Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 58(72), Fascicola 2, 2013

Automated test stand for automotive sensors

Daniel Doru Popa[1]

Abstract – In the automotive world of today, everything is very dynamic, each car manufacturer wants to be the first on the market with new functions and features for the cars, as a consequence all automotive suppliers development schedules are very tight. In order to shorten the testing time(some tests need to be performed on a large lot of sensors ~200), in the development phase it is preferred to build up automated test stands , in this way the results are well documented, accurate, and obtained ten times faster than if the test would be performed manually.

Keywords: automotive, sensors, test stand, time.

I. INTRODUCTION

In the development stage of an ECU (Electronic Control Unit) the production line is not equipped from the beginning with a functional test stand due to the fact that the PCB (Printed Circuit Board) layout and shape can change until the final version of the ECU and these test stands are very expensive ~ 50 000 euro, and are specifically designed for each project/sample variant, but the ECU's have to be tested and the results are expected to be accurate, fast and well documented, this are the reasons for building up the automated test stands.

These functional test stands are used in development for tests that have to be performed on a large lot of samples (100 - 200) in order to shorten the testing time and improve accuracy and documentation of the results.

The functional test stands used in development are not that expensive (2000-5000 euro) and are very versatile and most of them can be reused for any project/sample variant with few modifications (nails adaptor layout, load box, and testing software).

II. AUTOMATED TEST STAND DESIGN

In order to design an automated test stand, the following information are needed: the type of the measurements that have to be performed, the number of the signals and the testing software that shall be used, these are the most important factors in order to determine the raw effort that will be needed to build up the test stand.

First step after the requirements are clear is to draw a block diagram of the test stand in order to determine what measurement equipment and interfaces will be used.

As it can be seen in Fig. 1., the automated test stand used in this case is composed by a nails adapter (the fixture in which the PCB is placed), a DB25 connection cable(used to route the signals from nails adapter to the relay matrix board), a relay matrix board (a switching unit used to route each signal to the proper measurement equipment), a power supply (to power up the ECU), a load box (to simulate different load conditions), a CAN (Control Area Network) hardware box (to communicate with the computer via CAN), a digital multimeter (for measuring the DC voltages), an oscilloscope (for timing and frequency measurements), a GPIB(General Purpose Interface Bus) to USB(Universal Serial Bus) convertor and a computer used to run the testing software and store the test results.

This test stand can be used for any other project/ sample variant, with very small modifications (a different nails adapter layout that would fit with the new PCB layout, another load box - if is the case - , and the adaptation of the testing software if the values/signals are different that those used in the previous project / sample variant). All the measurement equipment, interfaces, and even the basic architecture of the testing software can be reused, therefore the cost of the test stand is higher in the beginning, for the first test stand then the needed

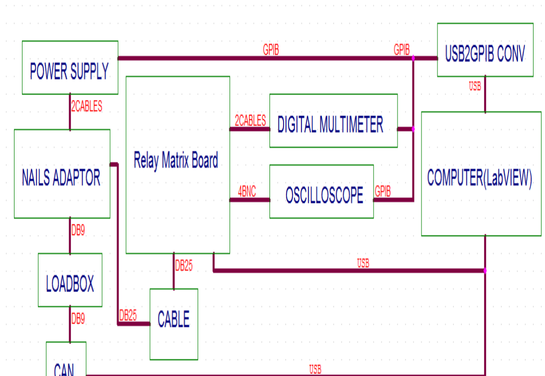

Fig. 1. Automated test stand block diagram

[1] "Politehnica" Uiversity of Timişoara, Electronics and Telecomunications Faculty, Advanced Tehniques in Electronics Master, Bd. V. Pârvan Nr. 2, 300223 Timişoara

BUPT

modifications for a different project / sample variant are quite cheap (200 - 500 euro).
Using this functional test stand it's possible to perform also test in temperature, in order to verify the behavior of the samples at different temperature values (-40° C and 85° C are the limit temperature values used in automotive applications for ECU's placed in the vehicle compartment [2]).

III. TESTING SOFTWARE

Probably the most important part of the functional test stand is the testing software, because it is the one that is responsible for the communication with all the measurement equipments, and it's the most important factor for the test duration, if it's designed in a proper way, the testing time is reduced at minimum possible.
One of the most used programming languages for functional test stands is LabView, a graphical programming language called "G". It is developed by National Instruments and is used for data acquisition, instrument control and analysis of results, in particular in the field of industrial automation. This program is available for various platforms: Windows, Linux or MAC OS, which gives the advantage to be accessible to a wide variety of users all over the world.
Language technique is to uses "dataflow" concept in the execution of a program. The program is based on a block diagram, in which the user connects the different terminals by wires to different nodes (symbols), Fig.2. These wires propagate variables. Execution rule says that a node is executed only when all its terminals are available to input values, from this rule it can be observed that the execution of the program is not done after executing the code from left to right or line by line from top to bottom (as in programming languages text you write lines of code). It may happen that multiple nodes simultaneously receive required input values, and begin execution simultaneously (in parallel). This feature is used heavily in automated test stands applications.

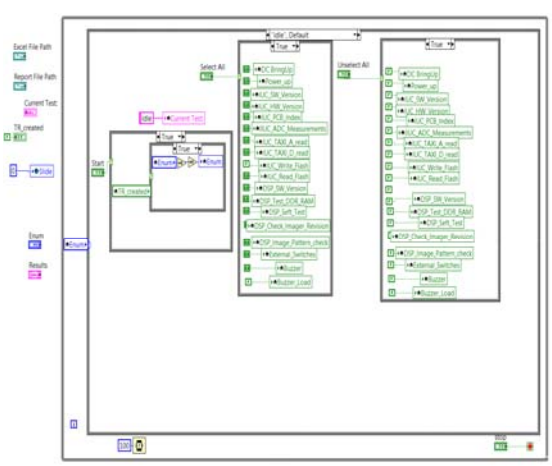

Fig. 2. Testing software block diagram

The GUI (Graphical User Interface) of the program is called front panel, Fig.3, while the executable program (which works to perform the results displayed on the front panel) is called block diagram. The user develops the application software through a hierarchical construction of Vis (Virtual Instruments). A virtual instrument is a graphical software package that looks and acts like a instrument, and each of these virtual instruments have the following three components: front panel (buttons, switches , dials of instruments, indicators, etc.), block diagram (executable program consists of nodes interconnected terminals and wires) and the connector (method VI 's interconnection to other virtual instruments), [1].
The testing software is design in a state machine model, each individual test, corresponding to a state of the state machine, in this way is very easy to add/delete tests and all the tests are performed sequentially, test after test, and none of the tests influences the other, each test can be seen as a separate part of the testing program, that does not influence the functionality of the other tests.
The Labview program is also responsible to evaluate the results and take the pass/fail decision for each test result, this is done by using some thresholds for each measurement and if the test result is in the proper range (between thresholds) it is set as passed otherwise is set as failed. In the end all performed tests have to pass in order to say that an ECU is functioning in the proper way.
Beside all the things listed above, all the information related to the tests have to be documented in a test report file (time and date when the test was performed, sample number, hardware and software version, test thresholds, test results and pass/fail status of each test). Testing software has to generate automatically a well documented test report that can be afterwards analyzed and understood by anyone who could be interested in test results.

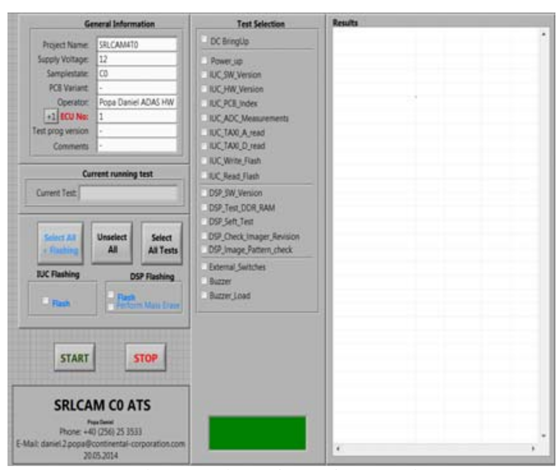

Fig. 3. Testing software front panel]

BUPT

IV. RESULT DOCUMENTATION

Each test report file contains all the important information regarding the ECU that was tested and testing parameters, in order to be able to repeat the test in same conditions any time is necessary, Fig.4. A big advantage of the automated test stand is that the test reports are generated automatically by the testing software, in this way the content of the test report is always presented in the same format, and the human mistake is avoided (the computer will always write the test results values in the correct field, unlike humans, especially when a large number of tests are listed in a single document, is very likely that a human would make some mistakes and will place the test results in the wrong fields).
After all tests are performed on the entire lot of samples, all test report files are imported in a special Microsoft Excel file that use some macros in order to import and display the content of the test report files in a single file in a readable format, Fig.5. Using that Excel file with all the results, conclusions can be drawn regarding the maturity level of the design,

```
[header]
filename:SRLCAM_C0_C0V06L088.dat
time:Tu May 20 10:57:12 2014
temperature:25°C
icname:SRLCAM
icsuppl:12
iccode:C0
icnumber:03
testprog:Daniel Popa
testprogversion:
comment:-


[teststep1]
name:DC_BringUp
spec:spec
comment:cmnt
first_res:1
last_res:20
struct:value1:unit1:pass1:lo_lim1:hi_lim1:res_name1:condition1:
1:3.0393250E-1:V:p:2.5000000E-1:4.0000000E-1:Current Consuption @12V::
2:1.1093975E+1:V:p:1.0750000E+1:1.2200000E+1:UBATT_DE::
3:1.0995147E+1:V:p:9.5000000E+0:1.2200000E+1:UBATT_SW::
4:3.3077797E+0:V:p:3.2020000E+0:3.3900000E+0:3V3_SMPS::
5:1.2064420E+0:V:p:1.1800000E+0:1.2200000E+0:1V2_SMPS::
6:1.8029107E+0:V:p:1.7600000E+0:1.8400000E+0:1V8_SMPS::
7:5.0174369E+0:V:p:4.7900000E+0:5.2100000E+0:5V0_SMPS::
8:1.7981830E+1:V:p:1.7280000E+1:1.8680000E+1:18V0_SMPS::
9:1.7885341E+0:V:p:1.7350000E+0:1.8330000E+0:1V8_SW::
10:1.2086301E+0:V:p:1.1680000E+0:1.2210000E+0:1V1_SMPS::
11:2.8214912E+0:V:p:2.6910000E+0:2.9050000E+0:2V8_LIN::
12:1.5808005E+0:V:p:1.1780000E+0:1.7500000E+0:IUC.ADC_TEMP0_LASER::
13:1.5262714E+0:V:p:1.1780000E+0:1.7500000E+0:IUC.ADC_TEMP0_DSP::
14:1.5322111E+0:V:p:1.1780000E+0:1.7500000E+0:IUC.ADC_TEMP0_IUC::
15:1.2026100E+1:V:p:1.1400000E+1:1.2600000E+1:SRL.VPIN::
16:1.3627822E+0:V:p:1.3000000E+0:1.4000000E+0:1V35_LIN::
17:3.3329718E+0:V:p:3.2670000E+0:3.3330000E+0:VREF::
18:3.2895502E+0:V:p:2.0500000E+0:3.4500000E+0:VT_REF::
```

Fig. 4. Test report file format

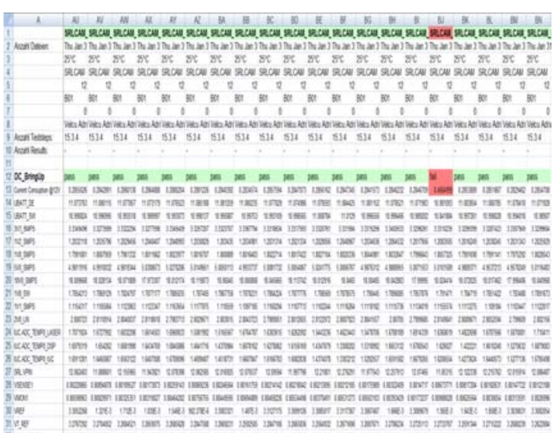

Fig. 5. Test report files imported in the special excel file

possible issues can be detected and be the input for the changes done for the next sample variant.
Using some Microsoft Excel macros the cells are colored in green or red for the pass/fail results (as it can be seen in the below picture), in order to be easily seen the most important things (failed tests/ ECU's).

V. CONCLUSIONS

Automated test stands are used in development for functional testing of the samples until the series functional test stand from the production line is available.
The setup can be used for flashing the ECU software and then perform the functional tests on the ECU, using the same fixture both tasks can be fulfilled.
These functional test stands are very useful because they can be used to test a large lot of samples (~ 200), they provide accurate and comparable results (the test procedure is always the same), they generate the test reports automatically and avoid human mistakes, and the tests are perform very fast (at least ten times faster than tests performed by hand).
The test stand is very flexible and can be very simple or complex depending of the requirements regarding number of signals that have to be measured and the type of measurements that have to be performed.
The setup is very adaptable, and almost all the measurement equipments and interfaces can be used for another sample variant / project with minimum cost.

REFERENCES

[1] http://zone.ni.com/dzhp/app/main
[2] Continental Automotive Specifications 2014

BUPT