

Testing Facilities for a Solar Tracking device using Boundary Scan Test Strategies

Raul Rotar

Department of Automation and Computer Science, Bulevardul Vasile Pârvan, Nr. 2, 300223 Timișoara, jud. Timiș, România
rotar_raul@hotmail.com

Abstract—The present paper focuses on establishing relevant problems which arise from the field of Design for Testability, more precisely seeking adequate testing facilities for a Solar Tracking device. The Arduino Mega 2560 that controls the automated system, supports JTAG debugging, which gives access to using Boundary Scan strategies for testing purposes. Furthermore, after ensuring the functionality of the Arduino AtMega 2560 chip, the work will proceed with the actual testing of the automated solar panel.

Keywords—JTAG Protocol, Boundary Scan, IEEE 1149.1 standard, Linux Kali, Solar tracking, Stepper motors, Solar cells, Cast-Shadow principle, Optocoupler, Arduino, Sensors, Data acquisition.

I. INTRODUCTION

In a world saturated with emerging technology, design for testability has become a major concern for the manufacturing of PCB boards.

Classical electrical test methods of an electric board are the in-circuit test and the functional test. The in-circuit test consists of the use of a fixture adapter through which most components on a populated electronic board (PCB) are accessed. Access is made in the electrical nodes of the PCB, on test pads, testing each component separately. The functional test allows to analyze the parameters of the electronic board functionality by testing the parameters of the input/output signals or by embedding the electronic board in a simulator of the final assembly [8].

Fundamentally, the in-circuit test is based on the possibility of physical access to the electrical nodes of the PCB. Since the mid-1970's, the structural testing of loaded circuit boards has relied very heavily on the use of the so-called in-circuit bed-of-nails technique. However, SMD technology has had a serious impact on the ability to accurately place test pads. SMD devices and multi-layer PCBs have made physical access to electrical nodes extremely limited. The internal structure of the IC can be very complicated, requiring millions of test vectors for a full test. Also, the miniaturization of the electronic boards made certain SMD components have very close pins, and access to these pins for testing was difficult, the useful space being less than

0.6 mm. In other words, the ability to physically probe onto the board with a bed-of-nails system was going away: physical access has become a problem.

To overcome this issue, in the mid-1980's, a group of test engineers examined the possibility of testing electronic boards with limited access to electrical nodes. The proposed solution was based on access to the integrated circuit pins using a shift register internally connected to all the pins of the component. The technology was called "Boundary Scan". The organization that developed the Boundary Scan standard - to which all component manufacturers have adhered is called Joint Test Action Group (JTAG). This technology has been internationally standardized and received the IEEE1149.1-1990 code, first published in 1990. Today, it is known as Boundary Scan or JTAG [7].

The Renewable Energy industry becomes more and more oriented to automation solutions in order to improve resource gathering such as sunlight, wind, rain tides, waves, and geothermal heat. In the case of Solar tracking devices the main goal is to follow the sun's trajectory and ensure that the solar panel is positioned for maximum exposure to sunlight. The companies that sell solar trackers are claiming an increase of 25-45% in power output compared to fixed-tilt solar systems (e.g. rooftop installations). This justifies why solar trackers were used in more than 85 % of all commercial photovoltaic (PV) installations larger than 1MW between 2009 and 2012.

In the context of the modern era, the use of SoC's in automated projects is almost unavoidable. Arduino boards are flexible programming platforms offering a friendly IDE with

C/C++ language support. However, in unfortunate cases, these chips can suffer from manufacturing errors. The failure of such a device may generate the malfunction of the automation. Therefore, the idea of including of testing facilities in such scenarios is not only necessary but also vital for the proper functioning of the whole system.

This paper aims to promote clean energy, the reliability of the stepper motors and the efficiency of the Boundary Scan method used in the testing phase of the Arduino chip.

II. STATE OF THE ART

The printed circuit board (PCB) testing represents a vital step of the production cycle of microelectronic systems, as it is an important instrument to ensure the product quality level. The state of the art of PCB testing is a mixture of Boundary Scan (BS), optical and x-ray inspection, functional and in-circuit testing.

From a historical point of view, most PCB testing was achieved using so-called bed-of-nails in-circuit test (ICT) equipment, which offers the physical contact to any desired point of the PCB surface. This method proved extremely efficient in the case of low complexity PCB's. However, an average modern PCB, normally, contains several inaccessible internal layers inside and complex components upon the surface. Modern component packaging types, like e.g. various grid arrays (PGA or BGA), can hold up to several hundreds of hidden pins. Since each such pin has to be tested, then complete in-circuit as well as functional testing becomes too expensive or even unattainable due to extremely reduced test access [6].

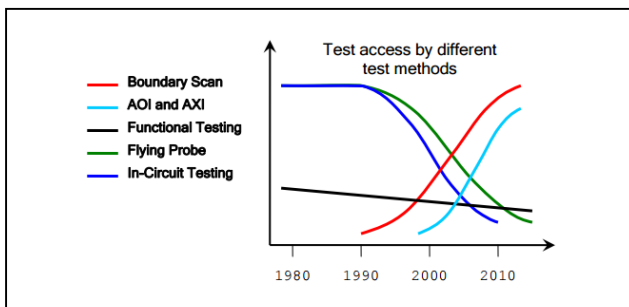


Fig. 1 - Historical Evolution of Test Methods [6]

As the result, over the last decade, there was a considerable increase in the number of test applications that require Boundary Scan as well as optical/x-ray inspection (AOI/AXI). On the other hand, most In-circuit testing and Flying Probe have dropped significantly in the same historical interval (see Fig. 1).

As an observation, no PCB testing method taken individually can guarantee full test coverage, the BS seems to be the most universal and the only realistic low-cost solution that supports structured approach and quantitative fault coverage measure.

JTAG has reached the title of dominant standard for in-circuit test, because, like many mature standards for information, it was created with a friendly environment in mind. It was designed to handle the natural enemies of digital

systems: faults in design, fabrication, packaging and PC boards. Although it provides high controllability and observability, it also confronts great security challenges. Kurt Rosenfeld and Ramesh Karri are analyzing, in their work "Attacks and defenses for JTAG" [9] a variety of attacks on JTAG systems and propose protection schemes.

In another article published in August 2014, Gian Luca Pizzocolo, Chief Engineer for Research & Development at IPSES, and Dr Bob Storey, Principal Software Engineer at XJTAG, are investigating the case of combining JTAG Boundary Scan with functional testing. This new formula proved to achieve maximum effectiveness by improving not only the reliability of the tests, but their cost and duration. The integrated approach of a Boundary Scan with functional test allows engineers to achieve full or near-full coverage of all circuits on the Device under Test (analog and digital) of all nets; shorter test time, high performance in-system programming; faster, more accurate fault diagnosis [10].

III. DUAL AXIS SOLAR TRACKER

A solar tracker, as the term suggests, represents the automated system of a photovoltaic panel, which has the objective to orient the payload toward the Sun.

Generally speaking, solar trackers come in two variants: single-axis and dual-axis solar trackers, each of them with their specific advantages and downsides. In both cases the life time of such a system is expected to exceed 20 years or more, with the only mentioning that the dual-axis model may require more maintenance during this period because of the higher complexity it exposes. Overall, a dual-axis solar tracker will be more accurate in pointing directly to the sun which is usually the brightest spot in the sky.

This section of the paper provides a short description of the design and working principle of an inexpensive active dual-axis solar tracking system for tracking the movement of the sun so as to get maximum power from the solar panels as they follow the sun.

Paper [2] proposes a similar work based on a low cost Arduino design model of a Dual-Axis Solar tracking system relying on the usage of LDR's (Light Dependent Resistors) to sense the position of the sun which is communicated to an Arduino Uno microcontroller which then commands a set of two servo-motors to re-orient the panel in order to stay perpendicular to the sun rays.

Paper [1], written by the student of the present work, tries to eliminate the need for Light Dependent Resistors by employing an original approach based on the "Cast-Shadow" principle.

To gain a better understanding of how this method works, let's consider the following scenario: A number of 40 monocrystalline solar cells, placed on a wooden plate, are divided in two major groups of 20 cells, serially connected, in order to obtain an output value of 10V. Each of the photovoltaic cells is capable of generating $V=0,5V$ and $P=0,3W$. With the information provided from the manufacturer's catalog and with the help of the equation:

$$P = V \cdot I \quad (1)$$

we can calculate the current which crosses one solar cell. Thus, the resulted current has the value $I=0,6A$.

formula we obtain $R=1/3.4$, resulting in a new angular resolution with the value $1.8/3.4=0.52$ degrees/step.

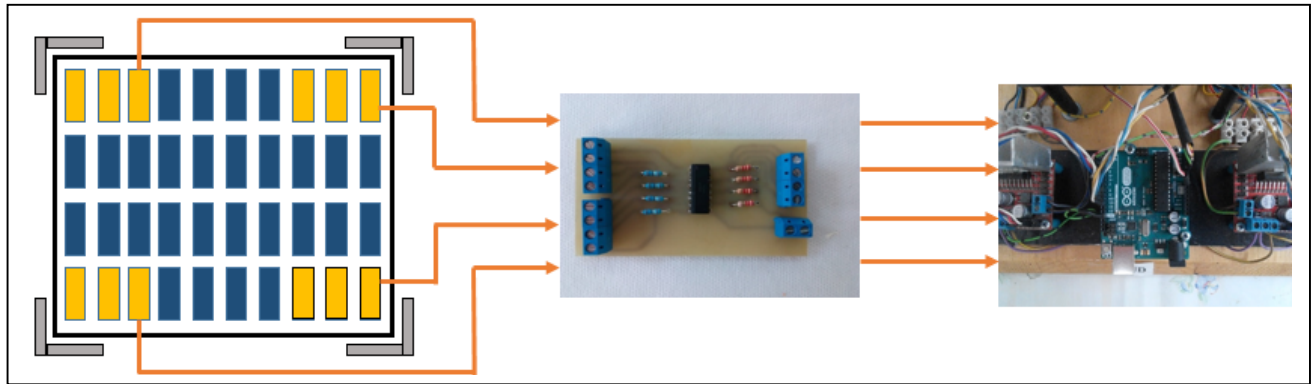


Fig. 2 – Solar tracking Concept and Connection Diagram

Once linked in a cascaded connection, the two main groups of solar cells were tethered in a parallel connection to double the current generated by one photovoltaic cell.

The original idea of the solar tracking method consists of grouping 3 photovoltaic cells in a serial connection from each corner of the solar panel (Fig.2 Left). To ensure galvanic isolation, a PCB containing a LTV-847 Optocoupler was introduced between the solar panel and the Arduino Uno (Fig.2 Center). Lastly, from here, the adjusted A-K voltages were transmitted to the Arduino board (Fig.2 Right). The Arduino microcontroller, being the “brain” of the project, processes the data via written C/C++ code and gives proper instructions to the mounted L298N circuits which command the movement of the two Stepper motors [1].

One important aspect of the work is the accuracy of the solar panel’s positioning. By mounting gears on a frame so that the teeth of the gears engage with each other, we obtained a gear train. Such a mechanical system, formed by a smaller gear of a stepping motor interacting with a bigger cogwheel which moves the solar panel, showed efficiency in improving the overall angular resolution of the stepper motors.

The mechanical advantage, or speed ratio, of a pair of meshing gears for which the input gear has N_a teeth and the output gear has N_b teeth is given by:

$$R = \frac{N_b}{N_a} \quad (2)$$

In other words, the gear ratio, or speed ratio, is inversely proportional to the radius of the pitch circle and the number of teeth of the input gear.

The horizontal, as well as the vertical motor, has a standard angular resolution equal to 1.8 degrees/step.

For the horizontal motor we have the following data: $N_a=85$, $N_b=21$. By applying the above formula we obtain $R=1/4$, which means the improved angular resolution will be $1.8/4=0.45$ degrees/step.

In the case of the vertical motor we make use of the following information: $N_a=85$, $N_b=25$. According to the

Returning to the main concept of the “Cast-Shadow” effect, we have to remind that the 3 serially connected solar cells from each corner produce an overall voltage of 1.5V. This voltage is necessary for the input of the LTV-847 to activate the photodiode located inside the Optocoupler. The only condition for the LED’s to activate is the presence of light, in our case solar radiation. By carefully analyzing the solar panel (see Fig. 2) we can observe that each corner is protected by a so called “solar barrier”. This aspect cannot be ignored because it is an important part of the automation [1].

From this perspective, the “Cast-Shadow” principle seems rather straightforward. To gain a better understanding of the process, we will provide a pseudocode that will resemble the working principle of the solar tracking system. For this, we will establish a few parameters that will suffer variations during the automation process. For this purpose, each solar cell group from the 4 corners will be designated as TR (Top Right), TL (Top Left), DR (Down Right), DL (Down Left), in concordance to their location. We will also consider the average voltages for each side as AVR (Average Voltage Right), AVL (Average Voltage Left), AVU (Average Voltage Up) and AVD (Average Voltage Down) [1].

For the horizontal motor we have:

```
AVR = ( TL + DL ) / 2
AVL = ( TR + DR ) / 2
if ( AVR > AVL ) // sun on right side, shadow on left side
    then rotate panel to the left // remove shadow
```

For the vertical motor we can write:

```
AVU = ( TL + TR ) / 2
AVD = ( DR + DL ) / 2
if ( AVU < AVD ) // sun ray on bottom, shadow on top
    then rotate panel up // remove shadow
```

Voltage imbalances can be corrected by mounting variable resistors in the output of the Optocoupler circuit. Value equalizing will eventually determine the solar tracking device to stop from moving.

IV. PROBLEM FORMULATION

In a previous manuscript [1], the main problem was the implementation of a Dual Axis Solar Tracking device which made use of the following equipment:

- Unipolar Stepper model EM-61 23LM-C352 for horizontal motor
- Bipolar Stepper model 103G771-0240 for vertical motor
- 2 Dual H-Bridge L298N specialized circuits
- LTV-847 Optocoupler
- Arduino UNO R3 microcontroller

The objective was to obtain the maximum potential for harvesting solar energy by optimizing the position of the photovoltaic panel in a programmed manner. The Arduino platform offers a C/C++ similar programming language environment making it easier to execute the coding part of the project.

The physical prototype engaged in several issues during the testing stage: overheating of Dual H Bridge L298N circuits, parasitic signals at the analog inputs of the Arduino board, imbalance of the collected voltages resulted in the rotation of the tracking device in a certain direction without stopping.

Regardless of mentioned problems, a much broader topic came in front: "How to test if the system works under the desired parameters?"

The present paper seeks to answer this question by integrating testability features at the Arduino board's circuit level design using JTAG programming/ debugging techniques. While the main focus is set on Boundary Scan method, efforts and further improvements for the Solar Tracker are made and explained to solve the remaining issues from the former paper.

V. BOUNDARY SCAN PRINCIPLE

An integrated circuit containing a Boundary Scan structure adds to each input or output pin an auxiliary memory bit (0 or 1 logic) which is called Boundary Scan cell. This cell can capture the logical level information from an input pin or write a logical level to an output pin.

The memory cells are chained to form a serial register, so they can shift sequential data from a standard TDI input (Test Data In) or transmit serial data via a dedicated standard pin called TDO (Test Data Out). The data transmission and reception is synchronized using a TCK clock (Test Clock), and the test mode operating modes are controlled by TMS (Test Mode Select) pin. The behavior of the scanned cells and the Boundary Scan logic is controlled by a dedicated controller named TAP controller (Test Access Port controller). The optional TRST (Test Reset) pin allows us to reset the Boundary Scan logic independent of the functional logic structure of the integrated circuit. It is very important to remember that the Boundary Scan (memory cells, TDI, TDO, TMS, TCK, TRST, TAP controller) are totally separate from the integrated circuit's logic structure and do not affect it in any way [4].

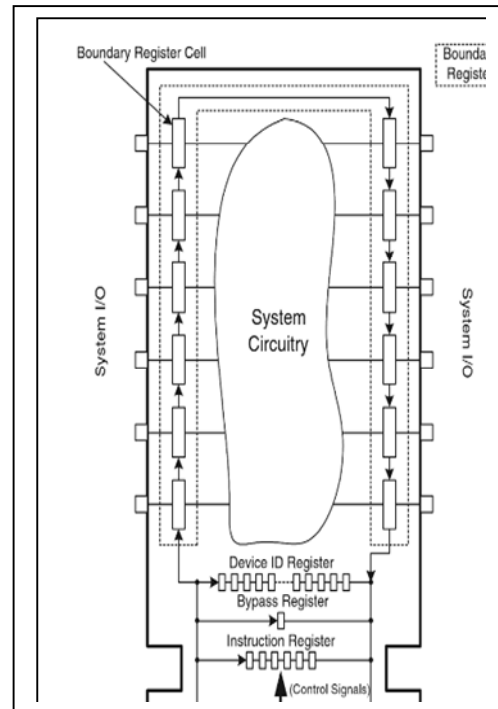


Fig. 2 - Boundary Scan IEEE 1149.1 [4]

Companies that produce integrated circuits containing this technology provide a free description of the Boundary Scan self-test behavior model for each integrated circuit in the form of a *.BSDL extension file.

It can be stated that the Boundary Scan (JTAG) test ensures high PCB testability by using only 5 standard interfacing pins using the same test in the development and production phases. In addition to the case, functional tests can be added and scheduling programming integrated, reducing development times and the cost of running test programs for populated electronic boards [5].

VI. PROPOSED SOLUTION

In order to perform a functional test on the desired integrated circuit a few aspects have to be taken into consideration:

- Deciding for the test strategy
- Verifying if the IC is compatible with the chosen method
- What equipment and/or software is required.

This section will aim to cover all three points mentioned above. In the case of integrated circuits there are a number of test technologies such as flying probe, automated optical/X-ray inspection or bed-of-nails which require specialized test equipment that will not be available on an engineer's bench.

One of the key benefits to boundary scan testing is that the only test hardware required is a JTAG controller. Using boundary scan during board bring-up can remove uncertainties – hardware engineers can test prototype boards for manufacturing defects before system testing, and even before firmware is complete. Test systems developed at this early stage of the product lifecycle can easily be reused, and extended for production.

The Arduino board from the initial Solar Tracking project was part of the AtMega328 family which didn't offer JTAG support. Therefore, a replacement with the AtMega2560 microprocessor helped passing the necessary requirements for Boundary Scan Testing.

After the compatibility check, the next step involves choosing the right equipment for testing. An affordable combination on the market today for JTAG programming/debugging is AVRDragon and the dedicated software Atmel Studio. However, in this paper, the focus is also set on seeking cheap or even free solutions for integrating Boundary Scan strategies. Therefore, a simple LPT cable which communicates with the printer port from older PC's can be transformed in a JTAG controller.

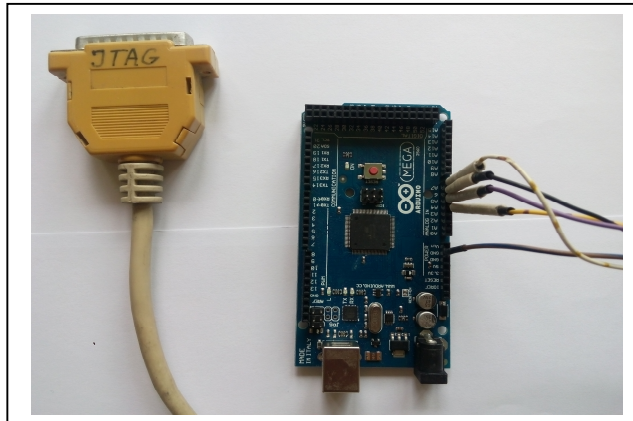


Fig. 3 – JTAG Controller

An important aspect here, and also requires a bit of research, is how to find the right correspondence between the LTP pins with the JTAG signals, and furthermore, how to connect the Boundary Scan terminals to the Arduino pins. The following table explains this chain of assignments.

Table I.

Crt. Nr.	LTP – JTAG – Arduino Pins Assignments		
	LTP Pins	JTAG Pins	Arduino Pins
1	2	TDI	A7
2	3	TCK	A4
3	4	TMS	A5
4	13	TDO	A6
5	16	Power	-

A USB adapter cable can be used to link the LTP cable to a USB port of the PC. However it is recommended to connect the cable directly to the parallel printer port of the computer, because there is a slight risk of data loss.

Therefore, tests have been performed on an older PC configuration, namely an AMD Athlon X2 5600+ Dual Core with 2 GB RAM and 40 GB hard disk space. The computer build proved powerful enough to run the Boundary Scan test at a decent speed.

In respect to the testing environment an operating system from the Unix family has been chosen, more exactly the Kali Linux distribution. It is considered the most advanced penetration testing distribution ever made. Between the testing tools of the Kali operating system, UrJTAG can also be listed. If it is not available in the latest version of the OS, there is an alternative of downloading the software directly from the Internet. The reason why a Unix operating system is preferred over the Windows version resides in the need of supplementary libraries for running the program, which can result in a frustrating experience.

The name UrJTAG comes from German, the prefix “Ur” meaning “ancestral”, an “Ur-Vater” being a forefather. In other words, UrJTAG is meant to become the prototype for many other JTAG tools [11].

The installation of the program can be executed easily using the Terminal with just a few command lines. The application can be started just by entering “jtag” in the command line.

UrJTAG offers a variety of commands for the user to become familiar with. In this section not all operations will be covered.

The first step in testing the Arduino chip was to initialize the cable type in the console window, followed by a specific argument.

```
jtag> cable EA253 parallel 0x378
```

Initializing ETC EA253 JTAG Cable on parallel port at 0x378

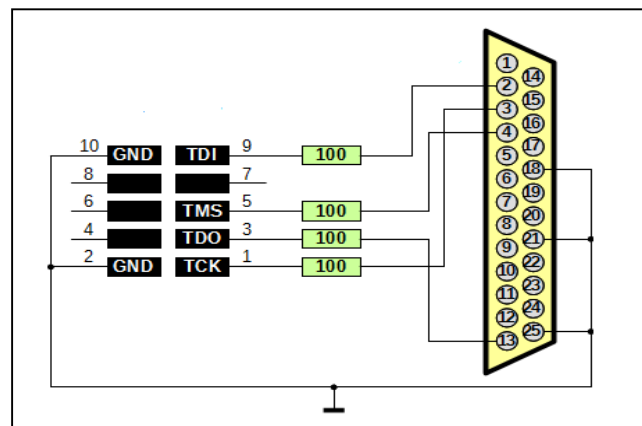


Fig. 4 – JTAG Interface [12]

The ETC EA253 cable has a very similar JTAG interface parallel structure to the homemade cable in this paper (see Figure 3). As can be observed from the above picture, proper

mounting of pull resistors is important for correct signal obtaining from each JTAG pin.

After the cable initialization, the next command that must be run is the “detect” operation. There is a high possibility of the software not being able to read the data from the tested board. In such a scenario, the program is not capable of matching the IDCODE of the board with the manufacturer’s files. The descriptions of the boards generally reside in the following branch /usr/local/share/urjtag. The directories contain a readable jtag file of the BSDL (Boundary Scan Description Language) and the necessary Steppings.

The major downside of the UrJTAG tools represents the partial support for Atmel family microcontrollers. As a result, the BSDL file for the Arduino Mega 2560 had to be downloaded from the Internet.

UrJTAG uses a different file format internally. So in order to add a new device to UrJTAG a conversion of BSDL files is required to produce a directory structure. Currently there are at least three tools available to do that; included with UrJTAG is “bsdl2jtag”.

UrJTAG contains a BSDL subsystem that retrieves the descriptions for chips in the chain from BSDL files on the fly. “bsdl2jtag” is in fact a wrapper that uses the BSDL subsystem to convert the BSDL file [11].

At the next phase, before running UrJTAG, it is necessary to convert the downloaded BSDL file in a readable jtag file for the application. This can be achieved by typing the following command line:

```
cat ATMEGA2560.bsd | bsd2jtag > jtagfile
```

After obtaining the JTAG file it is mandatory to put that file into the right place. It requires some careful editing and looking into the files resident in /usr/local/share/urjtag.

UrJTAG actually offers another option of doing the same task with equivalent functionality. From the command line the following operation can be executed:

```
jtag> bsd dump ATMEGA2560.bsd
```

This will write the relevant BSDL contents in jtag commands to stdout. However, before dumping the BSDL file it would be recommended to run a test on it. This is manageable by writing:

```
jtag> bsd test ATMEGA2560.bsd
```

After the execution of the command above, it should print the following information:

```
-N- Reading file 'ATMEGA2560.bsd'
-N- BSDL File 'ATMEGA2560.bsd' passed VHDL stage correctly
-N- Got IDCODE: 01001001100000000001000000111111
-N- BSDL file 'ATMEGA2560.bsd' passed BSDL stage correctly
```

In other words, if the IDCODE of the provided bsd file from the manufacturer’s page is identical with the ID of the Device under Test (DUT), the board will be successfully detected.

The control pin (together with clock) allows to switch device states. A state machine inside each chip can be controlled, e.g. to reset the device. This control machine also allows to have two internal shift registers in each device (although we only have on in- and one output-pin). The registers are called instruction register (IR) and data register (DR). The current UrJTAG tool allows the user to set the IR and set and get the DR. It doesn’t permit yet to directly control the state machine.

After detecting the parts of the JTAG chain, the final objective of the present paper can be reached by executing the following lines:

```
jtag> instruction SAMPLE/PRELOAD
jtag> shift ir
jtag> shift dr
jtag> dr
100011001000001000011001011111111111111111110011011
10...
```

After shifting the instruction register and data register a printed chain of ones and zero’s will appear in Terminal meaning that the board has no defects from manufacturing process.

VII. EMPIRICAL EVIDENCE AND PERFORMANCE ANALYSIS

Ensuring that the Arduino Mega 2560 is fault-free takes the present work to the final stage, the monitoring of the automated solar panel.

Before stepping into the Data interpretation part, it is necessary to make a short description of the ambient. Several experiments have been done during a full day cycle, meaning from 8:00 A.M. to 20:00 P.M. Generally speaking, the solar panel will execute a 180 degree rotation in this period of time. In this section the most relevant experiment was chosen for proving that an automated PV will generate more output voltage than a fixed panel.

For this, manual measurements have been undertaken using a multifunction tester. A very important observation here would involve the significant voltage drops during data acquisition. The reason for this would reside in the fact that the sun heat influences the connections between the solar cells in such a manner that their internal resistance rises. This represents a major challenge in solar harvesting, however it is part of a natural phenomenon and can’t be avoided. An ecological solution for this issue can be found in Paper [3] where the authors propose water cooling. This enhances the performance of photovoltaic panels in an efficient manner.

Table II.

Crt. Nr.	Data acquisition		
	Time (Hour)	Voltage (V) Static PV	Voltage (V) Automated PV
1	8:00	10,28	11,50
2	9:00	10,41	11,30
3	10:00	10,69	11,24
4	11:00	10,59	10,85
5	12:00	10,85	10,89

Crt. Nr.	Data acquisition		
	Time (Hour)	Voltage (V) Static PV	Voltage (V) Automated PV
6	13:00	10,90	11,24
7	14:00	10,40	10,47
8	15:00	10,07	10,99
9	16:00	10,95	11,15
10	17:00	11,17	11,45
11	18:00	10,94	11,54
12	19:00	9,50	11,45
13	20:00	6,6	11,40

As can be seen from the given Table, the highest recorded voltage is 11,54V, while the lowest value is 6,6V. The static panel achieves an average voltage of 10.25V per day, while the automated PV hits 11.19V in a full-day cycle. This shows an overall 9.17% voltage harnessing increase of the automated panel over the immobile photovoltaic panel.

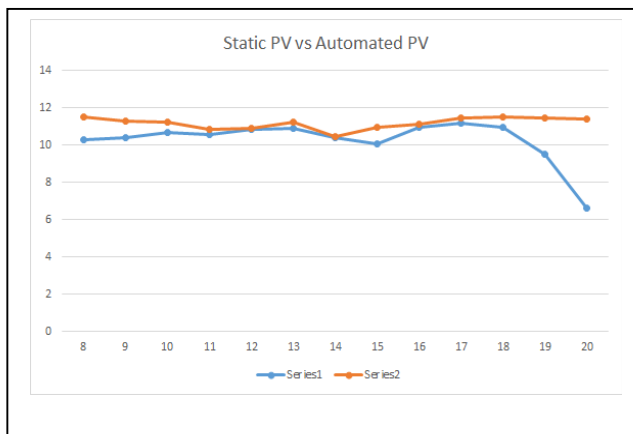


Fig. 5 - Output Voltage Comparison

The graph shows that the automated photovoltaic panel is able to maintain solar energy harvesting above the values registered by the static panel. In the eve it actually manages to outperform the fixed PV.

During peak hour, 12:00 P.M, the measured values are almost identical (10,85V and 10,89V).

VIII. CONCLUSIONS AND FUTURE WORK

The present work emphasis on finding a reliable solution for integrating testing facilities for a solar tracking device. The Boundary Scan method proved to be efficient from the speed point of view and also cost-efficient, offering a very simple serial design: One pin serial out, one pin serial output, One pin clock and one pin control.

The testing and active monitoring of the solar panel, in both variants: fixed at a 45 degree angle and positioned to the

south and also the automated version showed that the solar tracking device optimizes the position of the panel for maximum sun exposure.

Regarding the future work for this project, manual measurements are nowadays inconvenient which forces the user to act upon this problem with the help of modern technology. Therefore, an UART protocol enables two Arduino chips to be chained together in a serial communication so that the auxiliary board can read vital parameters, for instance temperature, voltages and send it via a wireless module ESP8266 to a website such as ESP Easy. From here the user can monitor gathered data in real-time. The Solar Tracking device can also be part of a much bigger project, such as a complete Home Automation.

ACKNOWLEDGEMENT

In the final part of this paper, I want to send my warm thanks to my scientific coordinator, Assistant Prof. Dr. Eng. Flavius Oprea, for all his support and guidance during the two years of Master studies and Research. I also want to show my gratitude towards his insistence on investigating testing facilities for the Solar Tracking device from my Bachelor's Degree. The books and articles he put available to me were clarifying in the field of Design for Testability. Also, his course "Testing of Computer Systems" helped me to get familiar with the IEEE 1149.1 standard, respectively with the Boundary Scan method.

REFERENCES

- [1] Raul Rotar: "Automation of Photovoltaic Panels" – Bachelor's Degree – Coordinator: Conf. Dr. Ing. Valentin Müller, "Aurel Vlaicu" University of Arad, 2015
- [2] Tarlochan Kaur; Shriya Mahajan; Shilpa Verma; Priyanka and Jaimala Gambhir: "Arduino based Low Cost Active Dual Axis Solar Tracker", 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems, 2016
- [3] K.A. Moharram, M.S. Ab-Elhady, H.A. Kandi, H. El-Sherif: "Enhancing the performance of photovoltaic panels by water cooling", Ain Shams Engineering Journal, p.869-877, 2013
- [4] Kenneth P. Parker: "The Boundary-Scan Handbook", Fourth Edition, 2016
- [5] Laung-Terng Wang, Charles E. Stroud, Nur A. Touba: "System-on-Chip Test Architectures: Nanometer Design for Testability", 2008
- [6] Understanding Boundary Scan – PCB Testing with IEEE 1149.1, Testonica
- [7] Jun Balague, "Successful ICT Boundary Scan Implementation," Circuits Assembly, September 2010
- [8] Jun Balague, "Overcoming Limited Access at ICT," Circuits Assembly, December 2008
- [9] Kurt Rosenfeld, Ramesh Karri, "Attacks and Defenses for JTAG", 2010
- [10] Gian Luca Pizzoccolo, Dr Bob Storey, "Combining JTAG Boundary Scan with Functional Testing", August 2014

[11] UrJTAG

[12] ETC EA253 JTAG Cable Schematic