Automated Test Fixture for In-production Functional Testing of Electronic Devices

Milan Vukajlovic, Srdjan Tadic

***Abstract*— We present one practical solution for automated test fixture for electronic devices. Testing is done during production, and the emphasis is not only on electrical tests but on verifying end-to-end device functionality. The proposed solution is fully automated, it provides reliable failure detection and it is affordable and suitable even for small production runs (such as 100-1000 pieces per month). The solution is shown in form of a case-study based on the real production of the consumer device, which has been proven on the market. We present the complete system, including details of the device under test, designed test fixture, a C++ application and web application.**

***Index Terms*—Automation, testing, ATF, test procedures, DUT**

I. Introduction

TESTING is an important part of the software and hardware development process associated with significant planning and execution costs. The main purpose of testing is ensuring that produced hardware and software reliably operates according to initial specifications and within various predefined environmental and usage conditions, and just as important, it does not do what it is not supposed to do. According to some estimates, roughly 50 percent of the elapsed time and more than 50 percent of total development cost is associated with planning and executing testing activities [1], [2]. The testing costs for hardware and safety-of-life critical systems are often much higher [3]. Various test strategies for software and hardware testing are suggested to match overall development approach [4], [5]. The inadequate testing can lead to catastrophic consequences to project/product success and sometimes it can endanger overall company operation. According to a recent report [6], the total value of automated test equipment market is expected to surpass $4.59 billion by 2025.

Testing takes a significant portion of the development effort, but it is also a very important part of the production process. Automated fault diagnosis becomes an ongoing demand for new technologies [7]. The objective of presented work is to optimize this effort and automate most of the process. Reliable in-production testing ensures functionality of the end-product and prevents potential multiplicative damage in after-sales and it affecting company reputation, supports optimization of the production time itself, and reduces the probability of human-caused errors. It also enables that all relevant in-production information are stored and post-analyzed if devices envisage sudden failures in exploitation.

We will outline the suggested testing process with the main focus on hardware functional verification and optimization of critical and labor-intensive segments in manufacturing testing. Testing will be depicted through a detailed case study of one consumer electronics device which reliability has already been market proven. The case study will show details of custom designed automated testing fixture (ATF) for the automatic, labor-saving and reliable acquisition of test data, which also reduces required test duration and simplifies the production process. According to our experience, this approach and suggested concept of ATF should be suitable for testing in many other use-cases, especially for medium-volume production (up to 100 thousands of pieces per year). Such production volume is also quite often in ramp-up phase of mass-market products. Hereby given approach does not include some product-specific testing, such as operation in different environments simulated by temperature/humidity chambers, shock tests or hypothetical calibration procedures.

The paper is divided into six sections. After an introduction, the second section is a review of state of the art in test automation. The third section briefly describes the device under test (DUT) and the ATF-related considerations made early at the design phase. The fourth section describes ATF. The fifth section describes the test system, briefly describes performed tests and PC and Web applications. In the end, we summarize achievements.

II. State of the Art and Problem Statement

Common testing of the assembled PCBs starts with the X-ray scanning or (at rare cases) visual inspections right after the component soldering. Following this, the "basic PCB tests" include:

- "Short circuit" testing between the power lines and the ground,
- Device power-up and verifying voltage levels at the predefined test-points.

This is followed by system "functional tests" including:

- Firmware download,
- Validating peripherals (sensor readings, establishing communication, networking, quality of TX/RX signals, etc.)

Tests are usually concluded by storing obtained results together with the identification data of each PCB, as a permanent record for post-analysis and further improvement of quality and optimization of associated processes.

Milan Vukajlovic is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: milanvukajlovic86@gmail.com).

Srdjan Tadic is with Bitgear Wireless Design Services, Stevana Markovica 8, 11080 Belgrade, Serbia (e-mail: srdjan.tadic@gmail.com).

In prototyping phase, the role of the test is verification of design concept. This is usually done manually. Once the design becomes more solid, there is a need for finding boundaries and weaknesses. At this stage, depending on the device complexity, some level of automation is recommended, to ensure repeatability of test results. At the manufacturing stage, repeatability and reliability become a paramount concern.

One of the main considerations of the test system is level of automation in control of the test process. On one side is the manual control where human operator should do all testing and records. At volume production, such a repetitive task needs to be controlled automatically. Automated control takes care of signal switching, measurements, recording, and sometimes some deeper analysis of the results for pass/fail determination. Table 1 shows the comparison of various control methods. Depending on exact product application, its complexity and development stage, many factors will influence which could be the most suitable one.

TABLE I
TEST SYSTEM CONTROL METHODS

	Manual	Common ATF	Custom ATF
Instrument cost	Low	High	Low
Development time	Low	High	High
Development cost	Low	High	Medium
Flexibility	High	Low	Low
Throughput	Low	High	High
Operator experience	Very high	Low	Low
Ease of instrument reuse	High	Medium	Low
Potential for human error	High	Low	Low

The common approach for test system control automation is the use of one of two major instrument forms: “rack-and-stack” and “Card cage” [8]. “Rack-and-stack” instruments are standalone devices, typically controlled via external PCs. “Card cage instruments” are modular test instruments based on plug-in cards. These cards are inserted to “Card cage” and controlled by external PC or an embedded controller (another plug-in card). In both of these forms, there is a rich span of various instruments. Two leading and the most popular equipment vendors are Agilent and National Instruments. Both companies also provide user-friendly and widely-known development environments. So, there are many pros to go for this common approach such as flexibility and fully off-the-shelf equipment. Drawbacks are that this is attached to relatively high associated software licensing and hardware costs. At some cases, this may not be suitable for small companies.

We propose efficient and reliable, yet reasonably complex to build, automated test fixture made of low-cost components. While the concept is almost universal, we will illustrate it on an example of a medium-complex device typical for a nowadays IoT-era. This should make hereby shown approach more useful for the other embedded system developers.

III. DEVICE UNDER TEST

In this case study, DUT is a wearable device built to track motion and send data to Web application using cellular connectivity. The device has a microcontroller (MCU) for executing application firmware, GPRS (General Packet Radio Service) module for communication with a Web application, MEMS (Micro-electromechanical Systems) accelerometer, external flash memory, RGB (Red-Green-Blue) diode for indication and multifunctional button. The device is powered from the rechargeable Li-Po (Lithium-Polymer) battery and charging is done over the USB connector (Fig. 1).

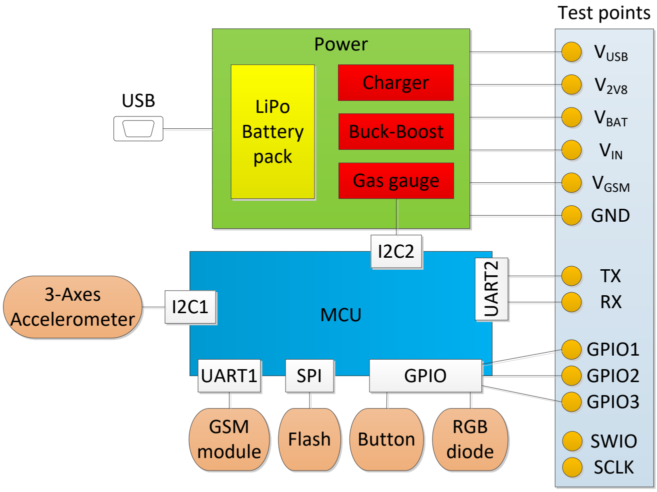

Fig. 1. DUT block diagram

The device was designed with careful consideration of in-manufacturing testing to simplify the design of required test-fixture hardware and overall verification process. All hardware test points are placed within the single PCB layer. Device shape secures that it can be mounted to ATF in only one way. There is a dedicated part of the application firmware for device self-test and communication with ATF. We have identified the need to verify the following test-points (Fig. 2):

- Voltage tests
  - V_{USB} – Voltage from USB connector
  - V_{IN} – Voltage before Buck-Boost
  - V_{BAT} – Battery voltage
  - V_{2V8} – Main board voltage
  - V_{GSM} – Voltage for GPRS module
- Charger
- NTC (Negative Temperature Coefficient) thermistor
- Button
- Flash memory
- 3-Axes accelerometer
- Gas gauge
- RGB diode
- Real Time Clock (RTC)
- GPRS module

Fig. 2. PCB ready for test with marked test points in top layer

IV. AUTOMATED TEST FIXTURE

Automated test fixture comprises of a custom designed and assembled PCB (Fig. 3) mounted within the commercial of-the-shelf mechanical fixture (Fig. 4). Mechanical test-fixture secures physical connection of DUTs and so called pogo pins ("bed of needles") which are carefully positioned to match test points on PCB and MCU programmer for programming devices. Test fixture communicates over the USB (via MCU) with a dedicated PC application and with DUTs over SPI2UART component. At hereby presented case, one of the design requirements was to secure device timely testing in case of monthly production volumes of 5000 devices. Since asingle test can last up to 3 minutes (due to slow GPRS communication), we have designed ATF with 6 test slots.

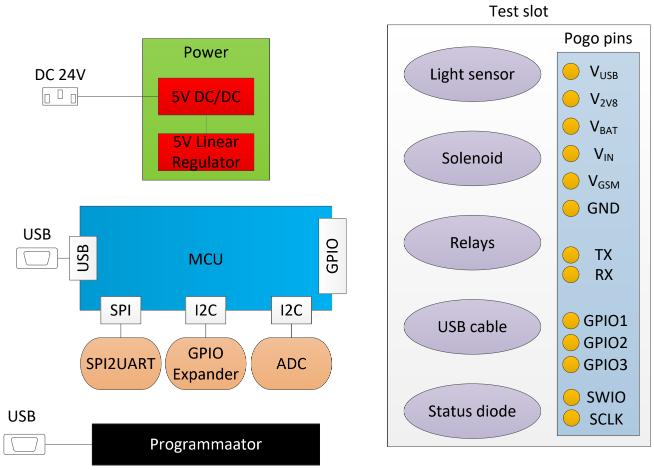

Fig. 3. Block diagram of ATF

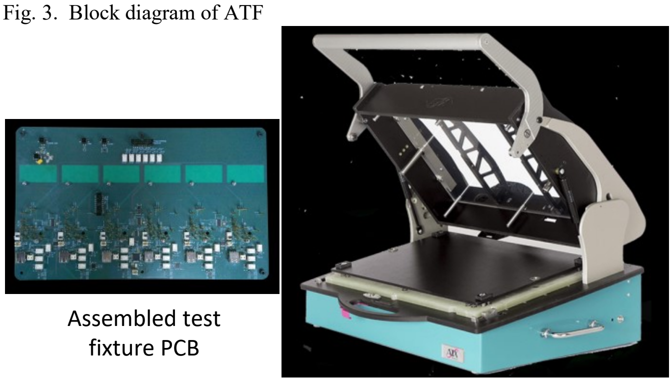

Fig. 4. Mechanical stand-alone fixture

V. TEST SYSTEM

A. Architecture

The architecture of the proposed test system is shown on Fig. 5. Board for testing is positioned in a dedicated mechanical slot of ATF and connected to it by USB. By using USB cable instead of test points, at the same time we are verifying soldering of the USB connector. Current on the USB port is limited until the ATF verifies proper range of all on-board voltages, meaning that there are no short-circuits. After success of these "basic PCB tests", the PC application and ATF automatically upload application firmware to DUTs and turn on the devices. After power-on, the devices check for presence of ATF by reading predefined pin. If this check is correct, devices will initiate self-tests.

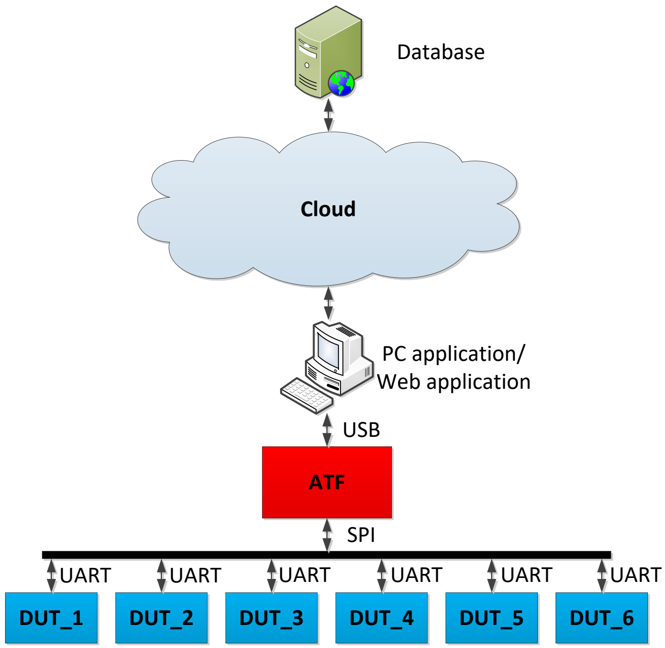

Fig. 5. Block diagram of test system

For this product, it was required to combine ATF and device self-test because of importance to test all peripherals from the device controller firmware. At the same time, the ATF tests voltage levels, programs the firmware, and communicates with the PC application. To finalize the test, ATF sends results to PC application that further uploads these to the cloud. User can initiate and monitor test over a simple GUI on a PC. Additionally, next to each mounted DUT on ATF there is a Red-Green light diode to display the final test pass/fail conclusion and reduce potential mistakes when devices are taken out from the ATF. A few seconds after the test is finished detailed test report is also available on a web application.

B. Brief explanation of tests

At the beginning the operator places devices in mechanical slots and initiates tests ("Start" button) via the PC application.

Voltage-level test is the first one and this is common practice. ATF provides power for all devices through USB connector and measures 4 voltages V_{USB}, V_{IN}, V_{BAT}, V_{2V8}. If these are within the specified bounds, PC application will upload the application firmware to all devices that passed the voltage test.

The second phase is the DUT's self-test. As a part of the initialization, the application firmware reads dedicated pins to detect presence of ATF and sends message over UART using custom designed communication protocol to ATF controller. ATF responds with ACK and configuration data. In this stage DUT performs different self-tests.

Accelerometer is tested by reading WHO_AM_I register and 10 consecutive sensor samples. If value of WHO_AM_I register matches the expected value and if measured acceleration is close to value of gravity force, device passes this test.

Charger test is done by reading charging status pin.

NTC thermistor is tested by reading value of the voltage between NTC thermistor and fixed resistor. Measure voltage should be within predefined bounds.

Flash memory is tested in steps: by erasing sectors, writing predefined data to sectors, reading and verifying data from sectors, and at the end erasing sectors and verifying erase.

Gas gauge is tested by writing predefined value to the configuration register and verifying this operation by reading the same register.

Real Time Clock is tested by comparing elapsed time from the RTC with elapsed time from the main clock.

Button is tested with support from the ATF. The device first requests from the ATF to press the button and checks the state. Then the device requests from the ATF to release the button and checks the state. If both states are as expected test is passed successfully.

RGB diode is tested by reading voltages from RGB light sensor which converts light to "red voltage", "blue voltage" and "green voltage". The device requests all 8 combinations. At each state, different combination of LEDs is ON and OFF. At the end, DUT calculates equivalent values for each of the states and compares them versus the expected values.

GPRS test is divided in 4 states because of complexity. The first state test communication lines by reading IMEI number. The second state tests SIM chip presence by reading CCID number. The third step verifies is the module able to register to any available cellular network and at the end test activates the GPRS context and performs PING command to the test server. If the server sends response, the test is declared as passed. In meanwhile, the PC application displays status of the test server, so in case of GPRS failure, the operator can check if test server was available during the test.

At the end, the device sends self-test results to the ATF and signal to test the voltage on the GPRS module V_{GSM}. When the voltage is measured the ATF sends complete test report for each device to the PC application. All test results are stored on the PC and uploaded to the cloud for easier integration with the device provisioning software and detecting potential issues with some specific batch of devices.

C. PC application and Web application

PC application has a simple but user-friendly interface to control the test and show results. Only two buttons are there for control and indicate tests pass/fail. The pop-up notification window appears to notify operator when tests are finished and all results uploaded.

Web application is connected with the PC application and provides overall production overview and history for each unit. Web application can be easily linked into stock applications.

VI. Conclusion

In this paper is presented realization of one in-production ATF. Cost of all required hardware components is quite affordable (less than 5 thousand EUR) considering that this setup guarantees quick and automatic verification of every single device in serial production. This solution could be cost-effective even for smaller production series, e.g. below 1000 devices per month.

Acknowledgment

This work is supported by company Bitgear Wireless Design Services.

References

[1] G. J. Myers, T. Badgett, T. M. Thomas, *The Art Of Software Testing*, 2nd ed, Hoboken, New Jersey, John Wiley & Sons, 2004.

[2] C. Tursky, R. Gordon, S. Cowie, *Test-System Design, A Systematic Approach*, University of Michigan, Prentice Hall, 2001.

[3] A. Hartman, "Software and Hardware Testing Using Combinatorial Covering Suites", in *Graph Theory, Combinatorics and Algorithms*, Boston, MA, Springer, 2005, pp 237-266

[4] P. M. Jacob, M. Prasanna, "A Comparative analysis on Black Box Testing Strategies", International Conference on Information Science (ICIS), Kochi, India, 2016.

[5] B. Beizer, *Black Box Testing*, New York: John Wiley & Sons, Inc., 1995.

[6] Grand View Research, Market Research Report, Oct 2017

[7] N. Paunovic, J. Kovacevic, I. Resetar , "A Methodology for Testing Complex Professional Electronic Systems", Serbian Journal Of Electrical Engineering, Vol 9, No. 1, pp 71-80, Feb 2012.

[8] Designing Automated Test Systems, *A Practical Guide to Software-Defined Test Engineering*, National Instruments

[9] http://www.atx-hardware.de