



## Automation & simulation techniques

---



**Author: Matteo Gallazzi**

**Software Test Engineer**





- Company profile

**vision**

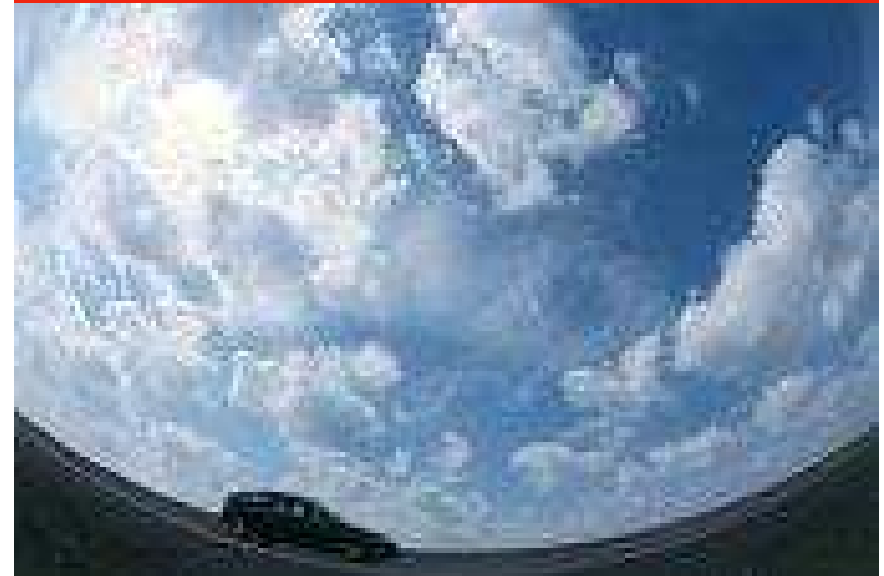
Our vision is a more secure world everywhere  
and at all times

**mission**

Cobra's mission is to provide worldwide, user friendly  
vehicle-centric solutions integrating advanced  
technologies that enhance **transportation and  
property security** and to work with public and private  
organisations that share our vision

# company profile '09

**Our aim is to make the world a safer place**





■ Cobra solutions

The antitheft



The manoeuvring aid



The Location Based Services





## Partnership with Vehicle Manufacturers

	<i>since</i>
Nissan	<b>1991</b>
Renault	<b>1993</b>
Volkswagen - Audi	<b>1994</b>
Honda Access Europe Motorcycle – Ducati Motor	<b>1996</b>
Toyota	<b>1999</b>
Renault Véhicules Industriels – Volvo Trucks – Porsche – Ford Daimler- Chrysler – Jaguar – Land Rover – Mazda – Skoda	<b>2001</b>
Honda – Ferrari – Maserati – Yamaha Motor Europe	<b>2002</b>
Mitsubishi – Bentley – Lamborghini	<b>2003</b>
Scania - MV Agusta Corse	<b>2004</b>
MV Agusta Motor	<b>2006</b>
Hyundai – Kia – Iran Khodro – Shanghai Volkswagen	<b>2008</b>
DPCA (Dongfeng Peugeot Citroen Automobile Company) SAIC (Shanghai Automotive Industry Corporation) BHMC (Beijing Hyundai Motor Company) – Geely	<b>2009</b>



Cobra aims at being recognised as a company constantly focused on understanding security needs. The goal is to give to client partners the solutions that contribute to reinforce their image of responsibility towards security.

## Partnership with Vehicle Manufacturers

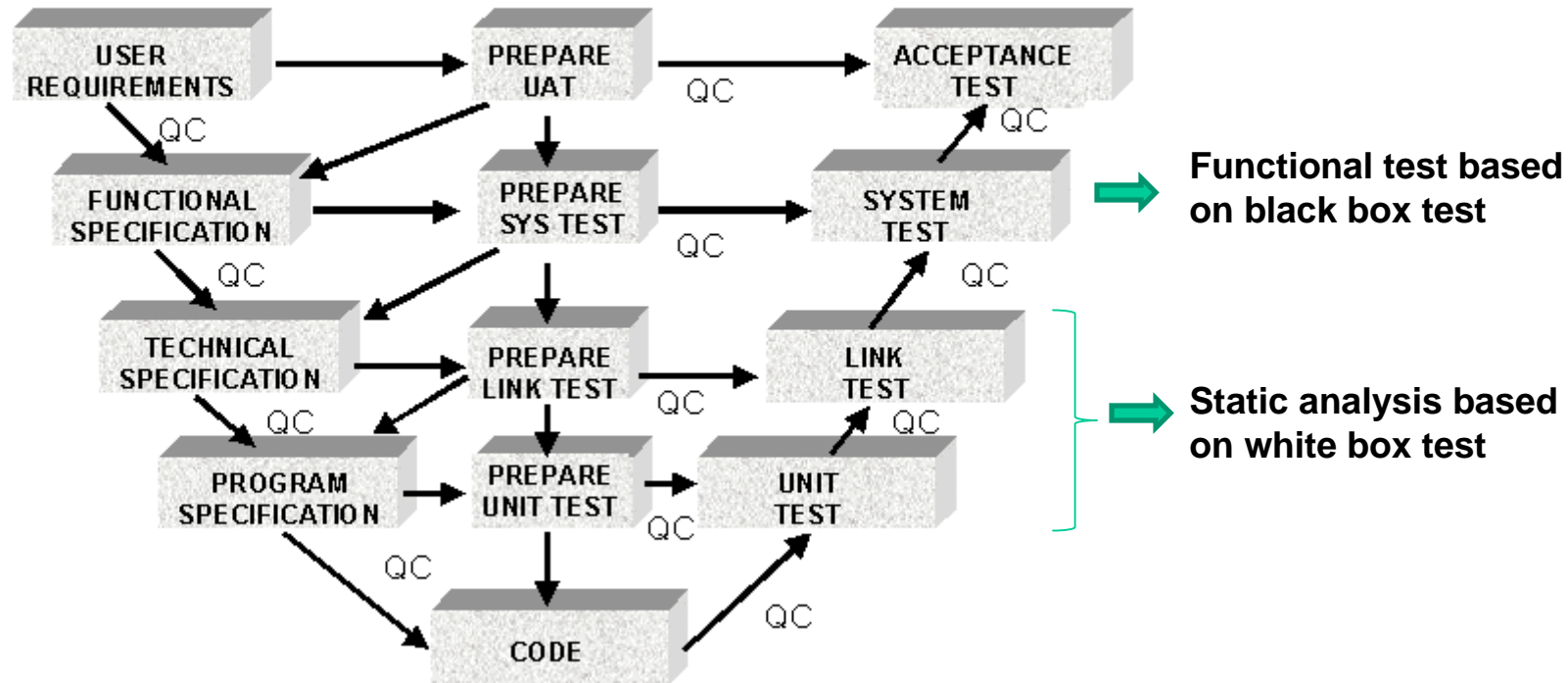




- Software life cycle

Cobra applies the V model in software process development

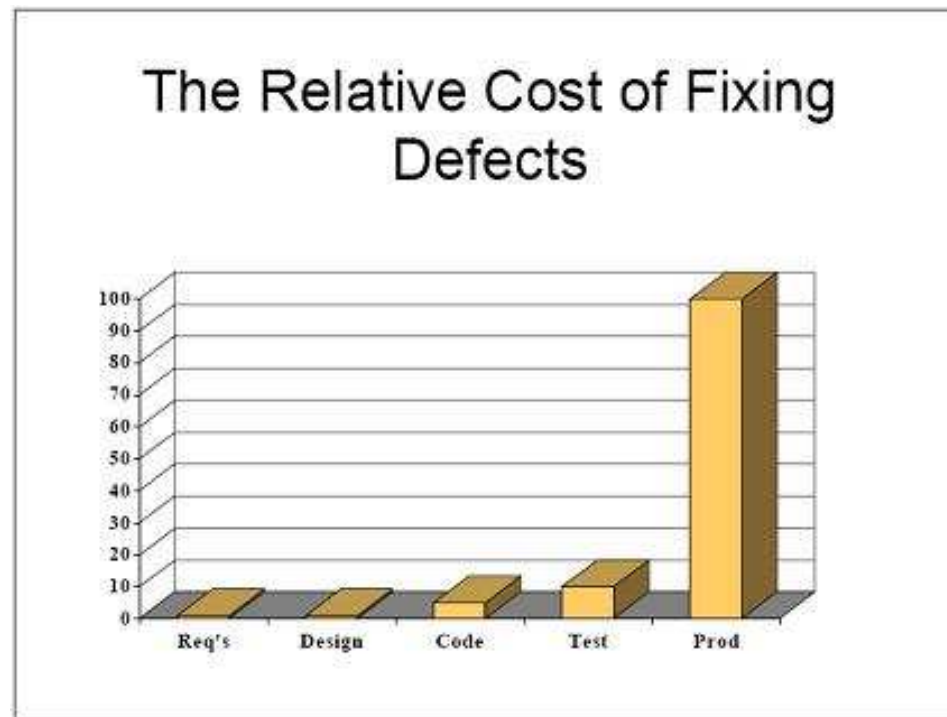
## THE "V" MODEL





- Bugs cost

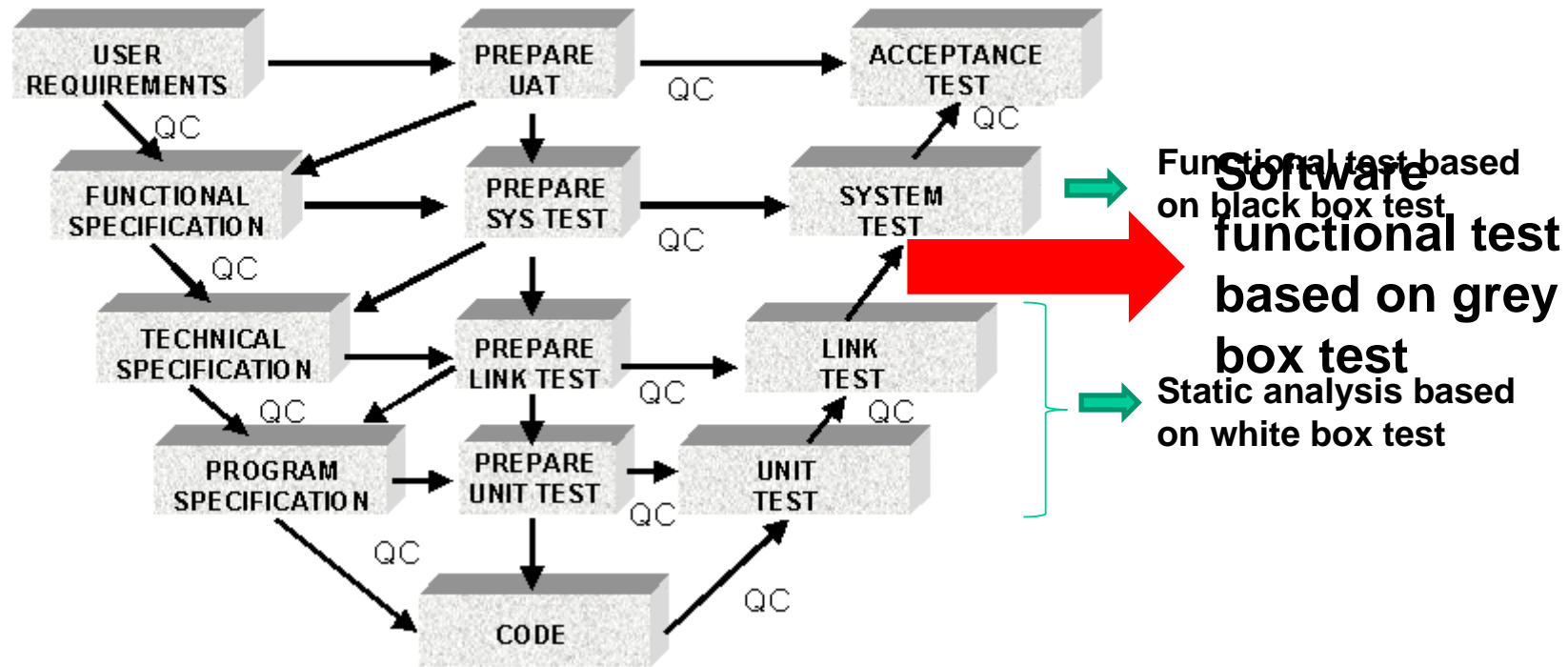
### Bugs costs vs. project phase





● COBRA's strategy

# THE "V" MODEL



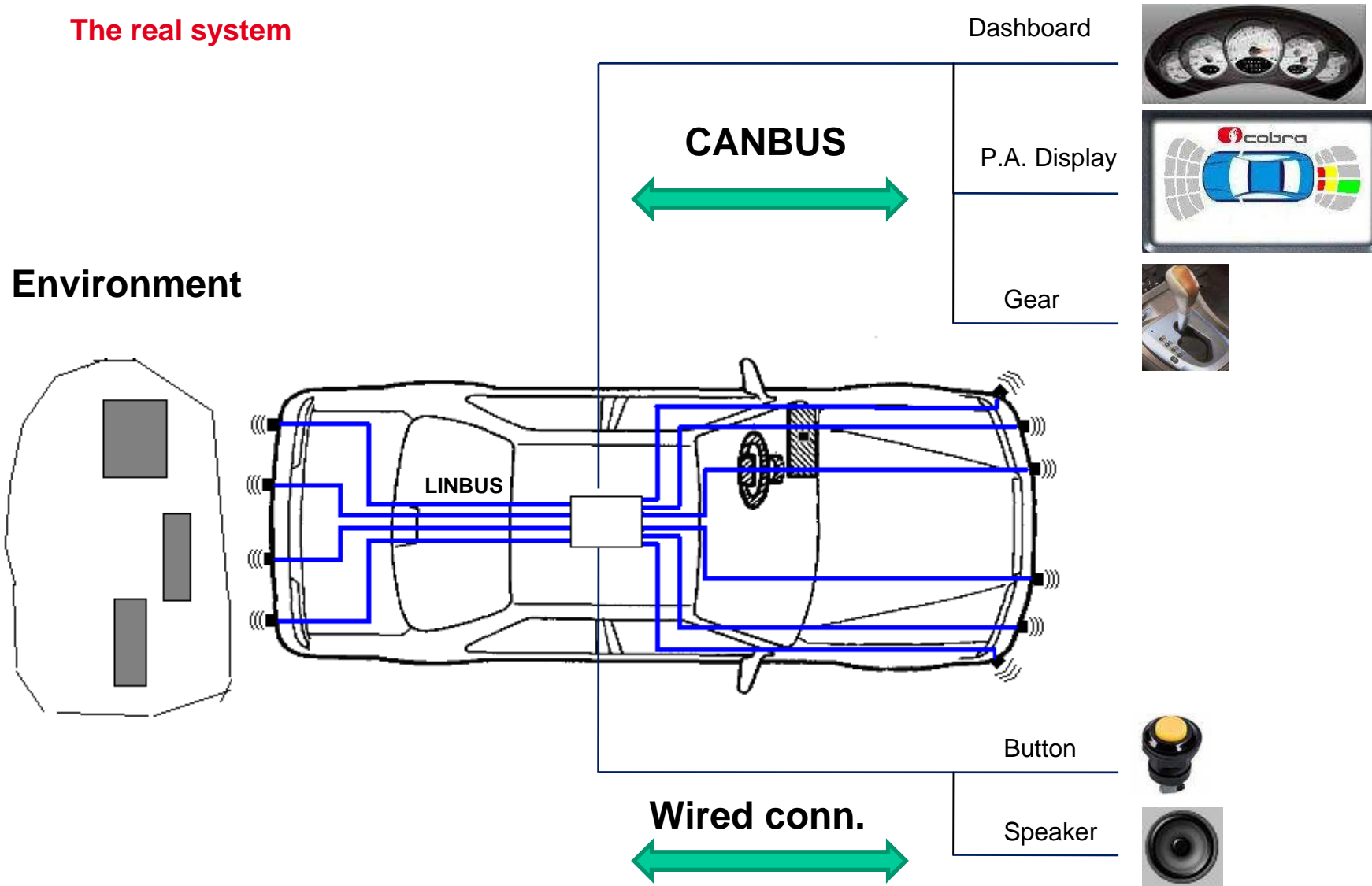
The aim is to test only the software functionalities on real target







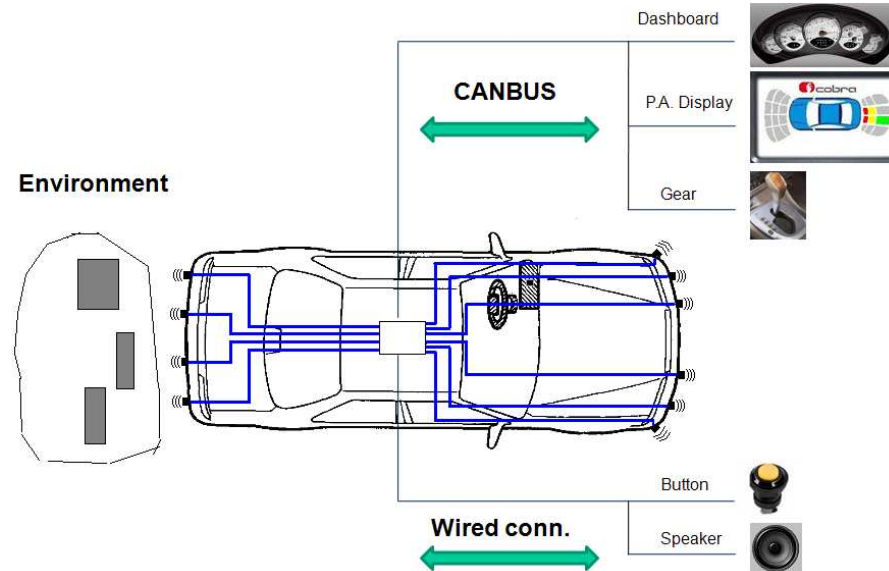
The real system







## ● The real system



January 2008

Isolate SW testing in real environment could be problematic because of the interaction with:

- Bus line
- Sensors
- Environment
- Electrical noise





## ● The challenge

Provide simulation tools that allow SW Engineers to debug SW Functionalities

Use the same simulation tools to execute the test cases

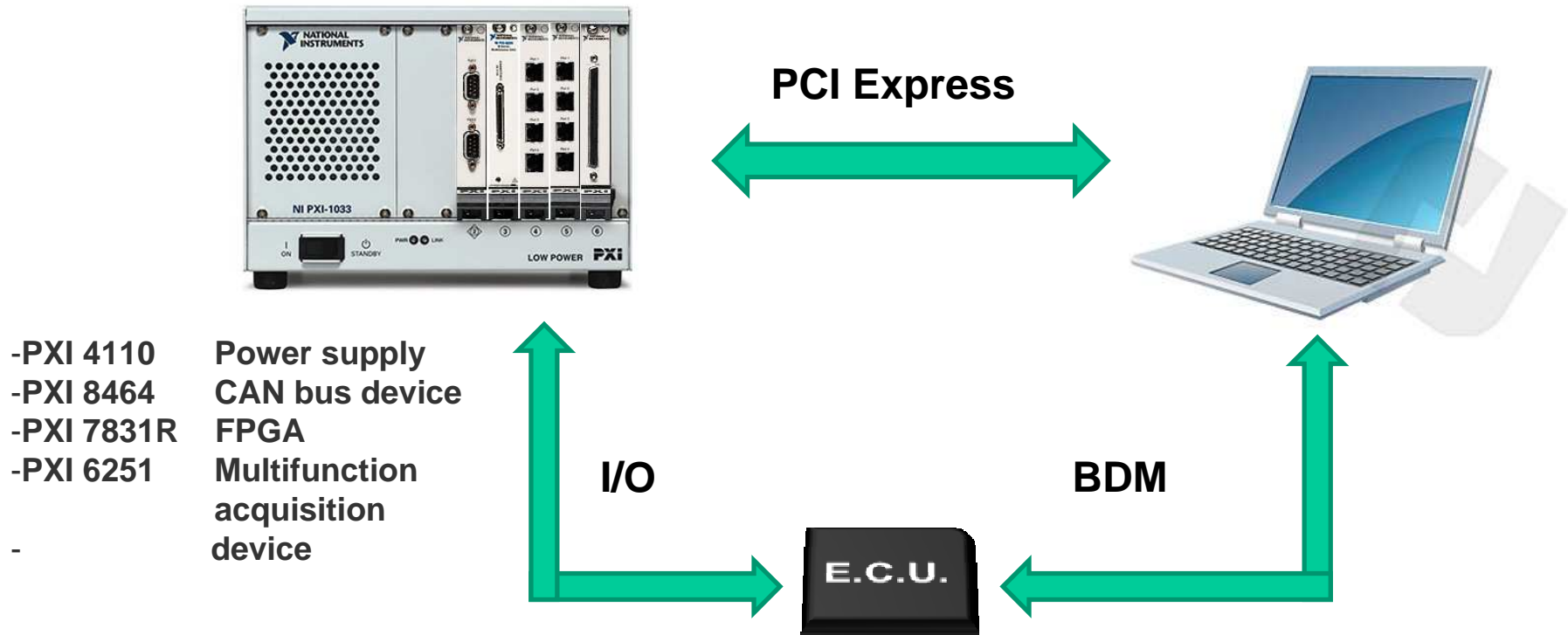
Allow test engineer to automate test cases execution

Reduce the cost impact of regression test



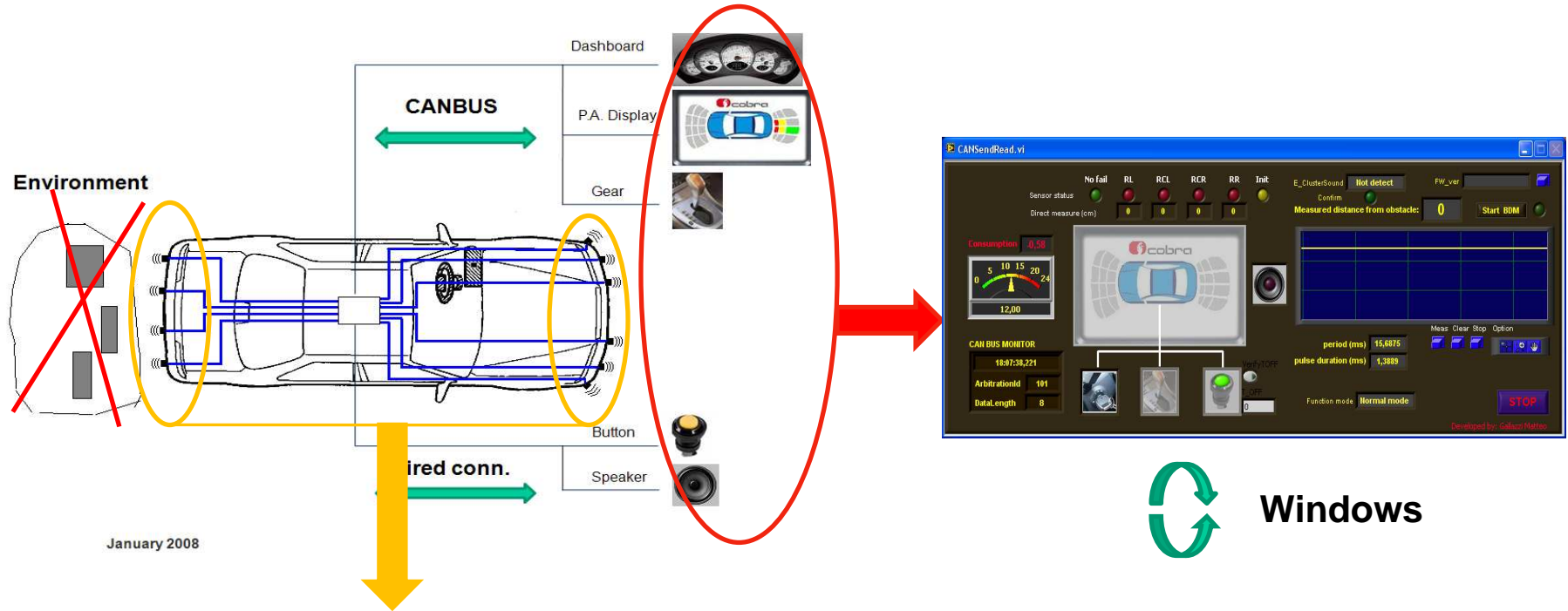


## ● The virtual system

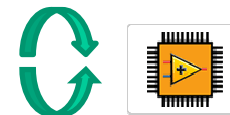
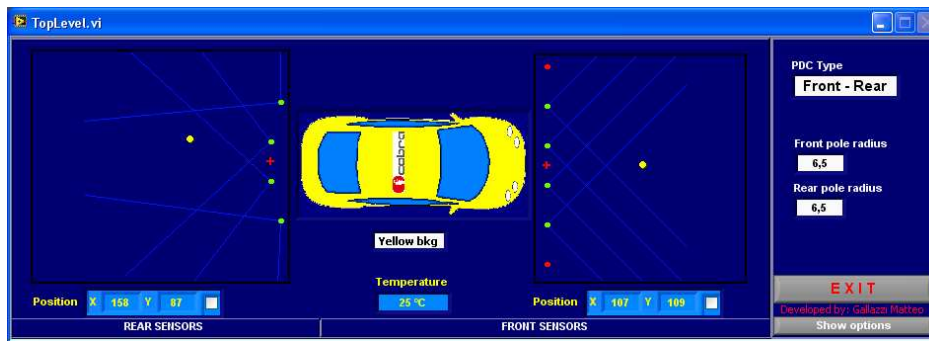




## The simulators



January 2008





## ● The monitors



This monitor is interfaced to:

- CAN card

- Shift gear simulation
  - Vehicle speed simulation
  - Display data monitor

- Power supply

- P&E BDM

- Measured distance
  - FW ver.
  - Any RAM address

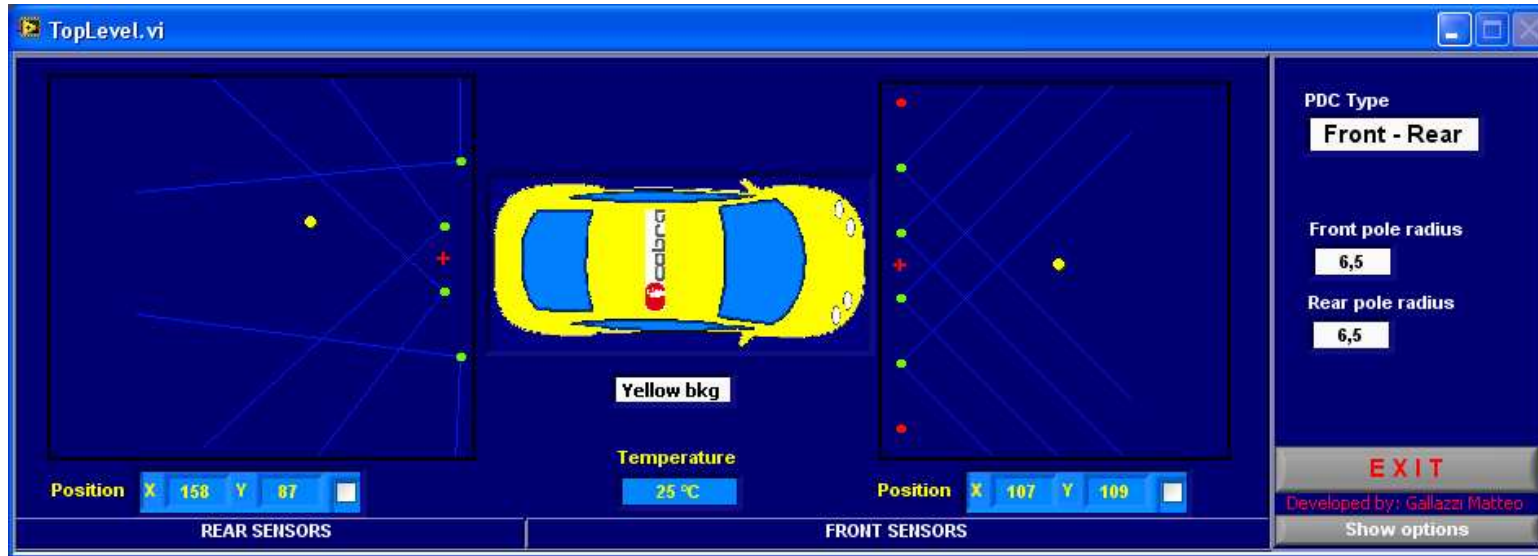
- Data acquisition device

- Speaker line monitor
  - Switch button simulation
  - Digital I/O monitor





● The monitors



This monitor is interfaced to:

- FPGA

- Simulate obstacle
- Simulate sensors failure
- Simulate sensors EEPROM
- Simulate protocol issues between sensors & E.C.U.



**● FPGA**

**Nonstandard LINBus protocol to reproduce**

**Flexibility**

**Cheaper than other commercial tools**

**Very fast communication schedule**

**Hardware design not needed**







## Simulation

**Obstacle simulation** ←

→ **BDM Data**


→ **Speaker output monitor**

→ **CAN information**





## Simulation



**TopLevel.vi**

Position X 158 Y 87

Temperature 25 °C

Position X 107 Y 109

**REAR SENSORS**

X 250	X 240	X 240	X 250
Y 50	Y 90	Y 130	Y 170
Dir. -132°	Dir. -95°	Dir. -85°	Dir. -50°
Tx ° 95°	Tx ° 95°	Tx ° 95°	Tx ° 95°
Rx ° 95°	Rx ° 95°	Rx ° 95°	Rx ° 95°

**FRONT SENSORS**

X 10	X 10	X 10	X 10	X 10	X 10
Y 10	Y 50	Y 90	Y 130	Y 170	Y 210
Dir. 90°	Dir. 90°	Dir. 90°	Dir. 90°	Dir. 90°	Dir. 90°
Tx ° 90°	Tx ° 90°	Tx ° 90°	Tx ° 90°	Tx ° 90°	Tx ° 90°
Rx ° 90°	Rx ° 90°	Rx ° 90°	Rx ° 90°	Rx ° 90°	Rx ° 90°

Sensors presence

Force NAD

Force PCI

Force RSID

Force CKS (Corrupted with +1)

Force negative answer

PDC Type: Front - Rear

Front pole radius: 6,5

Rear pole radius: 6,5

EXIT

Developed by: Gallazzi Matteo

Hide options

Show EEPROM page

HWver SWverM HWverL

1 6 1

InterTime (us): 0

ForcedNAD: 0

ForcedPCI: 0

ForcedRSID: 0

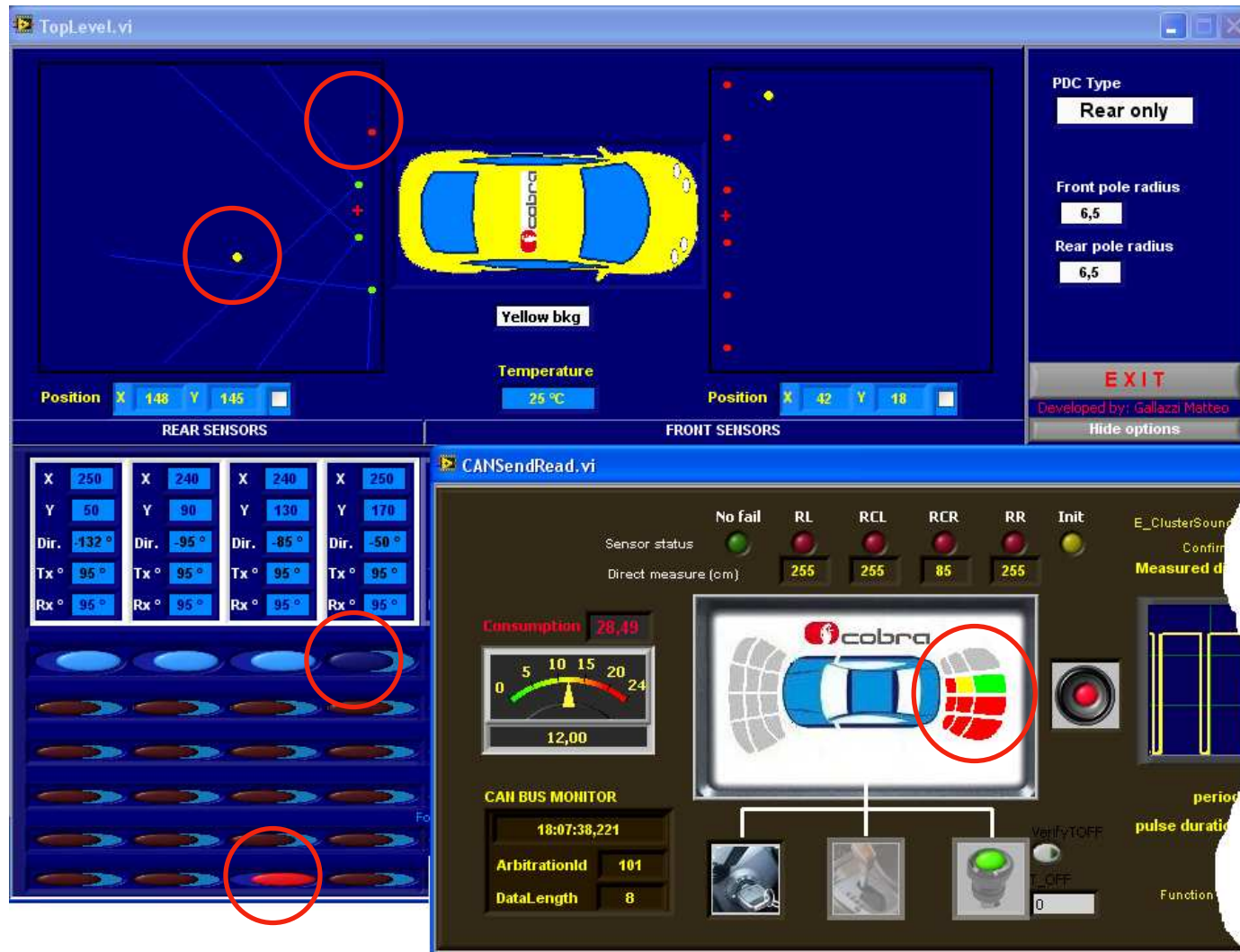
Error (no measure rec.)

Possibility to simulate protocol issues





- Simulation



**REAR SENSORS**

X 250	X 240	X 240	X 250
Y 50	Y 90	Y 130	Y 170
Dir. -132 °	Dir. -95 °	Dir. -85 °	Dir. -50 °
Tx ° 95 °	Tx ° 95 °	Tx ° 95 °	Tx ° 95 °
Rx ° 95 °	Rx ° 95 °	Rx ° 95 °	Rx ° 95 °

**FRONT SENSORS**

**CANSendRead.vi**

Sensor status: No fail, RL, RCL, RCR, RR, Init

Direct measure (cm): 255, 255, 85, 255

Consumption: 28,49

CAN BUS MONITOR

18:07:38,221

ArbitrationId: 101

DataLength: 8





### ■ **Simulation benefits**

**Create simulation environment that can not be easily reproduced in real environment**

**Reduce development time and helps engineer s to verify their code**

**Software isolation to avoid environmental interference**

**Reuse of simulation tools for testing purposes**





- **Software testing**

In order to assure high quality to customer, COBRA wants to perform all the test cases after each software change (100% regression test)

Software tests are fully automated to reduce regression test costs

A better quality is assured avoiding human interaction

Anyone can run the automatic test sequence

Test Reports are automatically generated







- **The automation**

**Sequence automation realized through NI TestStand**

**By using the VI Servers techniques we can create new test case as well as an operator that set the control and watch the result on the monitor**

**New test cases can be generated from a template using drag & drop functionality**

**The complete sequence is created using just 2 common step templates:**

**SET Control  
Get Indicator**

**Software knowledgment not needed to automate the test case**





- The automation

The screenshot shows the NI TestStand Sequence Editor interface. The main window displays a sequence diagram for 'AST\_RPAS\_v01.seq\*'. The sequence is structured as follows:

Step	Description	Settings
Setup (0)		
Main (7)		
SetSensorToWrite	Action, SetControl.vi	
SetAddressToWrite	Action, SetControl.vi	
SetDataToWrite	Action, SetControl.vi	
Wait	TimeInterval(1)	
WriteData	Action, SetControl.vi	
GET_INDICATOR	Action, SetControl.vi	Pre Expression, Do Not Rec...
SET_CONTROL	Call SUBSeq_RPASStart...	
<End Group>		
Cleanup (0)		

The 'Subsequences' list on the left includes 'SET\_CONTROL' and 'GET\_INDICATOR', which are circled in red. A red arrow points from this list to the 'SET\_CONTROL' step in the main diagram. The 'Sequences' list at the bottom shows the following entries:

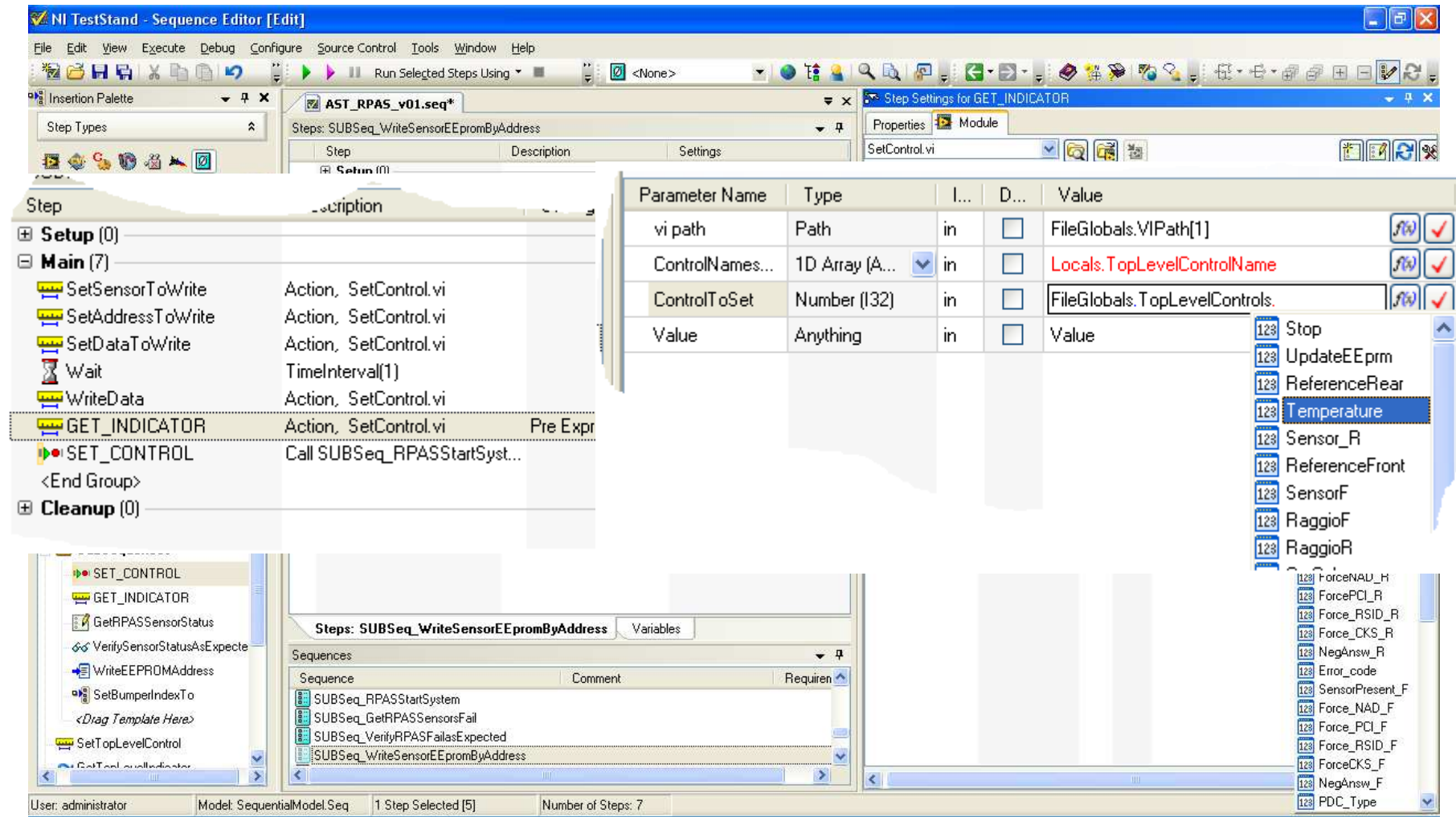
Sequence	Comment
SUBSeq_RPASStartSystem	
SUBSeq_GetRPASSensorsFail	
SUBSeq_VerifyRPASFailasExpected	
SUBSeq_WriteSensorEEPROMByAddress	







## The automation



The screenshot displays the NI TestStand - Sequence Editor [Edit] interface. The main window shows a sequence editor for 'AST\_RPAS\_v01.seq\*'. The 'Steps' pane on the left lists the sequence structure:

- Setup (0)
- Main (7)
  - SetSensorToWrite (Action, SetControl.vi)
  - SetAddressToWrite (Action, SetControl.vi)
  - SetDataToWrite (Action, SetControl.vi)
  - Wait (TimeInterval(1))
  - WriteData (Action, SetControl.vi)
  - GET\_INDICATOR (Action, SetControl.vi) Pre Expr**
  - SET\_CONTROL (Call SUBSeq\_RPASStartSystem...)
  - <End Group>
- Cleanup (0)

The 'Step Settings for GET\_INDICATOR' dialog is open, showing the following parameters:

Parameter Name	Type	I...	D...	Value
vi path	Path	in	<input type="checkbox"/>	FileGlobals.VIPath[1]
ControlNames...	1D Array (A...	in	<input type="checkbox"/>	Locals.TopLevelControlName
ControlToSet	Number (I32)	in	<input type="checkbox"/>	FileGlobals.TopLevelControls...
Value	Anything	in	<input type="checkbox"/>	Value

The 'Value' field for the 'Value' parameter is expanded to show a list of options:

- Stop
- UpdateEEPm
- ReferenceRear
- Temperature**
- Sensor\_R
- ReferenceFront
- SensorF
- RaggioF
- RaggioR
- ForceNAU\_H
- ForcePCI\_R
- Force\_RSID\_R
- Force\_CKS\_R
- NegAnsw\_R
- Error\_code
- SensorPresent\_F
- Force\_NAD\_F
- Force\_PCI\_F
- Force\_RSID\_F
- ForceCKS\_F
- NegAnsw\_F
- PDC\_Type

The 'Sequences' pane at the bottom shows the following sequence list:

Sequence	Comment	Require
SUBSeq_RPASStartSystem		
SUBSeq_GetRPASSensorsFail		
SUBSeq_VerifyRPASFailasExpected		
SUBSeq_WriteSensorEEPromByAddress		

At the bottom of the window, the status bar indicates: User: administrator, Model: SequentialModel.Seq, 1 Step Selected [5], Number of Steps: 7.





## Final result

The screenshot displays the NI TestStand interface on the left and a Mozilla Firefox browser window on the right showing a test report.

**NI TestStand - Sequence Editor [Running...]**

Step	Description	Settings	Status
SetPolePositionTo_41cm_RCL	Action, SetCont...	Pre Expression, Do Not Re...	Done
Wait	TimeInterval(4)	Do Not Record Result	Done
Get_RPASDisplay_RCL	Action, GetIndic...	Do Not Record Result	Done
TC014_Verify_RPASDisplay_RCL_2	Pass/Fail Test		Passed
SetPolePositionTo_40cm_RCL	Action, SetCont...	Pre Expression, Do Not Re...	Done
Wait	TimeInterval(4)	Do Not Record Result	Done
Get_RPASDisplay_RCL	Action, GetIndic...	Do Not Record Result	Done
TC015_Verify_RPASDisplay_RCL_3	Pass/Fail Test		Passed
SetPolePositionTo_30cm_RCL	Action, SetCont...	Pre Expression, Do Not Re...	Done
Wait	TimeInterval(4)	Do Not Record Result	Done
Get_RPASDisplay_RCL	Action, GetIndic...	Do Not Record Result	Done
TC016_Verify_RPASDisplay_RCL_3	Pass/Fail Test		Passed
VERIFY_RPASDisplay_RCR	Action, SetCont...	Pre Expression, Do Not Re...	Done
Wait	TimeInterval(4)	Do Not Record Result	Done
Get_RPASDisplay_RCR	Action, GetIndic...	Do Not Record Result	Done
TC017_Verify_RPASDisplay_RCR_0	Pass/Fail Test		Passed
SetPolePositionTo_120cm_RCR	Action, SetCont...	Pre Expression, Do Not Re...	Done
Wait	TimeInterval(4)	Do Not Record Result	Done
Get_RPASDisplay_RCR	Action, GetIndic...	Do Not Record Result	Done
TC018_Verify_RPASDisplay_RCR_1	Pass/Fail Test		Passed
SetPolePositionTo_81cm_RCR	Action, SetCont...	Pre Expression, Do Not Re...	Done
Wait	TimeInterval(4)	Do Not Record Result	Done

**Report - Mozilla Firefox**

Report

UIT Report

- Station ID: PPRI06
- Date: Juedi 16 novembre 2009
- Time: 15:48:20
- Software Tester: administrator
- Execution Time: 713.061 seconds
- Number of Results: 215
- UIT Result: Terminated

Begin Sequence: MainSequence  
(C:\COBRA\Test\_Del\_Software\SBP18CPP\_0370\_PDC\_CobraPlatform\Software\TestSequence\AST\_RPAS\_v00.seq)

Step	Result
StepFWvrchReport	Passed
FWvrchReport	Passed
TC001-004 (VerifySystemActivation)	Passed

Begin Sequence: TC001-004\_VerifySystemActivation  
(C:\COBRA\Test\_Del\_Software\SBP18CPP\_0370\_PDC\_CobraPlatform\Software\TestSequence\AST\_RPAS\_v00.seq)

Step	Result
SetRPASSwitchOFF	Passed
TC001 (VerifySystemNotActive)	Passed
SetReverseGear	Passed
TC002 (VerifySystemNotActive)	Passed
ResetReverseGear	Passed
GetRPASSwitchON	Passed
TC003 (VerifySystemNotActive)	Passed





➤ **Conclusion**

**Full coverage of Functional Requirements**

**Test Cases execution fully automated**

**Software knowledgment not needed to add test cases**

**Reduction of Project costs thanks to grey box test**

**Reduction of Regression Test costs**

**Higher quality for our customers**

**TARGET HAS BEEN ACHIEVED**

