

Tracking Automotive Electronics Related Sensory and Their Traceability by Boundary Scan Technology

Wang Shun Shen Peter^{1,2,3}, Wang Yin Tien¹, Chao Chong Lii¹, Yang Wei Bin², Lee Tzung Hang^{1,2}

¹Department of Mechanical and Electrical Engineering, Tam Kang University, Taipei, Taiwan, ROC

²Department of Electrical and Computer Engineering, Tam Kang University, Taipei, Taiwan, ROC

³JTAG Technologies Private Limited, Eindhoven, The Netherlands

Email address:

IEEE1149@JTAG.com.sg (Wang Shun Shen Peter), ytwang@gms.tku.edu.tw (Wang Yin Tien), clchao@mail.tku.edu.tw (Chao Chong Lii),

robin@ee.tku.edu.tw (Yang Wei Bin), zouhan@mail.tku.edu.tw (Lee Tzung Hang)

To cite this article:

Wang Shun Shen Peter, Wang Yin Tien, Chao Chong Lii, Yang Wei Bin, Lee Tzung Hang. Tracking Automotive Electronics Related Sensory and Their Traceability by Boundary Scan Technology. *American Journal of Electrical and Computer Engineering*.

Vol. 5, No. 2, 2021, pp. 88-97. doi: 10.11648/j.ajece.20210502.16

Received: November 8, 2021; **Accepted:** November 22, 2021; **Published:** November 24, 2021

Abstract: This paper proposes several Python-based test methods to study a series of MCU-associated sensors. In light of today's automotive electronics, many electric systems are involved in modularized bare dies such as power management ICs along with embedded programming. As to the application of autonomous driving like the ADAS system, having many Multiple-Chips Modules (MCM) which are popular in SoC related functional boards. It is necessary to have these electrical systems follow stringent safety standards and regulations by the International Electrotechnical Commission (IEC). That is to ensure the electronic systems to be designed, implemented, operated within safety range. That is traceable according to the prerequisite Safety Integrity Level (SIL) besides the Automotive Safety Integrity Level (ASIL) towards five classes. These classes further evolved into higher requirements such as IEC-61508 and more recently the ISO-26262. All of these, have driven many companies to research and develop a suite of methods and strategies to solve some of the "reliability" associated issues, especially in the area of autonomous driving. This paper proposes the boundary scan based test methodology to cop with government's regulations by tracking some of the major sensors in responses to motion, temperature and dynamic forces based on a JTAG's sensor board in case study A. Thereafter, using an NXP's automotive board to trace on board electronics, mainly the CPU and MCU via the MODBUS is made possible to log and record test results through VB.XML.NET. These languages speak uniformly to the relational data base in one "DataSet" by in-memory cache of the data retrieved from a database, in case study B. Both are realized by the JTAG's Functional Test (JFT) system, to log the "pass or fail" results in a sequential execution. This paper illustrates how-to reach this goal through the process in use of the public-domain based open frame-ware, and to demonstrate them on a JTAG's Provision platform throughout the development stages.

Keywords: ADAS, Boundary Scan, MCM, JTAG, BST, BIST, Automotive Electronics, Autonomous Vehicles

1. Introduction

Testing automotive electronics is a system work, it is because the circuit boards are usually equipped with complex CPUs, MCU in combination along with a slew of memory chips. They are tested by the conventional Boundary Scan-based Technology (BST) plus the emulation the Built-In-Self Test (BIST) as per IEEE1149.1.6 and dot 7 based standards. In the case A), it involves many sensors under test on the schematics; this study first introduces how

FCT extracts dynamic data from various sensory devices in response to temperate, pressure, vibration and in contacting to be tested functionally, and they are validated to pass or fail to their spec. In case A), it is proposed to use a typical NXP's LPC2468FET208, a master chip working around various sensors. In case b), another NXP's iMX6 chip is proposed In the case B). Both case A) and B) have JTAG o run traceability study. Both are accessible to the common Test Access Port (TAP). The indifference of TDI, TDO, TCK, TMS, TRST for BST, or just TCK and TMS for BIST. [17] The iMX 6 chip family is designed to have the

embedded system, and which is equipped with Built-In-Self-Test (BIST) for Signature Analysis. On the other hand, the Boundary Scan Test (BST) is to qualify these on-board dies [3] and MCMs [1] before burn-in test [5], are to be free of manufacturing defectives [6]. This proposal incorporates JTAG's Boundary Scan Diagnostics (BSD) and DLL Application Program in Interface (API) with a series of software modules by taking advantages of Python programming and MODBUS-based remote terminal unit (RTU), through Microsoft's NET framework. This approach is capable of transmitting and receiving the RS-485 based protocols via VB.Net-XML links. The test results are accumulated by JTAG's AEX Manager System for self-test [7], and thus to complete the traceability procedures in case B). Accomplished records log on a JTAG based Provision System, i.e., it uploading the Pass or FAIL data into an EEPROM automatically. This approach is to ensure both cases under test working around one data base by JTAG even though they belong to different technologies. [18]

2. Python Based Scripting Applications: JTAG Functional Tests: Case Study A)

In Figure 1, it demonstrates a JTAG based sensory board [4], consisting of four sensors such as 1) temperature sensor (I2C-TCN75AV) 2) pressure sensor (I2C-AT42QT2169) 3) vibration sensor (I2C MMA7660FCT) and 4) contact sensor (ADS7843). To do so, it is required to measure an ADC output and to check the digital voltage readouts of the temperature sensor within the expected limits. Besides, this UUT is expected to log on the time-set in Year/Date/Time as to set the CPU based Real Time Clock device. Other situations may require the test to make a 'decision' on the next GO, depending on the reading back data if necessary to cut off certain operations. Once these declarations are made, they are transmissible and receivable through an I2C bus [2]

based master and slave architecture, i.e., the NXP chip is the master and its associated sensors are slaves. In current approaches of the ADAS systems, it has been around for some time, started as early as 1995, the Mitsubishi Motors have introduced so called the Parking Distance Control (PDC). Later, the Toyota has developed the Cruise Control (ACC), both may not be as effective as today's Lidar but it is compact and weatherproof in a long-range detection of blind spots at 79 GHz RF frequency range. [10, 12].



Figure 1. JTAG Functional Test (JFT) based sensory board.

2.1. Temperature Sensor Measurement by JFT

The following Figure 2, it illustrates how the temperature measurement is developed for Microchip TCN75A I2C based temperature sensor, basically the JFT defines a specific set of Python functions dedicated to support functional instructions such as WriteVar, GetVar, DeclareGroup, WriteGroup, GetGroup, GetDeviceInstructionList, etc. drivers, used to activate the boundary scan based TAP pins, to write or read values to and from the device based on the IEEE1149.1 specific instructions such as Clamp, High-Z, Bypass, Infra, Inter, Ext. etc., and to allow the JFT access any register available within the boundary-scan devices, which is defined in the BSD file, and subsequently they are executed as the boundary scan cells to drive and sense the test signals in and out the sensor chips by the master chip as follows; [19].

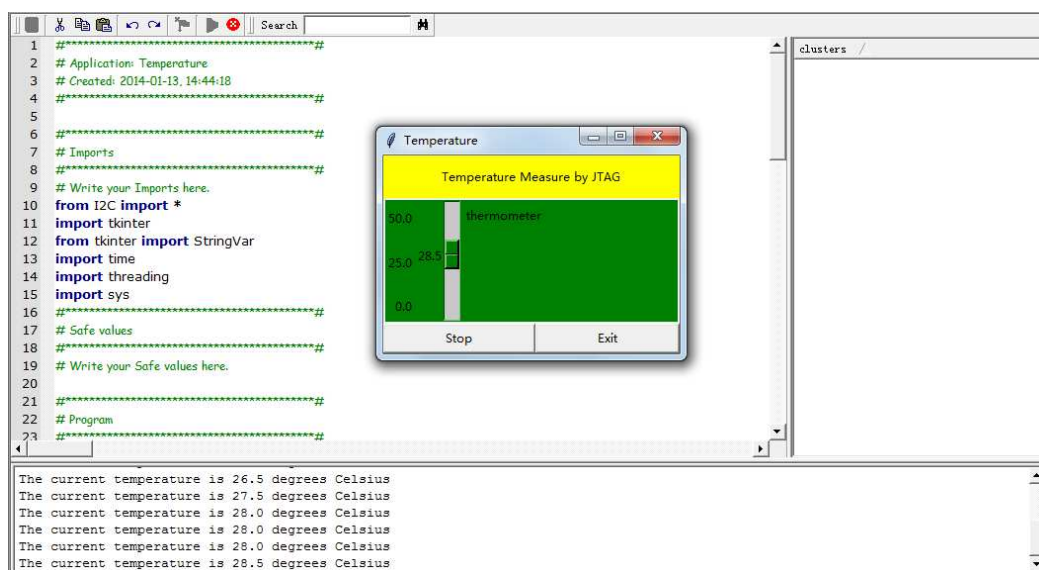


Figure 2. Microchip TCN75A I2C based temperature sensor measured by JFT.

2.2. Pressure Sensor Measurement by JFT

Still in the boundary scan test (BST) [8], the pressure sensor application is to use the functionality of the I2C-bus-based control of the chip from the Atmel Q touch controller

device, to check if the subject is encountering a “force” from a foreign object, its FCT based functional programming listed as shown in Figures 3, 4.

```

5
6 *****#
7 # Imports
8 *****#
9 # Write your Imports here.
10 from jft import *
11 from I2C import *
12 from time import *

```

Figure 3. Example of Import Script files from JFT, I2C in time domain.

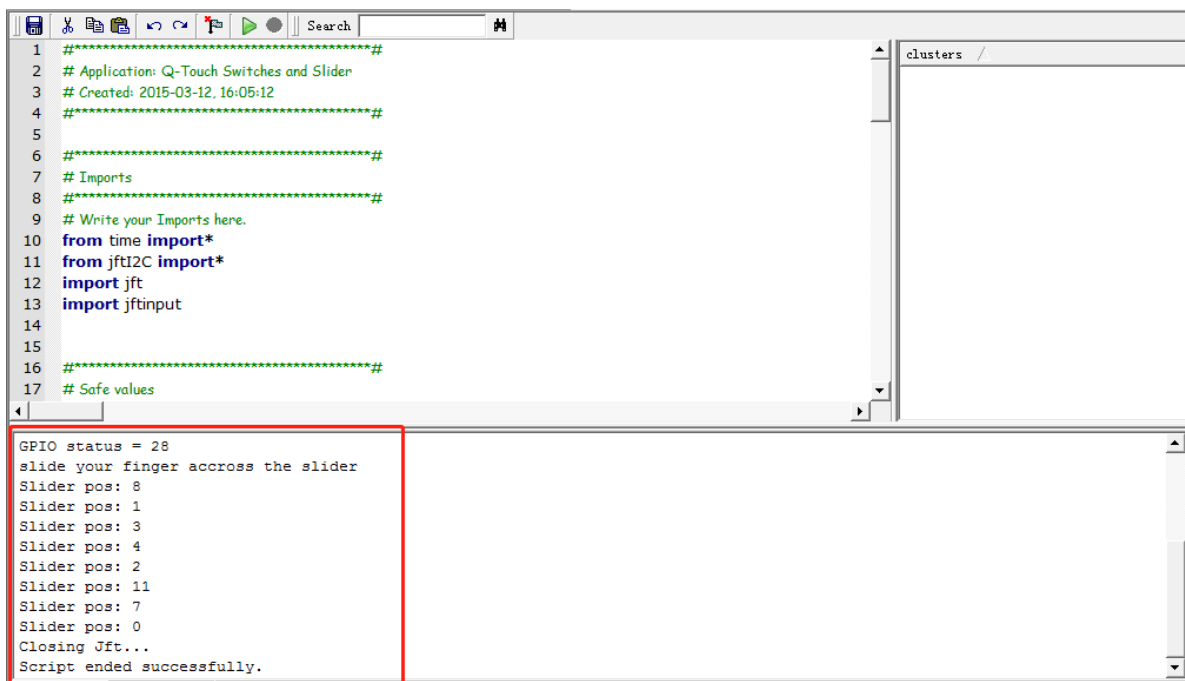


Figure 4. Pressure sensor-based Atmel Q touch controller undergoing “movement test”.

2.3. Dynamic Sensor Measurement by JFT

This device from NXP is an Orientation/Motion Detection Sensor, it's based on a “3-Axis accelerometer” with digital output via the I2C bus, of which is a capacitive MEMS sensor featuring a low pass filter, compensation for 0-g

offset and gain errors, and is converted into 16-bit digital values at a user configuration within in JFT. The device can be used for sensor data changes, object orientation, and gesture detection during an “interrupt”, effects taking place at the pin (INT) is designed in the schematics as depicted in Figure 5.

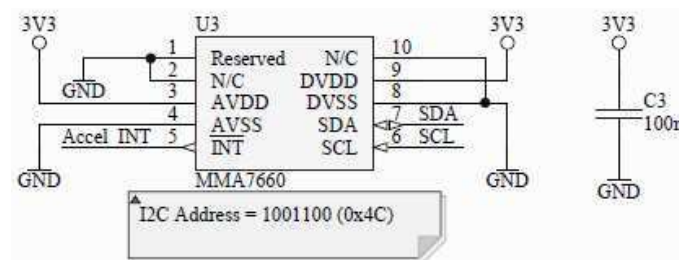


Figure 5. Schematics of Accelerator detection by JFT.

The Renesas' R-Car platform addresses a wide range of automotive applications such as Automated Driving or ADAS, connected Gateway, In-Vehicle Infotainment, Cockpit and Dashboard. It is designed for the next generation of automotive computing, and cars [15] have come to so called "derivable computers" by Tesla as well, countless LAN-connected MCUs, etc. acquiring the data from actuators and sensors to create autonomous driving, this advanced navigation systems and infotainment systems driving more pleasant by using video recognition based Artificial Intelligence (AI) technologies in supporting the ISO2626 safety standard up to ASIL D level. The proposed emulation according to the motion from a Freescale chip as per MMA7660FCT, its a I2C based "Accelerometer". This motion detection sensor is designed with MEMS based

capacitive sensing so called g-cell. It is a signal conditioning ASIC chip, sensing element of the g-cell like a mechanical structure, formed from the semiconductor materials (polysilicon) through the masking and etching processes; theoretically the sensor can be modeled as a movable beam that moves between two mechanically fixed beams. When two gaps are formed between the movable beams, the first stationary beam and the second movable beam are sensed by the ASIC using the switched capacitor techniques to measure the g-cell capacitors, and thus to extract the "acceleration data" from the difference between the two capacitors. The ASIC also performs signal conditioning and filtering, by providing a digital output that is proportional to the acceleration in relation to the JFT programmed scales, of which is depicted in Figure 6.

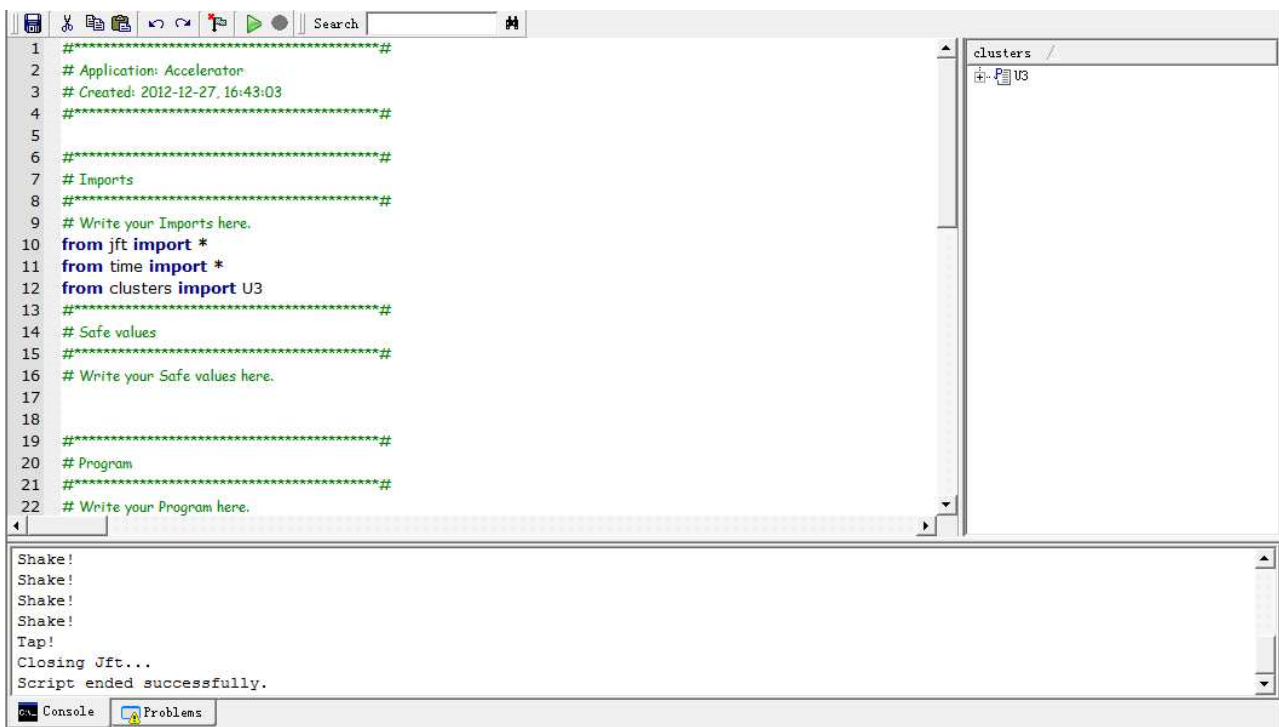


Figure 6. JFT program for a Freescale MMA7660FCT I2C based vibration test.

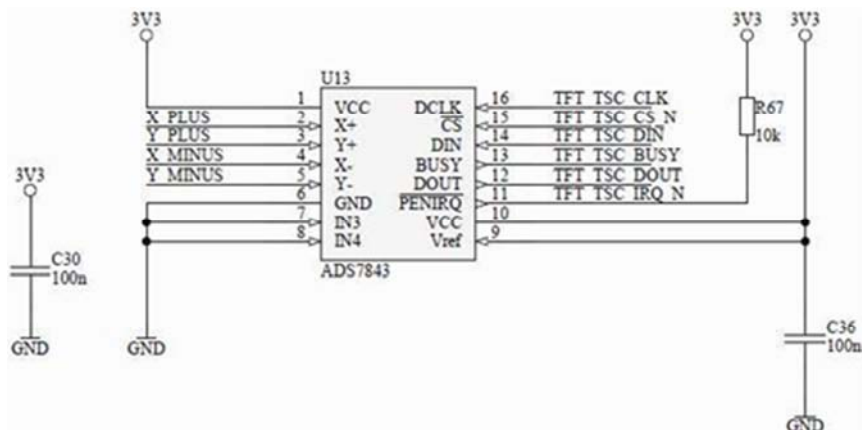


Figure 7. JFT import accelerator signals from the schematics for a TFT screen.

2.4. Evidence of Touch Contact on a LCD

In the application of touch-contacting, this approach uses the ADS7843 device which is a 12-bit sampling Analog-to-Digital Converter (ADC) device along with a synchronous serial interface, both low on resistance, switching them for driving the touch-on during interruption. This is patented

device including a shutdown mode which only needs power dissipation under $0.5\mu\text{W}$ during contact. An onboard switch makes this sensor ideally to experiment the sensory theory, and its associated instrumentation, the schematics in Figure 7 illustrates the TFT at work, and expressed in Figure 8.

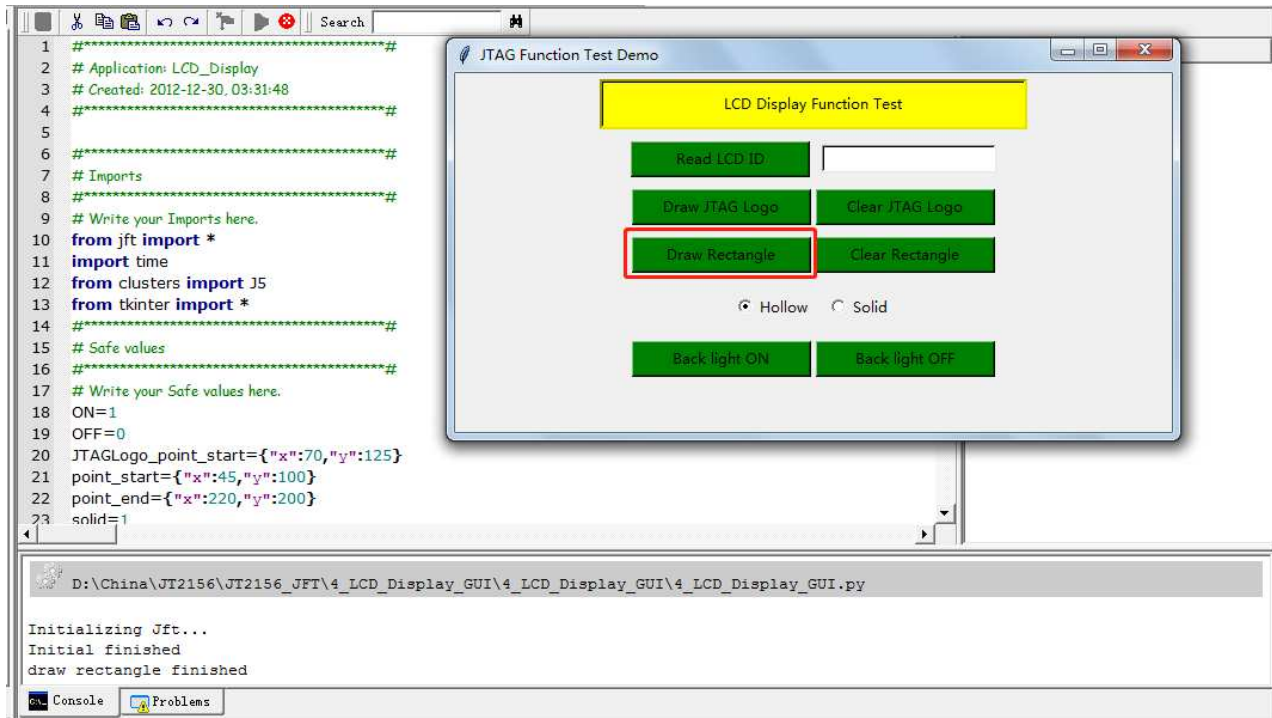


Figure 8. JFT imports expression test program from the design in Figure 7.

2.5. Results of Python Based FCT in a Sequential Test Procedure

JFT uses Python as the high-level language, and the Python is a well-known object-oriented interpreter-based

language, and is a freeware from www.python.org, which is installed on the JTAG's Provision Development System, and their results of the above test procedures are depicted in the following Figure 9.

The image shows a 'SEQUENCE REPORT' window. It contains a table with columns: Seq, Nr., Type, Operation, Item, Show, Status, and Time. The table lists 9 test items, all of which passed. The items include infrastructure tests, temperature GUI tests, touch switch slider tests, LCD display tests, RGB LED GUI tests, switch GUI tests, Ethernet internal loopback tests, Ethernet ID tests, and a crystal check test.

Seq	Nr.	Type	Operation	Item	Show	Status	Time
1	1	** Command Output Report: 1_Infra-Output-11_19_2019-21h_26m_47s_87ms.html **					
1	1	Infrastructure Only Test	Test Infrastructure	1_Infra	Status	Passed	21:26:50
1	2	** Command Output Report: 2_Temperature_GUI-Output-11_19_2019-21h_27m_52s_382ms.html **					
1	2	JTAG Functional Test	Execute Test	2_Temperature_GUI	Status	Passed	21:27:53
1	3	** Command Output Report: 3_Q-TouchSwitch_Slider_10-Output-11_19_2019-21h_27m_59s_297ms.html **					
1	3	JTAG Functional Test	Execute Test	3_Q-TouchSwitch_Slider_10	Status	Passed	21:28:01
1	4	** Command Output Report: 4_LCD_Display-Output-11_19_2019-21h_28m_29s_93ms.html **					
1	4	JTAG Functional Test	Execute Test	4_LCD_Display	Status	Passed	21:28:31
1	5	** Command Output Report: 5_RGB_LED_GUI-Output-11_19_2019-21h_31m_59s_31ms.html **					
1	5	JTAG Functional Test	Execute Test	5_RGB_LED_GUI	Status	Passed	21:32:00
1	6	** Command Output Report: 6_Switch_GUI-Output-11_19_2019-21h_32m_29s_969ms.html **					
1	6	JTAG Functional Test	Execute Test	6_Switch_GUI	Status	Passed	21:32:31
1	7	** Command Output Report: 8_Ethernet_Internal_Loopback-Output-11_19_2019-21h_32m_46s_587ms.html **					
1	7	JTAG Functional Test	Execute Test	8_Ethernet_Internal_Loopback	Status	Passed	21:32:48
1	8	** Command Output Report: 8_Ethernet_Id-Output-11_19_2019-21h_33m_01s_564ms.html **					
1	8	JTAG Functional Test	Execute Test	8_Ethernet_Id	Status	Passed	21:33:03
1	9	** Command Output Report: 9_Crystal_Check-Output-11_19_2019-21h_33m_04s_541ms.html **					

Figure 9. Test sequence reported by the hyperlinks of XML in execution of JFT programs.

3. Tracing on an SoC Based Board via the MODBUS-RTU Protocols: Case Study B)

An NXP [18] based automotive circuit board is proposed for this case study B), the iMX 6 Dual/Quad automotive processors represent an SoC-based processing power, with a high degree of functional integration. These processors were developed to meet the requirements in automotive infotainment, telematics, HMI, and display-based clusters like the one in Figure 10.

This block diagram of unit under case study is depicted in Figure 11, red marked terminals on both figures led to develop a series of software applications based on the open architecture from the MODBUS protocol. It is used to accept database in complex proceedings, due to its easy use and portable in file exchanges, and that is chosen for this study.



Figure 10. System under study based on an NXP designed automotive board.

3.1. XML Import JFT Tests

Reading and Writing JFT Files in Python, Extensible Markup Language, commonly known as XML, is a language that is designed specifically to be interpreted by both human knowledge and computer in engineering. The language defines a set of rules used to encode a database in a specific format. In this study, practical methods are introduced to experiment reading and writing *PHY through XML link. The JTAG based test sequence so-called AEX Manager is a fame work to process the importing of test data from an.XML file, and export its logical components, known as “Parsing”, illustrated as follows;

```
#####
# Application: JFT_TEST
# Created:
#####
# Imports
#####
# Write your Imports here.
from time import *
import subprocess
import os
from xml.dom import *PHY
#####
```

```
# Safe values
#####
# Write your Safe values here.
#####
# Program
#####
#.process1=subprocess.Popen("FunctionTest.exe")
print (process1.communicate()[0])
#####
# Output information into AEX Report
#####
# File_text1=open(r'result.ini')
int_result=int(file_text1.readline())
file_text1.close()
file_text2=open(r'fct_log.txt')
while 1:
log=file_text2.readline()
if not log:
break
print (log, end="")
file_text2.close
exit(int_result)
```

MODBUS was given by Schneider’s IP (Modbus is an open serial protocol derived from master/slave architecture, originally was by Modicon and now is Schneider Electric), it is based on the RS-485 bus protocols, preload-ed with two.BAS drivers in this case, to develop traceability used to run off tracking and tracing. A traceability protocol “gateway” is proposed in this paper to provide data exchange between XML and Modbus-RTU. [13] This is used to communicate between devices through the serial interface of RS-485 linking with the Modbus/RTU. This paper proposes a strategy to represent the data, object file (compiled file) in an XML format. This XML file will be sent to other devices through the serial interface of RS-485 using Modbus based Remote Terminal Units (RTU). The process “traceability” word is used to store all the different steps of the procedure in a VB.NET as follow:

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Parity				Nb C2 C2 x				Nb C1				C1 x x Tc			

Each X correspond to 1 Bit

Tc: (1 bit) Board test bit

x: (2 bits) reserved

C1: (1 bit) Burn In test

Nb C1: (3 bits) Number of Burn In Tests

x: (1 bit) reserved

C2: (1 bit) Unit final test

Nb C2: (3 bits) Number of unit final test

Parity: (4 bits) See below “Explanation of Parity”

Proposed WORD in 16-bits as follows:

X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1 0 0 1				0 1 0 0				1 0 1 1				1 0 0 1			
Parity				Nb C2 C2 X				Nb C1				C1 x x Tc			

Where:

Tc=1: Board test OK at least 1 time
 C1=1: Burn In test OK the last time
 Nb C1=3: The inverter was tested 3 times
 C2=0: The inverter was not OK
 Nb C2=2: The inverter was tested 2 times

3.2. Party Check

The above stated WORD in party is corresponding to its fourth byte; for an example, PPPP 0101 0000 0000 where the Ps corresponding to the word's parity; such as ABCD 0101 1111 0001; In case of computation mode, complemented exclusive OR of the first three bytes, to be experimented as follows:

A=complement of (0+1+0)=0
 B=complement of (1+1+0)=1
 C=complement of (0+1+0)=0
 D=complement of (1+1+1)=0

The parity is represented as "0100", this computation is applied to validate the consistency of a word used in the process which is reading in a consistent word: 0100 0101 1111 0001, i.e., the process of traceability in words is used to store all the different steps as follows:

X X X X X X X X X X X X X X X X
1 0 0 1 0 1 0 0 1 0 1 1 1 0 0 1
Parity Nb C2 C2 X Nb C1 C1 x x Tc

Where:

Each X correspond to 1 Bit

Tc: (1 bit) Board test bit

x: (2 bits) Reserved

C1: (1 bit) FCT test

Nb C1: (3 bits) Number of FCT Tests

x: (1 bit) reserved

C2: (1 bit) Unit final test

Nb C2: (3 bits) Number of unit final test

Assuming, the definition of parameters, i.e., P112, at its communication address I-254, when the system using RS-485 protocols to perform serial communication under supervision of MODBUS, each driver associated address is unique, and not repeatable, unless using "0" for broadcasting. With regarding RTU transmission, every 8-bit data is formatted in 2x4 bits per 16-bits word in carriage return check (CRC), they are formatted as follows:

RTU format: START (P112)

ADR 8-bit address (I-254)

CMD 8-bits instruction

DATA (n-1)..... DATA 0 CRC CHK Low CRC CHK High END

end:

3.3. Case Study of the NXP 6Q Board Under Testing

The i.MAX Dual/6 Quad processors represent the integrated multimedia processors. These processors are part of a growing family of multimedia that is focused, featuring an advanced implementation of the quad Arm® Cortex®-A9 core, and which operates at speeds up to 800 MHz. They include 2D and 3D graphics processors, 1080p video processing, and integrated power management. [9-11].

Each processor consists of a 64-bit DR3/DDR3L/LPDDR2-800 memory interface and several other interfaces for connecting peripherals, such as WLAN, Bluetooth®, GPS, hard drive, displays, and camera sensors, the block diagram is shown in Figure 11 below. This exercise has prepared to use "transferability" to validate the above-mentioned ADAS related algorithms in FCT, and to define the systems-related sensors such as camera rigging to produce images in the data-set in a test sequence, to mimic the off-road conditions related to control and command systems as to a human behavior model.

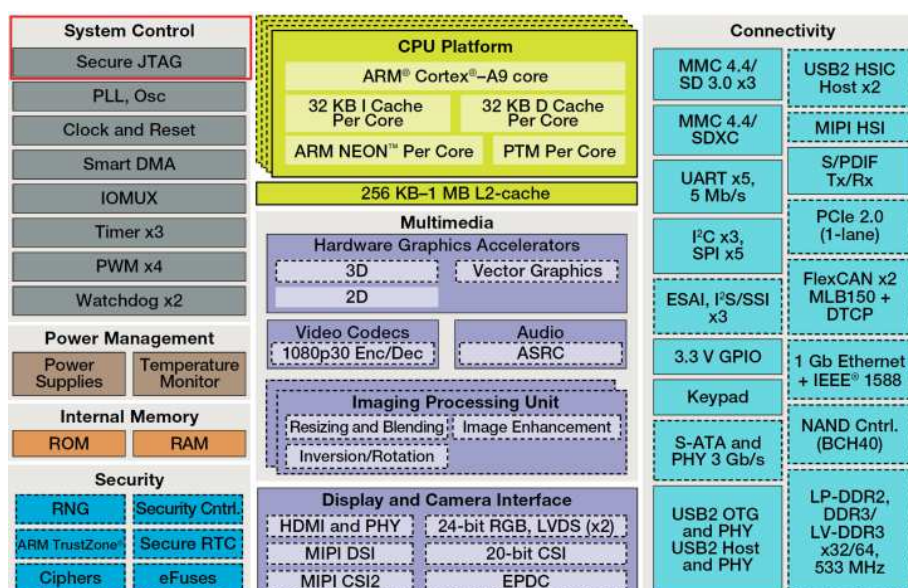


Figure 11. The architecture of the NXP 6Q board under "traceability" study.

3.4. Generation of .BAS drivers (VB.net) for MODBUS

This design approach utilizes the Modbus protocol stacks for C#, VB.net, Java, and for C++ while they are maintaining similar functionality. The C++ editions of the Modbus, aren't much different in VB.net, drivers of which written as follows:

```
Attribute
VB_Name="modProcessTraceabilityStatusWords"
Public Type TraceAndStatusWords
    int_step_count As Integer
    str_grid_display_flag () As String
    str_execute_type () As String
    str_execute_description () As String
    str_execute_param () As String
    dbl_rb_traceability_word As Double
    dbl_new_traceability_word As Double
    dbl_pass_status_word As Double
    dbl_fail_tatus_word As Double
    str_log As String
    str_status As String
    str_time As String
End Type
```

Besides, XML is a general-purpose tag-based language, used to transfer and store data across applications. XML is a platform-independent language so that the information formatted in XML can be used in any other platform, i.e., the VB.net format, thus chosen for traceability study, and that provides a sequence of reading, writing and other operations in XML formatted files as well.

4. Results of Traceability in XML format

When designing the program, some data with specific variability can be edited in XML file also, to facilitate the user modifications as follows:

```
<?Xml version="1. 0" encoding="UTF-8"?>
<?MyCustomNameHere attribute="val1"
<foo: bar xmlns:foo="http://tempuri.
PCB/foo">
<IC maker>
<description>
<part number>
<data sheet 1>
<spec.>
</applications>
</JTAG>
</BSDL>
```

```
</foo: bar>
```

From hardware point of view, the IC may be a MCU or CPU having data with system variability that can be edited in XML files in order to facilitate later changes, and the XML file is easy to use in any application as a generic file, illustrated as follows;

```
using System;
\Using System. Collections. Generic;
\Using System.Linq;
\Using System.Text;
\Using System.IO;
\Using System.Xml;
XmlProcess category
```

In addition, this method is proposed to write the test results via the VB-XML path during the storage process and to load the test results into the memory instead of logging test files on a disk, by using the following strings to save them in EEPROM;

```
private void GetChartData (string OC_Ttl1, string
OC_Ttl2, string OC_OL31)
{
    OC_Ttl_1=OC_Ttl1;
    OC_Ttl_2=OC_Ttl2;
    OC_OL3_1=OC_OL31;
    //Output xml
    Dataset or Dataset=new Dataset ();
    orgDataSet.ReadXml(cmd_Org.ExecuteXmlReader(),
    XmlReadMode.Auto);
    orgDataSet.WriteXml("InputXMLFiles/" + OC_OL3_1.
    Replace (" ",
    "_"). Replace ("/", "-") + ".xml");
}
```

As stated, to output the traceability files into an memory instead of writing the file to the disk, that is to capture the test results through the following steps from 1 to 7, and to report them in an EEPROM, depicted in Figures 12 and 13;

1. Finish calibration
2. Test of temperature
3. Test of pressure
4. Test of vibration
5. Test of LCD
4. Read traceability words
5. Write traceability words
6. Read status words
7. Save value EEPROM

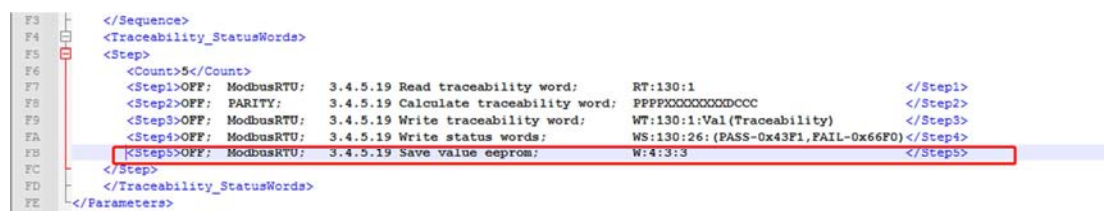
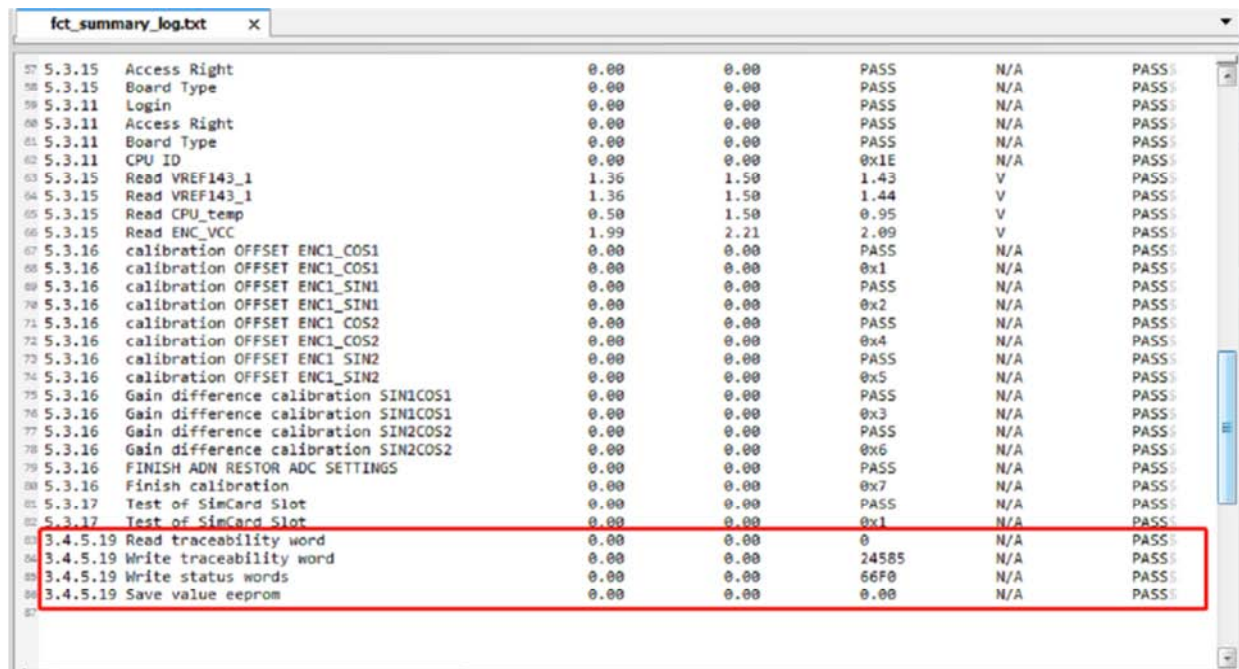


Figure 12. Step 6, Statue of Traceability.



Line	Test Step	Value 1	Value 2	Status 1	Status 2	Status 3
57	5.3.15 Access Right	0.00	0.00	PASS	N/A	PASS
58	5.3.15 Board Type	0.00	0.00	PASS	N/A	PASS
59	5.3.11 Login	0.00	0.00	PASS	N/A	PASS
60	5.3.11 Access Right	0.00	0.00	PASS	N/A	PASS
61	5.3.11 Board Type	0.00	0.00	PASS	N/A	PASS
62	5.3.11 CPU ID	0.00	0.00	0x1E	N/A	PASS
63	5.3.15 Read VREF143_1	1.36	1.50	1.43	V	PASS
64	5.3.15 Read VREF143_1	1.36	1.50	1.44	V	PASS
65	5.3.15 Read CPU_temp	0.50	1.50	0.95	V	PASS
66	5.3.15 Read ENC_VCC	1.99	2.21	2.09	V	PASS
67	5.3.16 calibration OFFSET ENC1_COS1	0.00	0.00	PASS	N/A	PASS
68	5.3.16 calibration OFFSET ENC1_COS1	0.00	0.00	0x1	N/A	PASS
69	5.3.16 calibration OFFSET ENC1_SIN1	0.00	0.00	PASS	N/A	PASS
70	5.3.16 calibration OFFSET ENC1_SIN1	0.00	0.00	0x2	N/A	PASS
71	5.3.16 calibration OFFSET ENC1_COS2	0.00	0.00	PASS	N/A	PASS
72	5.3.16 calibration OFFSET ENC1_COS2	0.00	0.00	0x4	N/A	PASS
73	5.3.16 calibration OFFSET ENC1_SIN2	0.00	0.00	PASS	N/A	PASS
74	5.3.16 calibration OFFSET ENC1_SIN2	0.00	0.00	0x5	N/A	PASS
75	5.3.16 Gain difference calibration SIN1COS1	0.00	0.00	PASS	N/A	PASS
76	5.3.16 Gain difference calibration SIN1COS1	0.00	0.00	0x3	N/A	PASS
77	5.3.16 Gain difference calibration SIN2COS2	0.00	0.00	PASS	N/A	PASS
78	5.3.16 Gain difference calibration SIN2COS2	0.00	0.00	0x6	N/A	PASS
79	5.3.16 FINISH ADN RESTOR ADC SETTINGS	0.00	0.00	PASS	N/A	PASS
80	5.3.16 Finish calibration	0.00	0.00	0x7	N/A	PASS
81	5.3.17 Test of SimCard Slot	0.00	0.00	PASS	N/A	PASS
82	5.3.17 Test of SimCard Slot	0.00	0.00	0x1	N/A	PASS
83	3.4.5.19 Read traceability word	0.00	0.00	0	N/A	PASS
84	3.4.5.19 Write traceability word	0.00	0.00	24585	N/A	PASS
85	3.4.5.19 Write status words	0.00	0.00	66F0	N/A	PASS
86	3.4.5.19 Save value eeprom	0.00	0.00	0.00	N/A	PASS

Figure 13. Step 7, Save value of test results in EEPROM.

5. Conclusion

In conclusion, this R&D work demonstrates using an MCU-based NXP chip to experiment the automotive sensory-related motions in the areas of vibration, pressure, temperature, and detection. They are accompanied by the advancement of JTAG based boundary-scan technology as per IEEE1149.1 specified functional test (FCT), written in Python language. Further, this proposal applies another NXP i.MX Dual/6 board with a built-in SoC chip, to validate the automotive electronics required traceability, through a string of MODBUS-based RS-485 protocols. This is made to data linking the Remote Terminal Units (RTU), written in the standard net.XML. This procedure is to execute the protocol in a 16-bit word, recording the performance of the devices under evaluation through the VB.net-based computer programming. In this study, the next step of work is planned to pursue a higher degree of execution, related to the area of the IEEE1149.7 based Signature Analysis (SI). The SI is a linear calculation as averse to the current vector-based boundary scan test, it gives validation of the automotive electronics during autonomous driving conditions. [14, 16]

Acknowledgement

The authors would like to acknowledge and thank those from the TamKang University as well as JTAG Technologies' president: Peter van den Eijnden, for supporting hardware, software and instrument besides technical assistance for our research works.

References

- [1] Wang Shun Shen Peter "Instrumentation of Twin-MCM based Mutual Test", published by "Microelectronics Journal", Vol. 114 on August 21, 2021, <http://doi.org.j.mejo.2021.105108>.
- [2] Wang Shun Shen Peter, IEEE/International Test Conference, Washington D.C. 2020, Nov. 6-10, published on <http://dx.doi.org/10.6180/>, "Switch Mode Interposer developed to self-test an MCM without Known Good Dice", preprint on https://easychair.org/publications/preprint_open.
- [3] Wang, Shun Shen Peter, Wu Tien-Yu, "Wafer Scale Rerouting Process in making Known Good Dies", accepted by Journal of "Nan Kai University of Technology", Taichung, Taiwan, ISSN-1991-492x.
- [4] Wang, Shun Shen Peter, "Method of making a High-Density Multi-Layer Wiring Board", U.S. Patent No. 6,026,564.
- [5] Wang Shun Shen Peter and Chee Wong, "Wafer Scale Burn-in Testing", U.S. Patent No. 6,121,065 assigned to IME, University of Singapore.
- [6] Wang Shun Shen Peter, "Method and Arrangement for analyzing manufacturing defects of Multi-Chip-Module without Known Good Dies", U.S. Patent Pending No. 17076496.
- [7] Wang, Shun Shen Peter, "Switch-Mode based interposer enabling Self-Testing of an MCM without Known-Good-Die" U.S. Patent Pending No. 17240956.
- [8] Harry Bleeker and Peter van den Eijnden: "Boundary Scan Test A Practical Approach", published book by Eluwer Academic, Netherlands @1993.
- [9] Data sheet: AN5375, The S32R27x is a 32-bit Power Architecture® based Microcontroller Unit (MCU) targeted for automotive applications.

- [10] NXP document: introduces a complete system power solution using FS84/FS85 and PF502x family such as PF5020, PF5023, and PF5024.
- [11] Data sheet NXP TEF810X, an automotive radar transceiver for all range radar applications, covering the full car radar frequency band from 76 GHz to 81 GHz.
- [12] Data sheet: NXP MCUs: strike the optimal performance-per-watt balance for hardware-accelerated, high-resolution RADAR systems designed for safer, smarter vehicle.
- [13] <https://www.rtautomation.com/technologies/modbus-rtu/>, RTU in automation.
- [14] <https://www.autonomousvehiclesymposium.com/detroit/en/>, autonomic vehicls safety.
- [15] <https://Eetimes.com/autonomous-vehicle-test-development>, autonomous vehicles test development.
- [16] <https://www.researchgate.net/publication/301272911>, Challenges_in_Autonomous_Vehicle_Testing_Validation.
- [17] <https://www.renesas.com/us/en/products/automotive-products/automotive-system-chips-socs>.
- [18] https://datasheet.lcsc.com/lcsc/1811091610_NXP-Semicon-MCIMX6Q5EYM10AD_C133182.pdf.
- [19] Peter Wang Retrieved from www.jtag.com, Oct. 10, 2021.