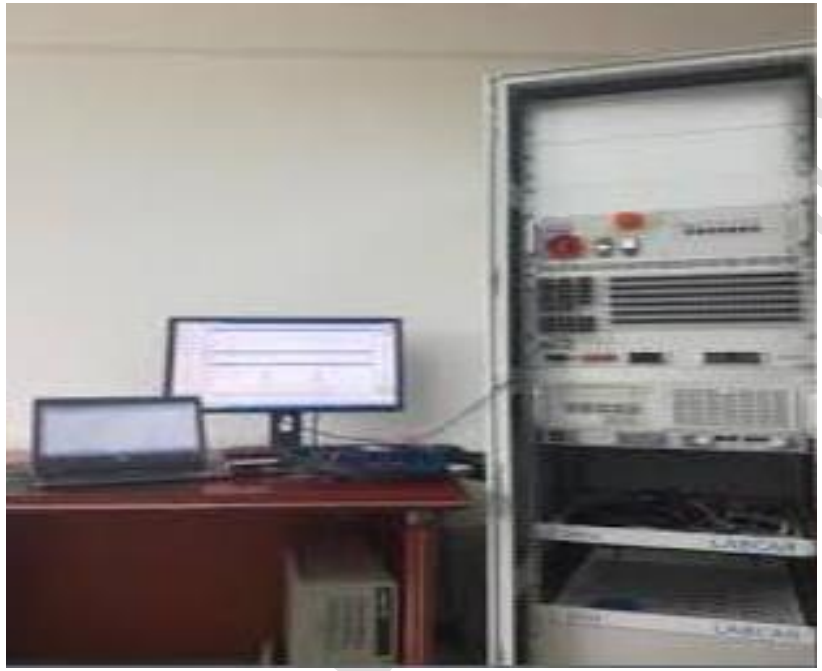

Model-Based Design
Design
Realization
Requirements Definition
Desktop Modeling and Simulation
Rapid Control Prototyping
Code Generation
Software-in-the-Loop Simulation
Processor-in-the-Loop Simulation
Hardware-in-the-Loop Simulation
Validation

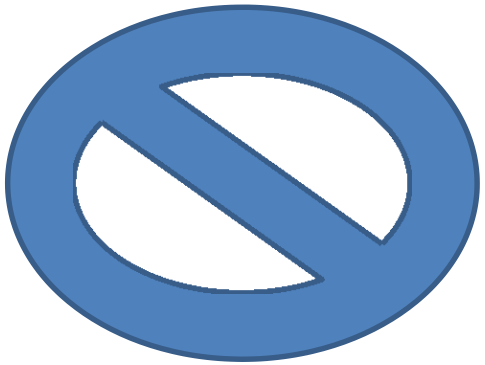
HIL Simulation



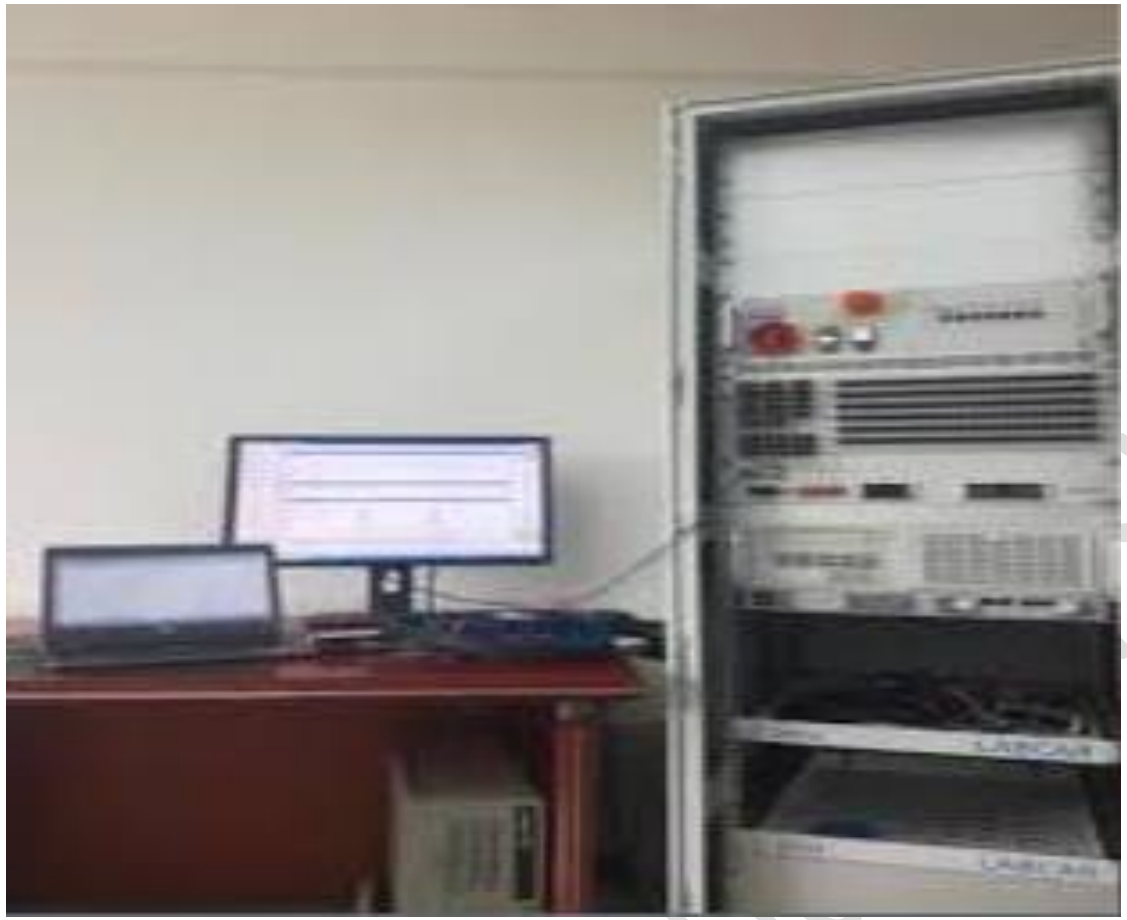
What is Plant Model

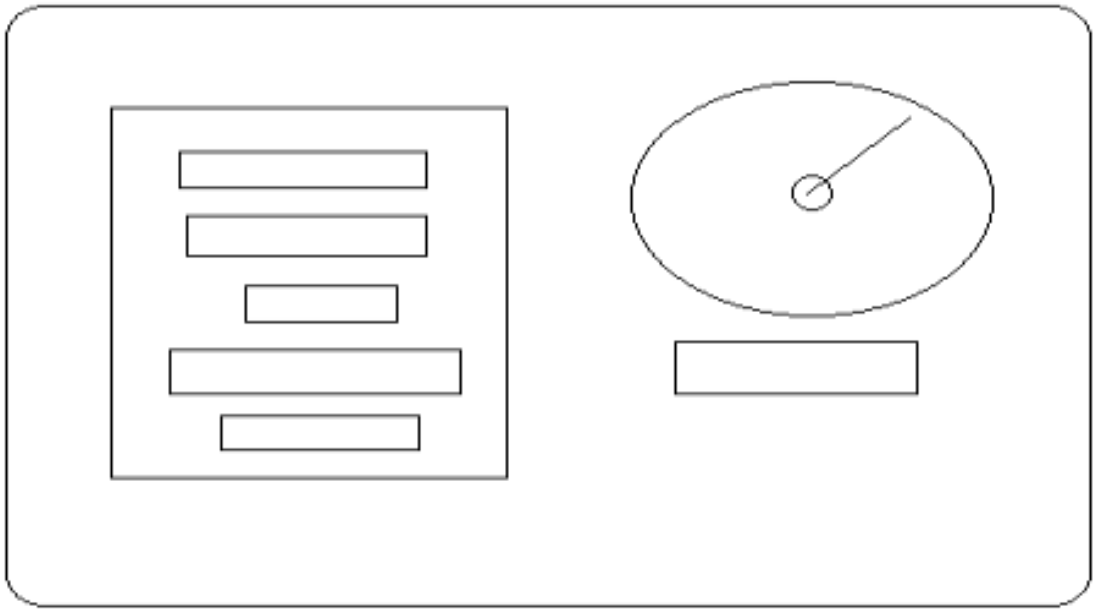
- Plant Model is a Replication for HIL Plant Simulator
- The Reason behind Plant Model is ?





What is Plant Model





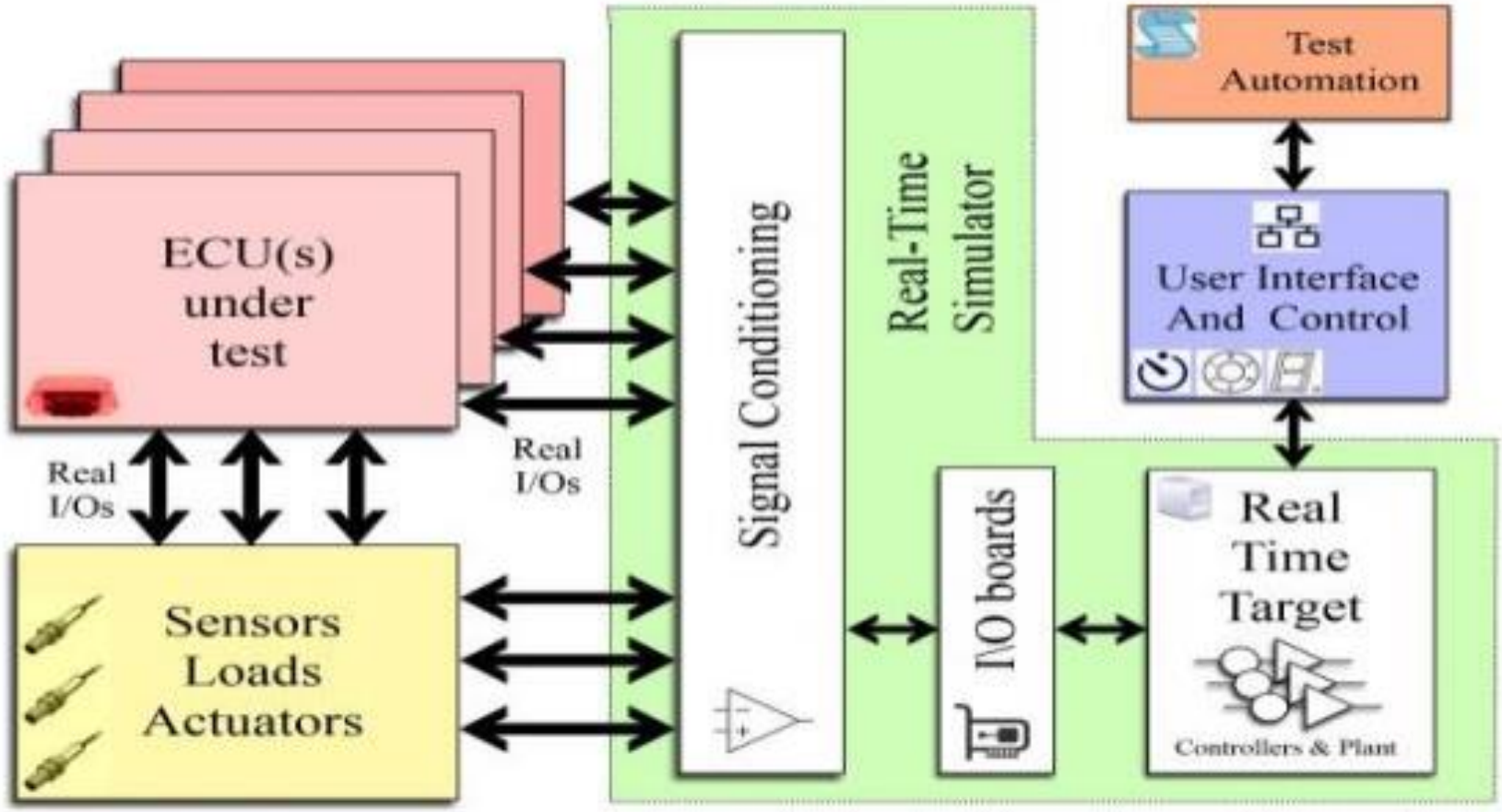
Who creates Plant Model ? How

- Plant Model can be created by anyone who knows to handle the tools and **configure I/Os**
- **What to be done when creating PM**
  - When creating plant model, Engineer should configure the I/O Cards, Harness Pins, Calibration Values, Switches & Relays, Sensors & Actuators under the boundary of certain tool which is used to generate model.

Plant Model :::: Replica for Plant Simulator

Plant Simulator

Generic Components



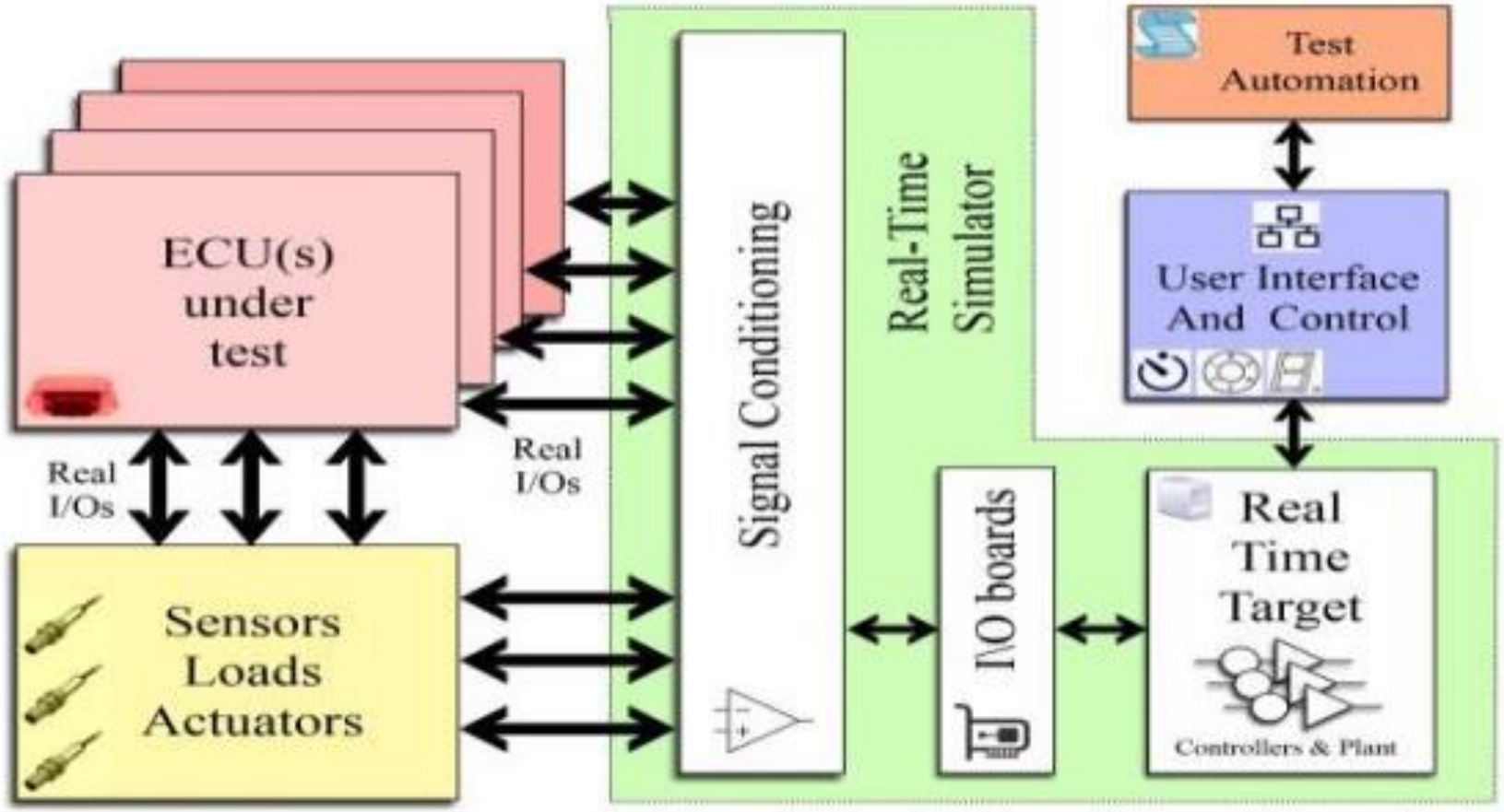
Plant Simulation

- Hardware-in-the-loop (HIL) simulation is a type of **real-time simulation**.
- We can use HIL simulation to test the **controller design & functionality**.
- HIL simulation shows how the controller responds in real time to **realistic virtual stimulation**.
- In HIL simulation, using a real-time computer as a virtual representation of the **plant model & real action** of the controller.

Plant Simulation

- Desktop computer Device Under Test (DUT) contains the **real-time model** of the controller & the plant.
- Flashed controller contains an **interface** with which to control the **virtual input** to the plant.
- Flashed Controller contains the controller software that is **developed** for the controller model and requirement.
- Real-time processor **(target hardware)** contains code for the physical system that is generated from the plant model (Matlab simulated & generated code)

I/O Interface Ports



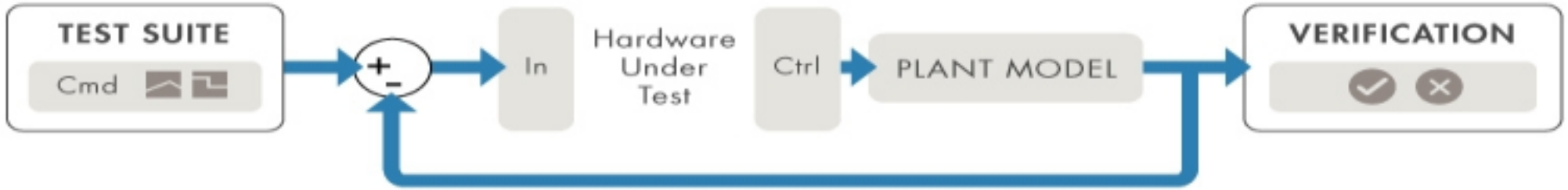
More about Plant !

- Testing involves using actual plant hardware or Emulator ECU or Hydraulic ECU to test the controller in **real-life situations** or in HIL environment.
- Plant simulation rely on a **naturalistic & environmental** test setup by allowing the plant model to represent the plant this signifies the HIL simulation offers benefits in cost and practicality.
- Plant Model **HIL Setup ECU** is more likely to approve the changes in embedded software.
- The Main Advantage of **HIL simulation is less expensive** and more practical than validation because the controller can set it up to run on its own with respect to the real conditions.
- HIL simulation is **more practical than validation** for testing the controller's response to unusual events also.
  - Example : The model can validate in extreme weather conditions or road gradients like **earthquakes, Up-slope, Down-slopes** etc.

More about Plant !!

- **Hardware-in-the-loop (HIL)** simulation is a technique for validating your control algorithm, running on **an intended or unintended target controller**, by creating a **virtual real-time** environment that replicates the physical system to control.
- HIL helps to test the behavior of your **control algorithms** without real hardware or physical prototypes.
- Virtual real-time implementation of physical real-time components such as **plant, sensors & actuators** on a real-time target computer.
- The software representations of HIL components are **gradually replace parts of the system** environment with the actual hardware components when the software starts to stabilized.

Loop of Plant/Plant Model



Hardware-in-the-loop (HIL) simulation setup. The block diagram shows a HIL simulation in which the hardware under test is an embedded controller and the plant model is a representation of a physical system.

Where is HIL simulation used ?

HIL simulation is especially useful when testing the control algorithm on the real **physical system is costly or dangerous.**

HIL simulation is widely used in the **automotive, aerospace and defense, and industrial automation and machinery** industries to test embedded designs.

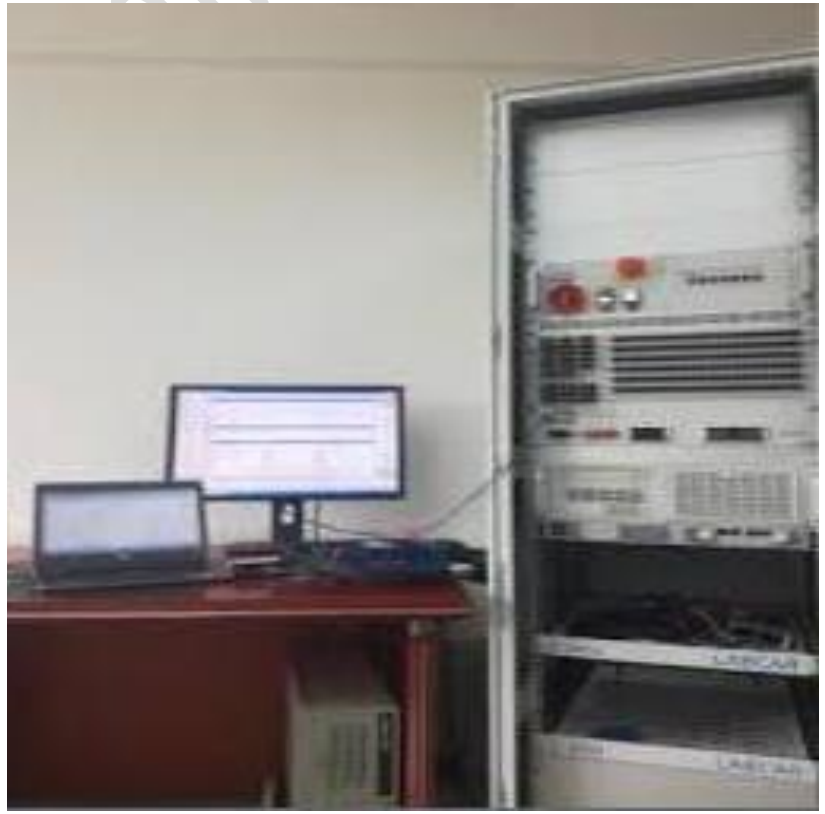
HIL is also being adopted in **medical devices, communications, semiconductors, and other industries**.

Application of HIL

HIL simulation Applies -

- **Aerospace and defense:** Flight simulators and flight dynamic control, where it is **too complex** to test the control algorithm on the actual aircraft
- **Automotive:** Vehicle dynamics and controls, where it is **impractical to test** the functionality on the road in the initial phases
- **Industrial automation:** Controller-plant testing, when stopping the **production or assembly** line to test control algorithms involves a huge amount of resources and business loss

Laboratory Setup



Laboratory Setup

HIL Testing

Next Part – 7

Process of HIL Workflow - 1

HIL Testing

Part – 7

Process of HIL Workflow - 1

DSpace

- Setting Up HIL Environment
- Flashing & Fault Free System
- Configuration Desk
- Control Desk
- Automation Desk
- Animation Desk

ETAS

- Setting Up HIL Environment
- Flashing & Fault Free System
- Configuration
- Manual Execution
- Automation

Car Maker

- Setting Up HIL Environment
- Flashing & Fault Free System
- Simulation (Manual / Automation)
  - Driver
  - Road
  - Vehicle
  - Traffic

Setting Up HIL Environment

Fault Free System

System with Fault

- In HIL or SIL Environment whenever the software flashes in to **ECU or VCU** then System goes to some fault irrespective of any inputs or by default.
- Tester/Developer responsibility to **debug and resolve** the issue

Common Faults to be clear before start execution

✓ **CAN Channels** to be configured properly in Plant simulator

✓ **I/O Cards** must be configured properly

✓ Calibration variables to be **ensured/overwritten** in plant model

Cockpit

Cockpit


Experiment
Edit
View
Measurement
Hardware
Workspace
Custom Cockpit [17]
ETAS
Google Maps Viewer [7]
Measurement stopped

Cockpit


ETAS
Pedals Graph
Engine Torque
Battery
Brake
Accelerator

Configuration Desk

- Configuration Desk supports you with **pre-compiling containers for Simulink models** and **Functional Mock-up Units.**
- This means **models can pass without the source files**, thus **protecting your intellectual property and simplifying model exchange**.
- In addition, **build time can be saved**, because once a **precompiled container** is generated, it can be **reused in different projects or variants**, without having to generate and compile the C code again.

Configuration Desk

- With Configuration Desk can be used to easily implement **the behavior model code** and the I/O function code from Configuration Desk on the dSPACE real-time hardware.
- The **entire build process** for a real-time application is handled by **Configuration Desk**.
- Being able to interactively work with MATLAB Simulink models, **previously generated C code** can be imported from **different modeling tools**, such as code from the Simulink Coder via Simulink Implementation Containers (SICs) or code from other modeling tools via Functional Mock-up Units (FMUs).

Configuration Desk

- Comprehensive documentation options and graphical displays gives a **great project transparency.**
- **Assembling and configuring** the project-specific hardware offline as a **virtual system.**
- A real-time application can be executed for test runs even if parts of the necessary and configured I/O hardware are not physically available.
- In addition, there are options to generate a **Microsoft® Excel® file** with information on the wiring harness and on external devices.

Configuration Desk

- The new Configuration Desk navigation bar **offers task-specific view sets that are optimized for the chosen workflow.**
- Each view displays only the functions required for the respective task.
- Using **model containers reduces external dependencies**, because **no model tool installation is required** to use the models.
- It also simplifies the **exchange and re-use of models**, because it **saves code generation and build time.**

Configuration Desk

- SICs are **ZIP containers includes the C code and other artifacts,** such as precompiled libraries and a model interface description.
- Configuration Desk provides **a method to convert SIC files** with source files into SIC files without readable source files but **with a SCALEXIO-compatible library file**,
- This might **be desirable for protecting your intellectual property.**
- Once the **SICs are generated, they can be reused in different projects,** without having to generate the C code or library again, thus saving build time.

Configuration Desk

- FMUs enables to **use different modeling approaches**, for example, based on physical modeling with **Modelica**.
- In Configuration Desk, **FMUs can be integrated together with Simulink® models**.
- **Pre-complied FMU files can be used**, thus saving build time and protecting your intellectual property.
- The **user workflow for importing and connecting FMUs to other model interfaces and to I/O is identical to the workflow for SICs.**
- The Configuration Desk **version 6.0** offers optimized view sets for **two different working methods.**

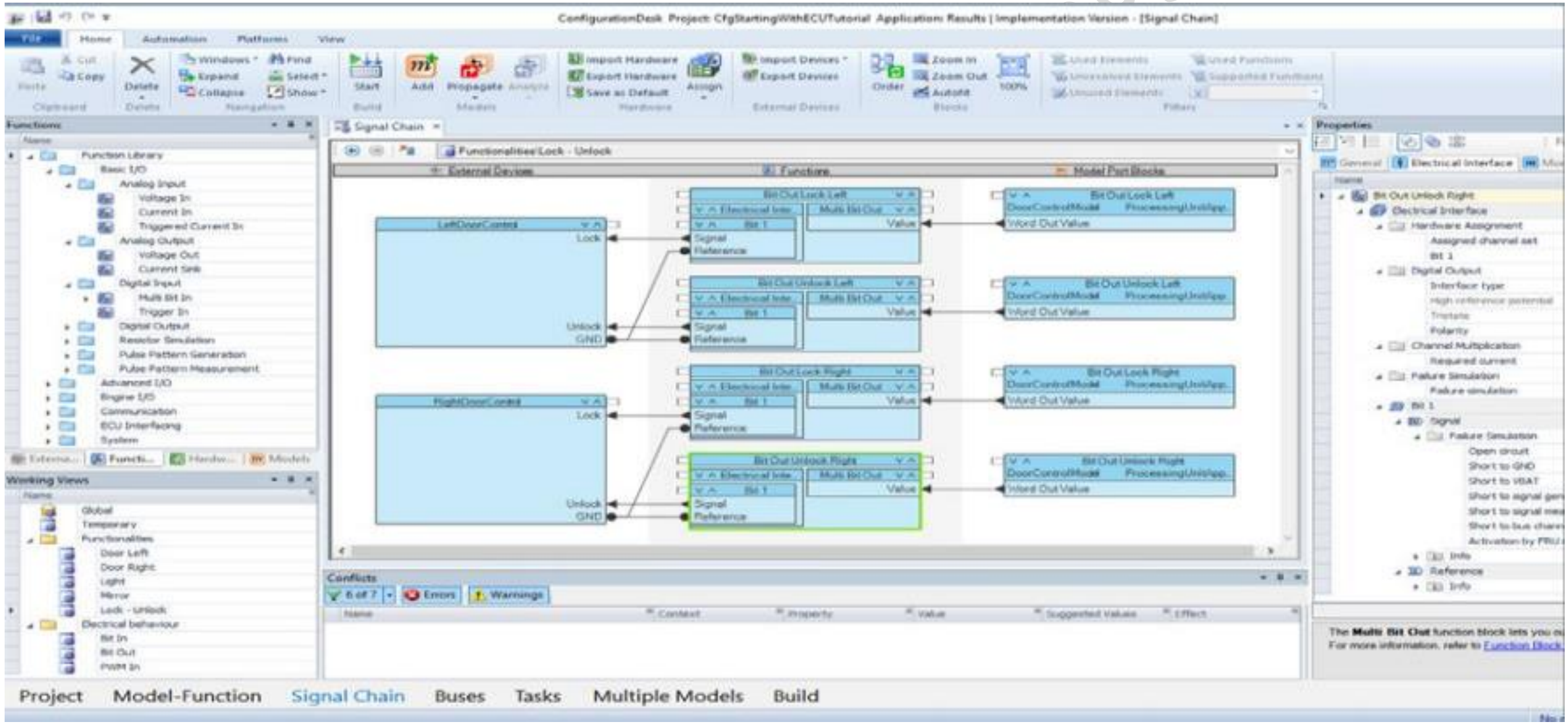
Configuration Desk

- The **Model-Function page** is optimized **for Simulink-oriented work with projects** for which the **Simulink model interface still has to be adapted or changed.**
- The **Signal Chain page** is optimized **for container-oriented work with projects** for which **the model interfaces are already fixed.**
- Convenient Model Exchange.
- Making exchanging simulation models easy, **dSPACE** offers a **Model Interface Package for Simulink®** (MIPS) for generating Simulink implementation container (SIC) files.

Configuration Desk

- With the **free-of-charge** MIPS, **modeling experts can generate the (C code) SIC file with Simulink Coder, without needing a VEOS or ConfigurationDesk license.**
- Out of their Simulink models and together **with dSPACE Run-Time Target, they can generate code and create ZIP files** that contain all the necessary code and artifacts **for executing the models on different simulation platforms, such as VEOS and SCALEXIO.**
- **Model integrators using SIC files do not have to generate code again for building the simulation.**
- Using SICs therefore **significantly reduces the amount of time needed for reusing** the SICs in different projects.

Configuring Panels / Variables



Control Desk

Control Desk

- Control Desk is a kind of Platform in Dspace to control the **configuration variables and other parameters** manually without any automation
- Control Desks helps to find the issues or errors in the development or testing phase with **smoke level configuration**
- As it is the part of Dspace the significant usage of **HIL plays crucial role** by control desk.
- These kind of manual execution is common in all the HIL / SIL platforms to ensure the **software is testable**

Control Desk

HIL Testing

Next Part – 8

Process of HIL Workflow - 2

HIL Testing

Part – 8

Process of HIL Workflow - 2

Automation Desk

- With Automation Desk, tests can be executed **24 hours a day, seven days a week**, letting engineers increase **test coverage & improve ECU software** quality while saving time and costs.
- Graphical description of test
- **Advanced custom library**
- Remote control of calibration, measurement, and diagnostic tools such as Control Desk Next Generation.

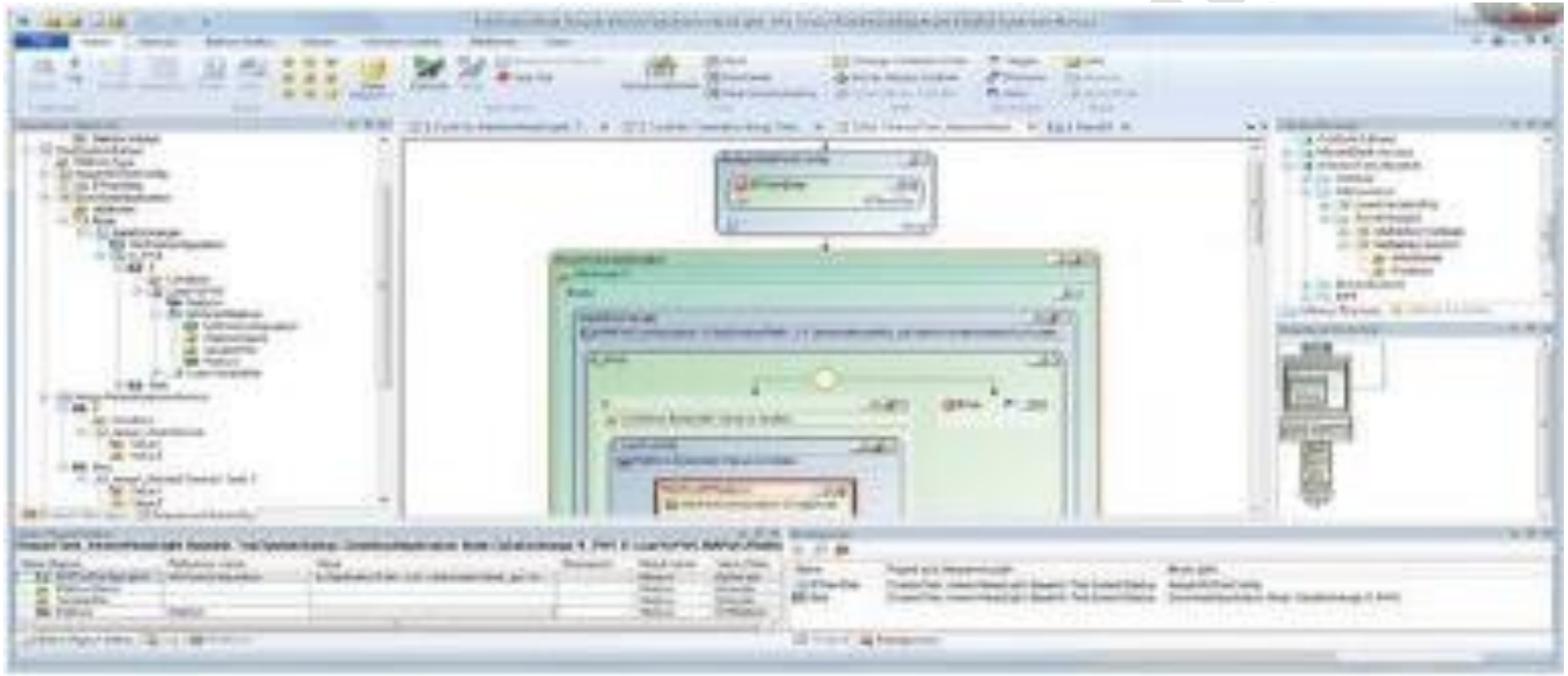
Automation Desk

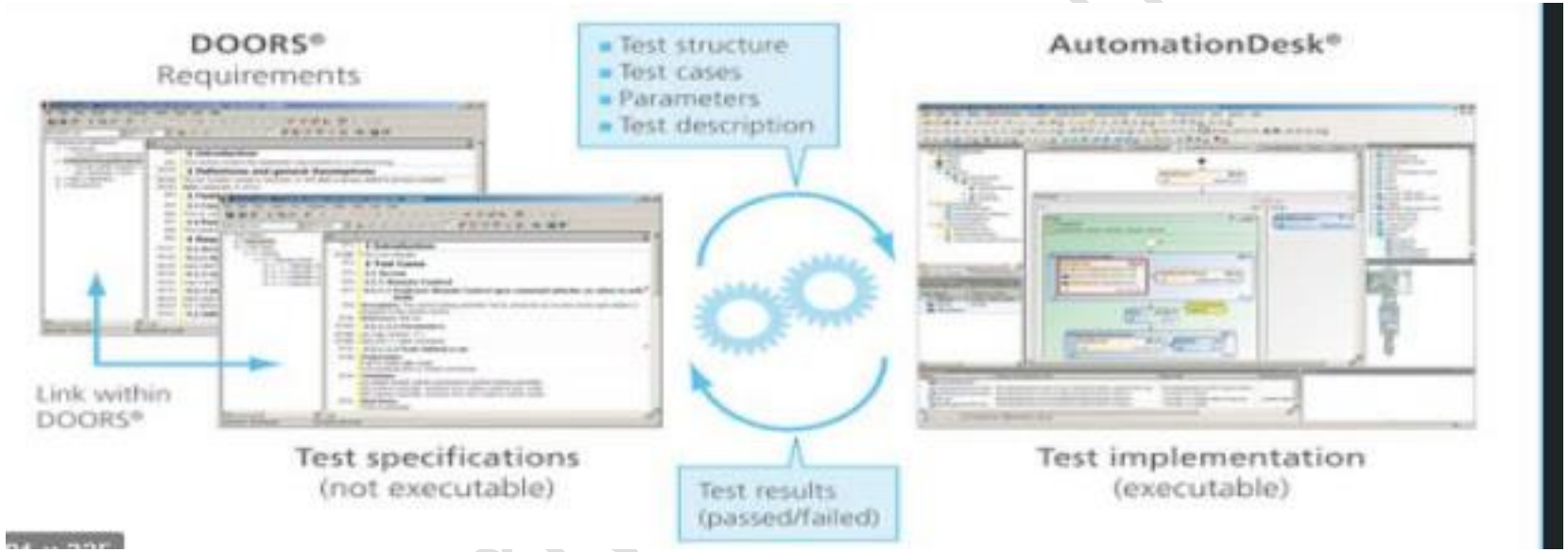
- Automation Desk **provides libraries containing a large number of predefined test steps** (e.g., for easy access to the HIL simulator, to a Failure Insertion Unit (FIU), or to calibration or diagnostics software).
- With Automation Desk, **tests can be executed 24x7**, letting engineers increase test coverage **and improve ECU software quality while saving time and costs.**
- Automation **Desk lets you describe test routines graphically without expert knowledge in programming.**

Automation Desk

- Automated Hardware-in-the-Loop (HIL) Testing
- New-Look of User Interface
- NEW: Advanced Sequence Builder
- NEW: Offline Test Execution & Development
- NEW: Enhanced Multi-User Support

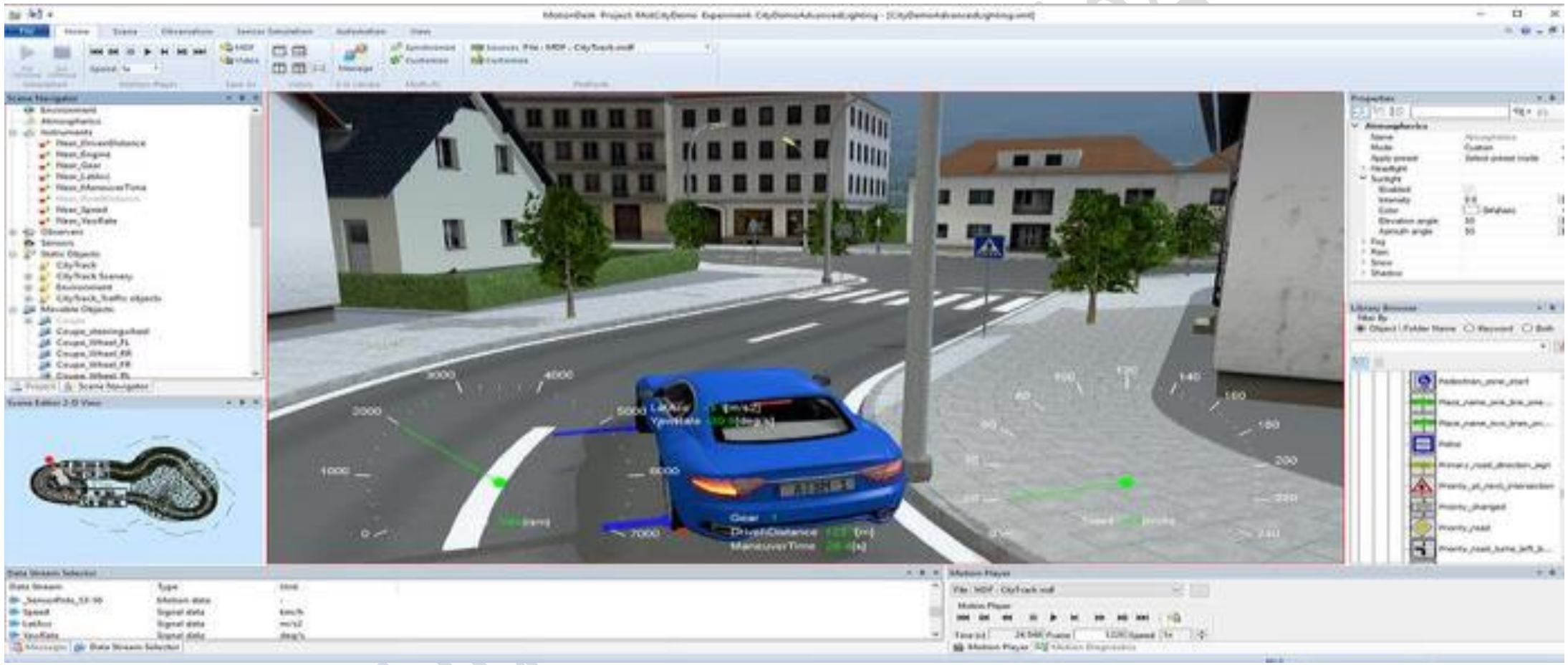
Automation Desk

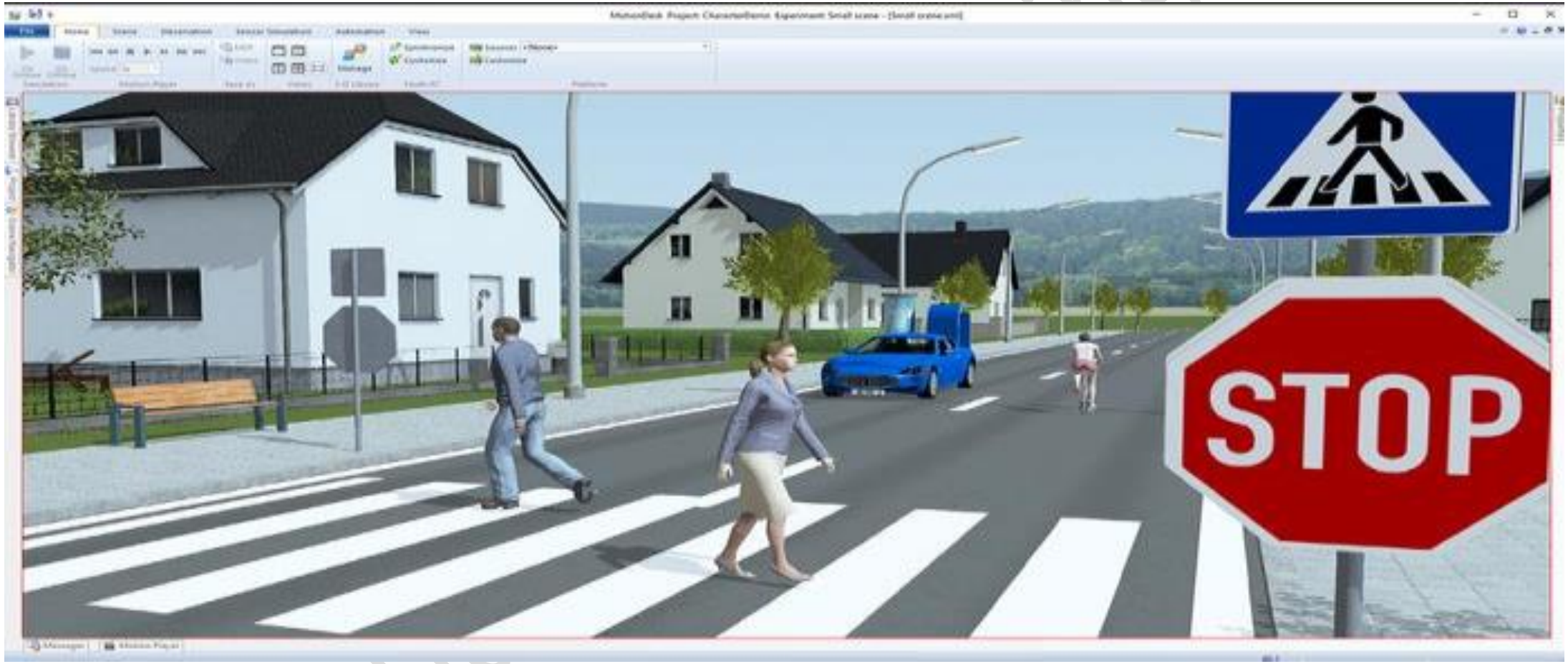



DOORS®
Requirements
Test structure
Test cases
Parameters
Test description
AutomationDesk®
Link within DOORS®
Test specifications
(not executable)
Test results
(passed/failed)
Test implementation
(executable)

Motion Desk

- dSPACE Motion Desk is a **3-D online animation software application** that **visualizes the results of hardware-in-the-loop (HIL) simulations for ECU tests.**
- **Motion Desk supports** hardware simulation platforms from dSPACE, such as **SCALEXIO and DS1006**.
- The tool also supports **offline simulation** based on dSPACE VEOS and Simulink® and comes with **valuable features for developing and testing ADAS scenarios.**
- These include **camera sensor simulation** during which Motion Desk **video data is fed directly to the camera ECU and ideal point-cloud sensor models for the development of radar and lidar sensors.**




STOP

ETAS

- Configuring the Software & Related files in the tools.
- **Manual Testing can be Executed**.
- **Automation scripts can be generated and executed by framework.**
- **Virtual Simulation can be done.**

CAR Maker

- Configuring the Software & Related files in the tools.
- **Manual Testing can be Executed**.
- **Automation scripts can be generated and executed by framework.**
- **Virtual Simulation can be done.**

HIL Testing

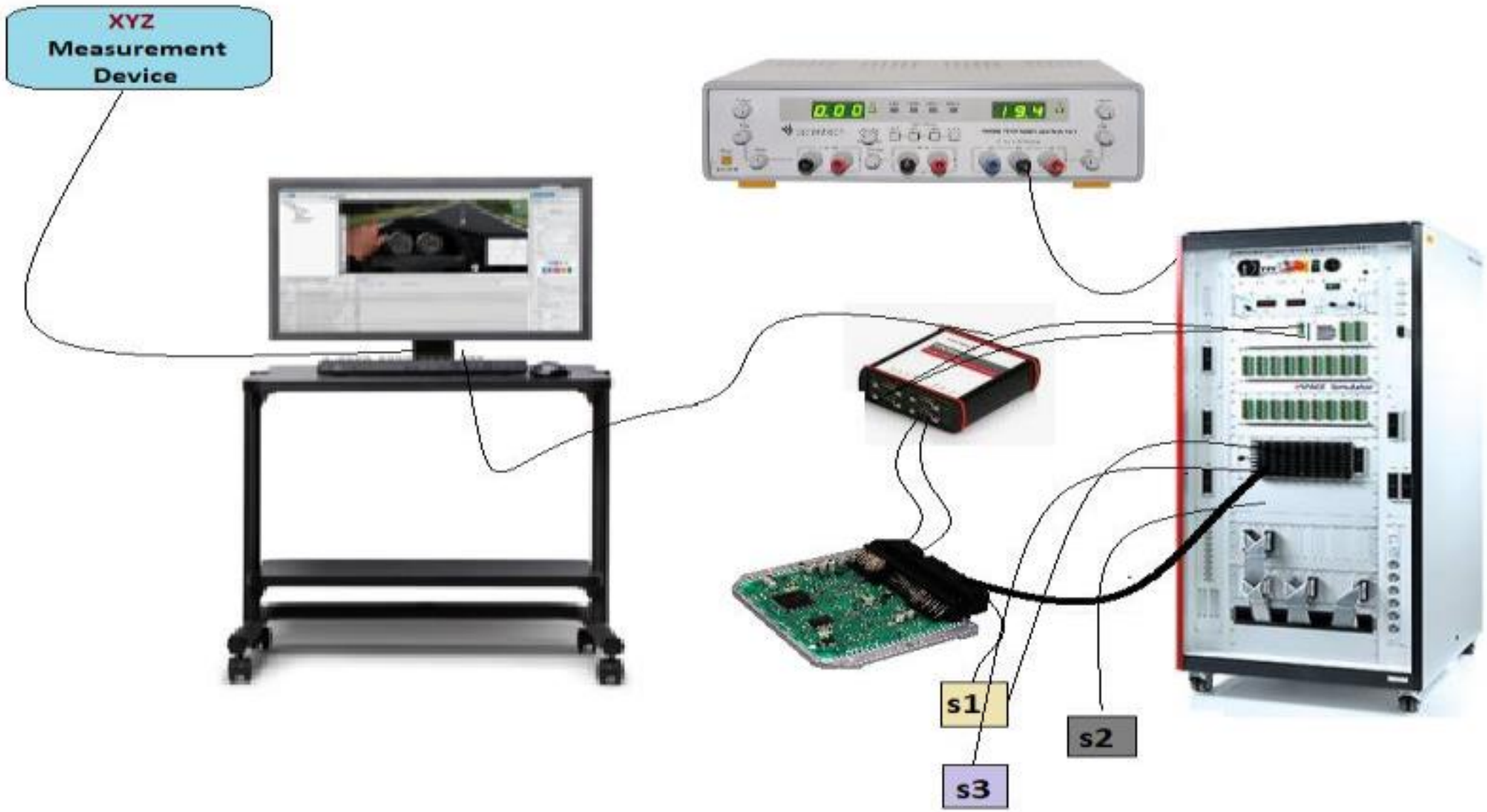
Next Part – 9

Clear cut explanation on Components in Plant Simulator

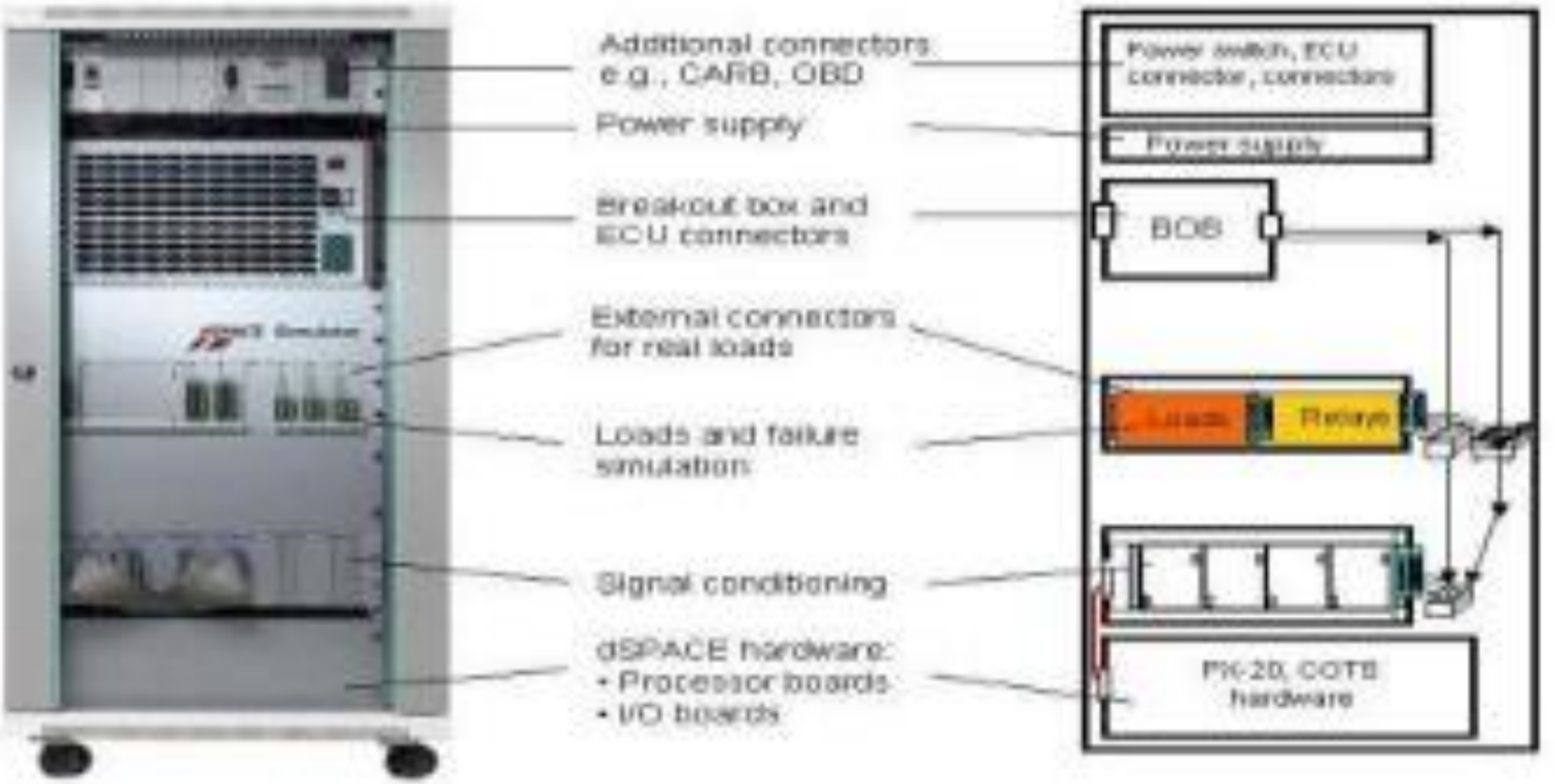
HIL Testing

Part - 9

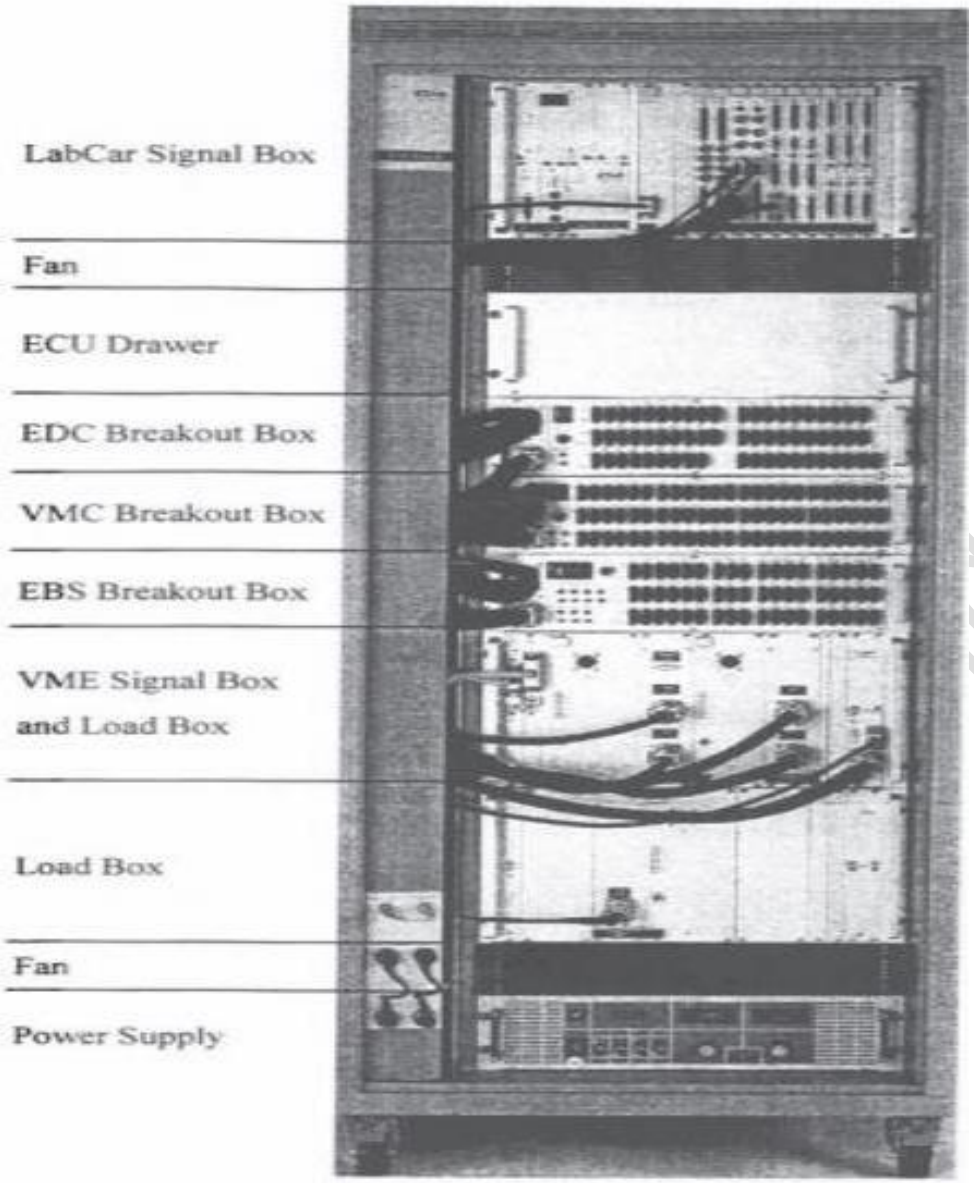
Components in Plant Simulator


XYZ
Measurement
Device
s1
s2
s3

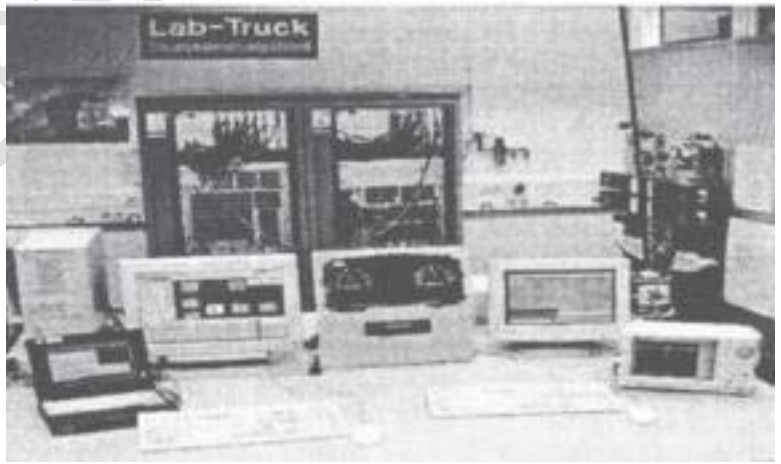
Plant Simulator



Plant Simulator



Lab Setup of HIL



Introduction about Plant Simulator

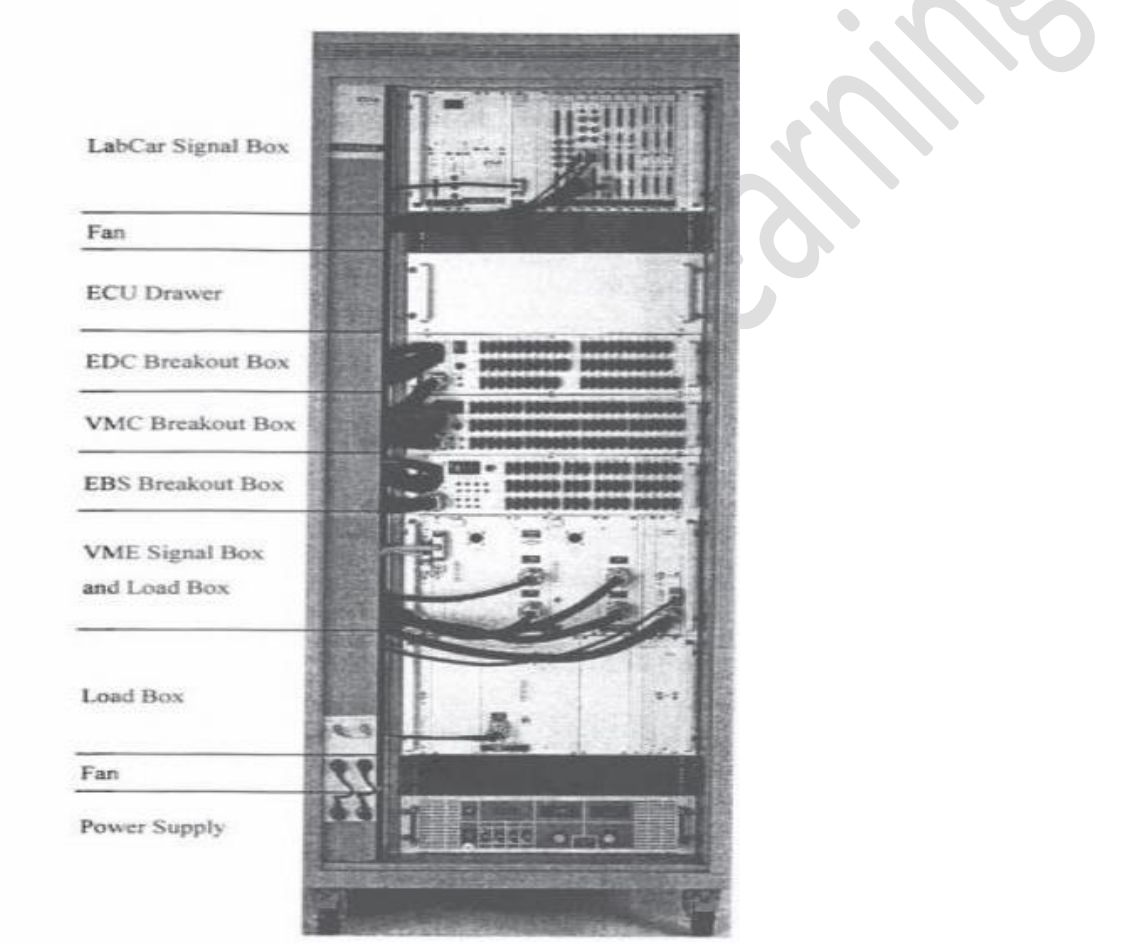
- Plant Simulator are used for laboratory tests of automotive control systems (ECUs) as per test conditions.
- Plant Simulator stimulates the ECU **via sensor inputs and network ports**.
- This allows users to observe and to measure the ECUs response in **absence of physical vehicle.**
- In added with that Plant Simulator simulates **a real vehicle**, making the ECU under test believe that it is in its real operating environment.
- A typical Plant Simulator installation consists of four components:
  - Software to control the experiments,
  - Hardware to stimulate and Measure the ECU inputs and outputs,
  - Models which is sub for Plant Simulator
  - Software to create and automatically execute test scenarios.
- Developers can test the ECU in **all realistic operating** and fault conditions with perfect repeatability and in improbable and dangerous conditions before live driving situations.
- For a lot of tests, **Plant Simulator can replace physical** prototypes of engines, transmissions, Airbags and so on ..

ECU Drawer

- In the standard configuration, the ECU drawer is located between the **signal box and the breakout** box. It can safely hold the ECU and its associated wiring.

- Purpose of **ECU Drawer** is to hold the ECU and can safely remove whenever its required to do so

Break out Box



Break out Box

- Breakout box is a breakout device that makes possible to measure, modify, or manipulate signals on the **Plant Simulator however required.**
- This hardware interface for **ECUs splits up signals** and converts them as per the inputs provided.
- **100+ identical channels** (own ground connection (GND) per channel) provided in the breakout box makes plant to safer and reliable
- Suitable for voltage up to 60 V DC
- Suitable for current up to 10 A per channel (total current < 400 A)
- 4-millimeter jumpers

Note : Without proper information plug any connections like Sensors, Ground, Batt into the breakout box with / without jumper

External Break out Box

- External breakout **doesn't have any specific feature** rather it can isolate from the Plant simulator without disturbing functionality or connections
- The significance of breakout is to **avoid harm** to the plant simulator by doing physical interactions with plant simulator replica i.e external breakout box

Harness Cables

Harness Cables

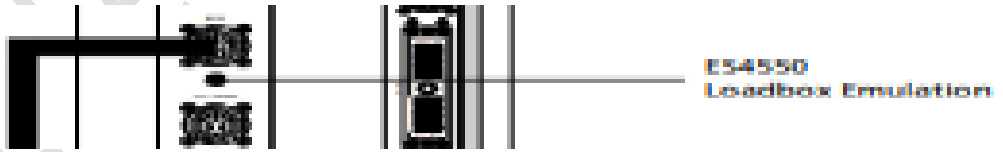
- **Wiring harness or Harness Cables** is an assembly of electrical cables or wires which **transmit signals or electrical power**.
- The cables are bound together by a durable material such as rubber, vinyl, electrical tape, **a weave of extruded string**, or a combination thereof
- Automobiles and spacecraft contain many masses of wires which would stretch over **several kilometers** if fully extended.
- By binding the many wires and cables into a cable harness, the wires and cables can be better secured against the **adverse effects of vibrations, abrasions, and moisture**.
- Binding the wires into a **flame-retardant sleeve** also lowers the risk of electrical fires.

Harness Cables

- **Class 1:** General Electronic Products, for objects where the functionality of the final product is the major requirement. This can include objects such as **toys and other items** that do **not serve a critical purpose**.
- **Class 2:** Dedicated Service Electronic Products, where consistent and **extended performance is needed**, but uninterrupted service is **not vital**. The failure of this product would not result in **significant failures or danger**.
- **Class 3:** High Performance Electronic Products, for products that require continued and consistent performance and where periods of inoperativeness **cannot be tolerated**. Most of the **critical applications** uses this class harness.

Signal Box & Load Box

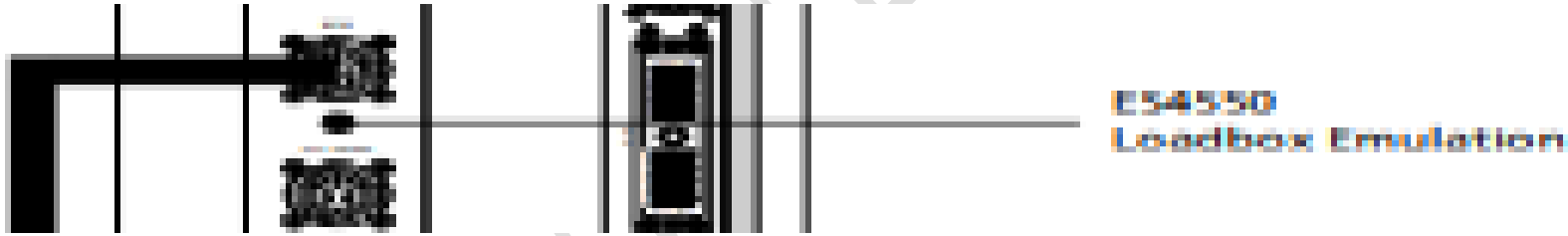
- Signaling control provides an interface between the **human signal operator and the signaling equipment**.



- The signal box can accommodate approximately 30 boards for generating and measuring ECU signals. The basic system can thus be easily extended for handling tests of ECUs that control **4-16 cylinder engines**. The optional failure simulation component produces faults for up to **100 channels**

Signal Box & Load Box

- Load box is used to simulate loads **(e.g. apply brake or inflate airbag).** This enables Plant simulator to simulate the respective module once the inputs has triggered. For example, the operating characteristics of actuators of Brake solenoid or airbag inflator is to be done the **Load box Emulation Board** makes it possible to simulate a electrical responses.



Signal Conditioning

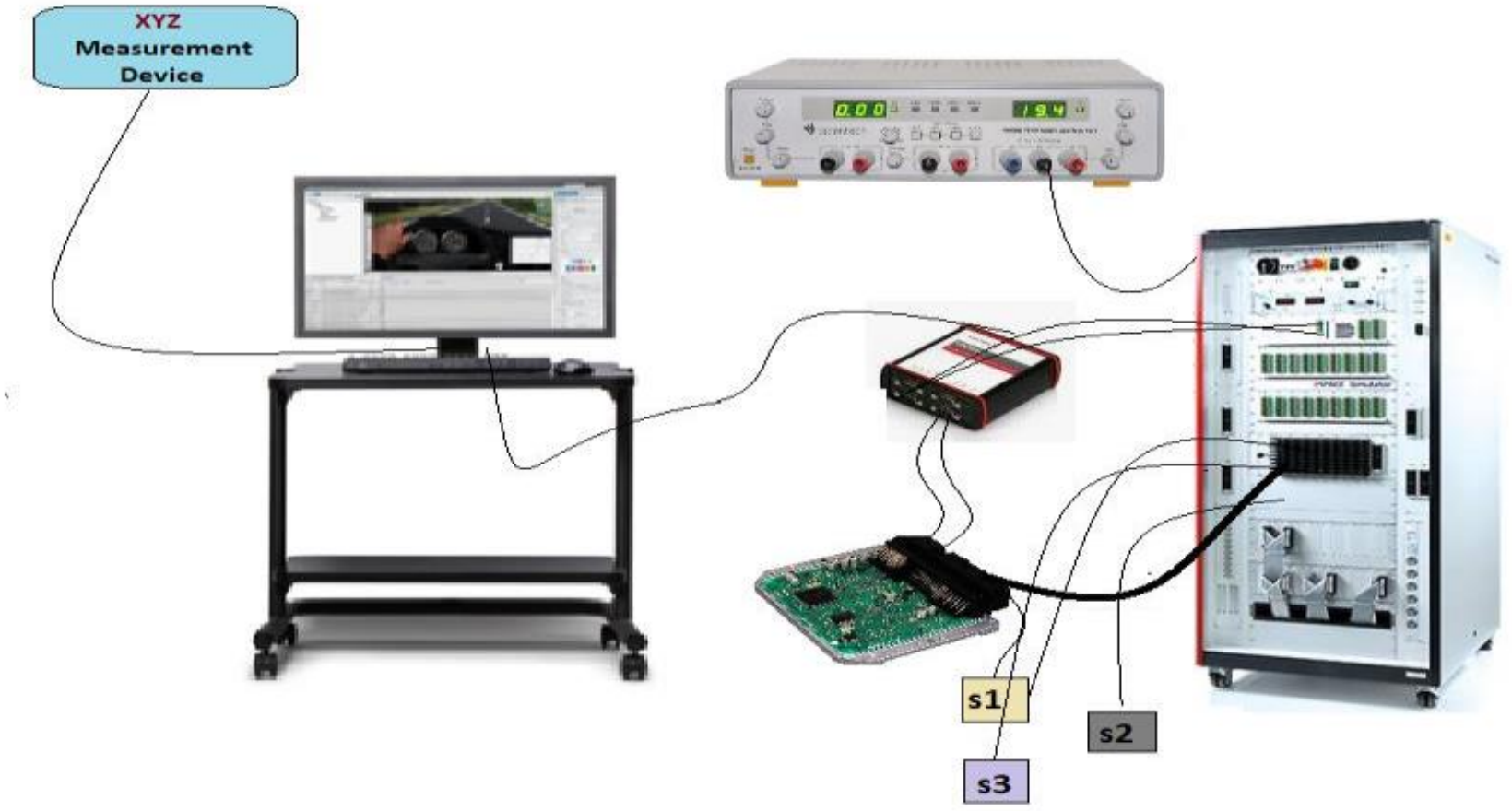
- **Signal conditioning** is the manipulation of a **signal** in a way that prepares it for the next stage of processing. Many applications involve environmental or structural measurement, such as temperature and vibration, from sensors

- In other words, Signal conditioning circuits condition the signal as per **user needs**

CAN Channels - Ports

CAN Channels - Ports

Sensor Cards



End of HIL Tutorial !!

Bonus !!!

1. What is HIL, Why HIL, Importance/Significance
2. ECU
3. Flashing into ECU
4. Workflow of HIL
5. Components of Plant Simulator
6. What is Plant Model
7. Different Suppliers / OEMs
8. Manual
9. Automation
10. VAFs in HIL Testing
11. ……

Please find the Downloads !!

» Control Desk

» Automation Desk

Input & Output in HIL