Relying on dSPACE: Automated Test Platform for EV Controllers in BJEV

R&D Institute, BJEV - Xi Liu
2019.11
Table of Contents

1. Introduction of BAIC and BJEV
2. BJEV Automated Test System
3. Automation on Multi-controller HIL
4. Future works
Introduction - BAIC (Beijing Automotive Industry Corporation)

**Leading** car manufacturer and service provider in CHINA.

Best brand layout, wide range of brands, Diverse range of business, Leading company for the BEV business

One of **TOP 5** vehicle corporations in China, ranking **129** in 2019 Fortune Global 500

- **Peking**
- 5 largest OEMs in China
- 130,000 employee worldwide
- 60 years history

**2019 Jan ~ June**
- Car sales: 1,109,000
- Revenue: RMB 246.45bn

**2019 Target**
- Car sales: 2,550,000
- 营业收入: RMB 510bn
BJEV is a subsidiary of BAIC, with the commitment of Battery Electric Vehicle (BEV) business.

Cumulative sales volume is over **440,000**, Yearly EV sales volume **NO.1** in China from 2013 to 2018, **NO.2** worldwide in **2018**.

From 2018, BJEV leads the construction of **National** New Vehicle Technology Innovation Center
Introduction - HIL Test Lab

BJEV Test Center construction area: over 50,000 m²; Number of Labs: 88; cost: RMB 2.05 bn (€ 264.79 million)

HIL Test Laboratory: own 25 dSPACE equipment with test capability of LV (VCU, MCU) and HV (BMS)

HIL lab of the Largest scale and highest automation degree in china.

22 July 2019, BJEV Test Center officially open for use

Controller Test Lab (HIL Lab)

Vice Chairman of National Committee of the Chinese People’s Political Consultative Conference Wan Gang visit HIL Lab

Vice Chairman of CPPCC Wan Gang visit HIL Lab
Automated Test System - Background

Why we need to develop our Automated Test System?

New test engineers need to spend a lot of time learning how to operate the HIL equipment, and how to write the test script.

In case of more than one HIL equipment brand, the learning stage gets even longer.
Automated Test System - Overview

Objectives of Automated Test System:

- **Normalization and automation** of each test phase
- Improvement of HIL equipment usage **efficiency**
- **Time cost reduction**
Automated Test System – Data flow

- Software SOR
- Test case Designer
- Test cases
- Test environment
- Test result
- Test Execution Tool
- Equipment Auto arrangement
- Test Management System
- Software SOR
- IO interface of ECU Communication protocol
- Test report
- IO interface of ECU Communication protocol
- Environment Builder
**Automated Test System – key-word driven test cases**

**Test case definition in our system**
A test case only provide signals and values, is not executable by its own. The execution of it needs Test Execution Tool (Test script).

**2-D excel format enabling more test item integrated**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Signal 1</td>
<td>Signal 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>W10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>S10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>C10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>V10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Time axis** | **Signals to Write** | **Signals to Read**
---|---|---

**Key word library and functions**

<table>
<thead>
<tr>
<th>Library</th>
<th>Supported Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>keep expectation during a certain period reach expectation in a certain period send curved values …</td>
</tr>
<tr>
<td>Data</td>
<td>save value Equations &amp; expressions …</td>
</tr>
<tr>
<td>UDS</td>
<td>Read data by identifier DTC Clear Diagnostic Security Access Write data …</td>
</tr>
<tr>
<td>Bootloader</td>
<td>enter bootloading state …</td>
</tr>
<tr>
<td>FIU</td>
<td>FIU functions</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

---
Automated Test System – Key-word driven test cases

Strength:

- Fully Mother language based test case, **NO EXTRA requirements** Programming language for test engineers, **lower** the requirements for engineers' ability
- **High understandability, high reviewability**
- Modular **reusability**, high **extendibility**, functions already integrated: UDS tests, FIU tests, Bootloader, …
- Basis of test automation
Test case design and generation process

Test requirements

- DSL about execution & Graph algorithms
- Computer language

Test flow (formal language)

Test case design techniques

- Black box & White box techniques
- Previous experience

Test items

SOR (in natural language)

Positive testing
- Check ECU does what it is supposed to do

Negative testing
- Test ECU does not do anything that it is not supposed to do

Test case configuration

Black box & White box techniques

Previous experience

Computer language
Automated Test System – **Test case Designer**

**Test techniques integrated:**

- Boundary value analysis
- Modified condition / decision coverage (MC/DC)
- Positive testing & negative testing
- Equivalence partitioning
- Path analysis
- Modified condition / decision coverage (MC/DC)

**Outcomes:**

- Compared with manual approach, number of test conditions increased **averagely by 100%**, test case design time **shorten by 50%**, test case execution time **shorten by 20%** (due to the high redundancy time rate of manual method)
- Generated test cases can be **executed straightly** with no more work

**Strength:**

- High **understandability**, high **reviewability**: test cases easy to be reviewed by software developers
- **Test coverage Guarantee**: Less experienced engineers can also get test cases of high quality;
- **Time cost reduction**: both test case design period and test execution time decreased
Automated Test System – Test environment builder

Test environment building process: Manual VS Automatic

Current state:
- **Platform based** test environment model (Simulink) and ConfigurationDesk project, applicable for all controllers with same hardware interface
- Automatically generate bus and IO interface model based on DBC file and IO interface file (m script).
- **Parameterization**: model parameters configuration in an excel file, of high convenience.
Automated Test System – Test execution tool

WHY a self-developed Test execution tool is needed?

Test execution tool

Focusing on key functions

- Project download
- Equipment connection
- Signal reading & writing

UPs and DOWNs of commercial software

- Powerful tool with plenty of functions!
- Complicate to operate
- Need time to learn
- Not all functions is needed

Easy to operate
- No need to learn
- Specialist needed in test team
Automated Test System – Test execution tool

Software Architecture

Operations:

- Build connection
- Project download
- API initialization
- Calibration device
- UDS device
- ECU connection
- Test case reading
- Signal handle correctness

- Test bench setup
- Peripheral device setup
- Test environment check
- Test case reading
- Test case check

- Test case execution
- Test completio

- Overall test report generation

- Recover test bench to initial state
- Recover all signals
  Involved to their Default state
- Test result generation

- Test report generation

  of all test cases

  of single case

  of single case
- Signal sending & reading
- Result determination

Enumerate all test cases
Automated Test System – Test execution tool

Development history

Generation 2 (from 2019): Based on RTT toolkit (no time latency)

Strength

- independent of test data and test script, test script (test execution tool) can be maintained by a few engineers, most engineers do not need to know the scripting language.
- High modular extendibility: CCP/XCP calibration, UDS diagnostics test, FIU test has already integrated
- High platform extendibility: other HIL even SIL, MIL device or software supportable
- Easy to operate: only a few configurations required before execution
Automated Test System: Test Management System (overall)

Physical layer

Monitor center

Application layer

User management

Configuration
Task application
Project management
Test implementation
Work implementation
Information center

Remote control
Test coverage analysis
Result management
Task distribution

Resource library

Maintenance & recovery

User interface

Physical layer

Application layer

Monitor center
Automated Test System: Test Management System (process control)

Test process

- Task application
- Test inputs upload
- Test environment upload
- Test case upload
- Test execution
- Test report generation

Bug tracking process

- Bug propose
- Bug confirmation
- Cause location & responsibility distribution
- Bug fixing
- Verification
- Problem close

Key Features:
On-line processing coverage of overall test process
Process editable
Traceability: function requirement – Test case – test result
Automated Test System: Test Management System (data statistics)

**Data Collection**
- Project statistics: Test period, Number of tests, ...
- Resource statistics: Usage rate, Task arrangement, ...
- Labor statistics: Work load, KPI, ...
- Bug statistics: Processing period, High risk function, Bug trend, ...

**Data Analysis**

**Process Improvement**
Automated Test System: Achievement

**Hardware status**

Quantity of dSPACE equipment and HIL test hours in the last 5 years

<table>
<thead>
<tr>
<th>Year</th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
<th>2018</th>
<th>2019(1-10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dSPACE quantity</td>
<td>1</td>
<td>7</td>
<td>11</td>
<td>15</td>
<td>25</td>
</tr>
</tbody>
</table>

**Software status**

Based on the self-developed Automated Test System, the automated rate of HIL process is 82% (time cost reduction rate)

**RELIABLE!**

HIL equipment in our lab is 7x24 running.

<table>
<thead>
<tr>
<th>Year</th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
<th>2018</th>
<th>2019(1-10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total test duration(h)</td>
<td>3780</td>
<td>25680</td>
<td>51840</td>
<td>66240</td>
<td>93460</td>
</tr>
</tbody>
</table>

Last 5 years, a total of **241,000 hours** test was carried out = a single HIL equipment runs for **28 years** non-stop

**No accident** occurred!

**VALUABLE!**

4500 software bugs were detected totally (by Oct 2019)
Success of component level HIL test with dSPACE drive us to extend dSPACE application to SYSTEM LEVEL TEST.

Current state:

- VCU, BMS, MCU, CCU controllers are integrated in the system
- Automated Test System works well
Automated on Multi-controller HIL problems and future work

**Common problems** of Multi-controller HIL:

- Long environment building period
- Lack of description files (SOR)

**Solving approach:**
Automated Test system reduce time cost of environment building process (DONE)
System SOR extraction from component SOR (In implementation)
Future works

1. Automated test system: to establish connection between natural language (software SOR) and pre-defined DSL, Test cases can be generated fully automatically based on the SOR. That means **the whole test process** realize **automation**.

2. Multi-controller HIL: currently it is tested on one project for trial operation. In future the application will be extended to all projects, partly replacing vehicle test.
THANK YOU