Affordable Flying Probe-Inspired In-Circuit-Tester for Printed Circuit Boards Evaluation with Application in Test Engineering Education

Sorin Liviu Jurj, Raul Rotar, Flavius Opritoiu, Mircea Vladutiu
Advanced Computing Systems and Architectures (ACSA) Laboratory,
Computers and Information Technology Department, "Politehnica" University of Timisoara,
Timisoara, Romania
jurjsorinliviu@yahoo.de; raul.rotar@student.upt.ro; flavius.opritoiu@cs.upt.ro; mircea.vladutiu@cs.upt.ro

***Abstract*—This paper presents a hybrid sensorless In-Circuit Tester (ICT) design by combining the features of Flying Probe Testing (FPT) and the capabilities of a Coordinate Measuring Machine (CMM). The experimental results show that the proposed Flying Probe-inspired ICT (FPICT) is suitable for smaller sized Printed Circuit Boards (PCBs) and proves efficient regarding precision (overall precision of 95.70% for the measurements testing), test time (an average of 10.35s for a single-point test cycle), power consumption (an overall of 3.92 Watts for all considered test cases) and cost (around 25 dollars) points of view, being much more affordable and user-friendly when compared to traditional and expensive FPTs found in the industry.**

Keywords—In-Circuit Testing, Flying Probe, Coordinate Measuring Machine, Printed Circuit Board, Sensorless Testing

I. INTRODUCTION

Education is considered one of the most important factors that drive our society into new horizons, bringing new understandings of our reality and thus resulting in new and better technologies. Recent efforts in bringing affordable and equal access to education are seen also on the United Nations (UN) agenda, one example being the UN Sustainable Development Goals [1]. Regarding test engineering education, a major issue found in many of the technical schools and universities around the globe is the huge amount of technical books available but without giving the students also a chance to have a hands-on experience with real parameter values of a PCB. Concerning FPTs, this situation is usually the result of expensive ICT versions found in the industry [2], which is the leading factor for the lack of proper testing equipment in engineering laboratories.

One of the main concepts that help students be familiarized with the important and delicate process of evaluating the performance of PCB testing measurements is called testability. Testability is the property of a PCB to enable the test engineer to easily define the electronic board checking procedure at the desired level. Generally, it is given by a) mechanical parameters (the shape of the populated PCB and the test adapter design); and b) electrical parameters (access to the test samples, test methods and electrical isolation possibilities of surrounding components). ICT enables a very fast testing procedure where access can be made simultaneously on all test fields. However, this type of testing is demanding and requires a suitable electronic board design with test pads.

Despite the fact that FPTs are able to perform high-speed testing with flawless accuracy for each tested board, incorporating the latest technologies such as Boundary Scan, ICT and even Optical Inspection [3], these features require additional expensive hardware such as optical sensors and encoders as well as forcing the test engineer to reconstruct the fixture every time a new board under test is used, resulting in high costs and time consumption.

Considering these aspects, in this paper, we propose an efficient FPICT that has educational purposes and which is based on an Arduino MEGA2560 microcontroller, three ULN2003A motor drivers with their associated 28BYJ-48 Steppers as well as three Mechanical Endstop Limit Switches (MELS). Our education-oriented FPICT is designed in a way to leverage the difficulties students have when trying to learn new concepts in the domain of test engineering.

The paper is organized as follows. In Section II, we present the related work regarding modern approaches in the ICT field. Section III details the hardware components of the proposed FPICT. In Section IV, we present the sensorless-based test point tracking software implementation of the proposed FPICT. Section V describes the experimental setup and results. Finally, in Section VI we present the conclusions of this paper.

II. RELATED WORK

Placement accuracy is one of the primary issues in modern FPT systems. The authors in [4] are aware of the fact that the growing complexity of PCB's can introduce risks of faults at any stage of the manufacturing process. In their opinion, large components such as bulky capacitors and sophisticated packaging designs can obstruct the physical access of the moving probe to the required test points. To overcome this issue, the authors propose a hybrid approach based on a combination of a traditional FPT and an Automated Optical Inspection (AOI) device.

The second issue regarding modern FPT systems is the probe's navigation time between test nodes and is generally associated with the Traveling Salesman Problem (TSP). The authors in [5] investigate the ordering requirements for the complete amount of sample points and consider it an extension of the above-mentioned TSP because they use more than one probe in their research. Luciano Bonaria et al. [6] employ a test optimizer algorithm in order to reduce the test time for SPEA 4080 test equipment which makes use of 8 high-speed flying probes. Their approach is based on a dynamic greedy procedure that is able to choose the optimal sequence of tests resulting in an average test time reduction of 32%.

Test pad localization is the third issue that is concerned in recent FPT designs and can be solved by applying a clustering method that retrieves coordinates extracted from photo images of PCBs representing the potential test pads of the inspected boards. Swee Chuan Tan et al. [7] employ a two-stage clustering procedure on a 71040-pixel dataset derived from a PCB image with a precision ratio of 93.25 %

978-1-7281-7455-6/20/$31.00 ©2020 IEEE

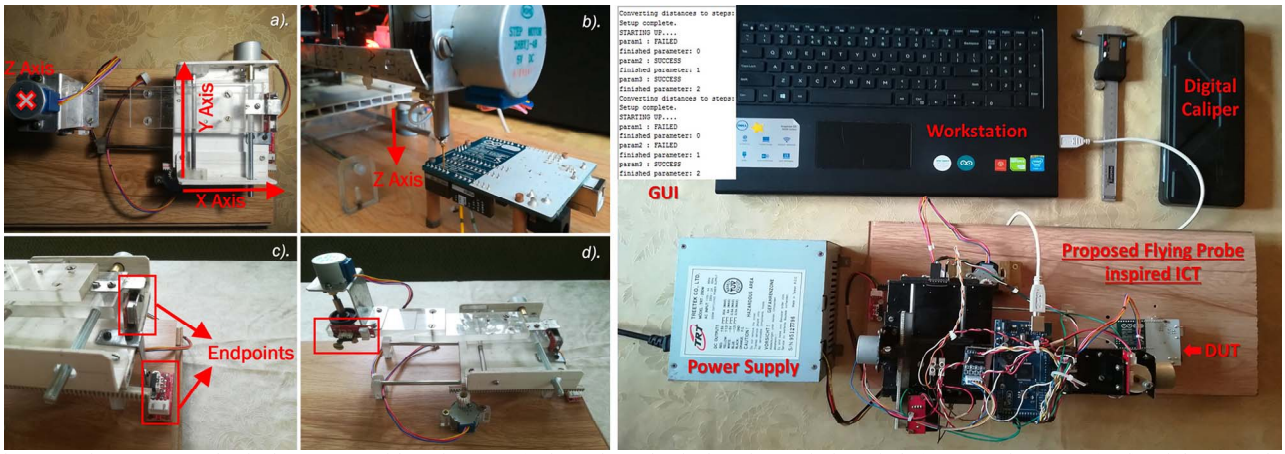

Fig. 1. Left: FPICT Mechanical Structure with Axis Array (a, b) and Mechanical Endstop Limit Switch (MELS) Placement (c, d). Right: Complete experimental setup for the proposed FPICT.

proving that their method is highly efficient in identifying test pad locations for electronic boards which lack proper documentation.

Finally, the test coverage problem is analyzed by Soh Ying Seah et al. [8] in their work which targets test load boards that are used as an interface between Automatic Test Equipment (ATE) and Integrated Circuits (IC) during package level testing. The authors highlight the fact that the SEICA Pilot LX FPT scheme and the Eagle Test ETS-364 ATE are not able to detect certain defects. To overcome this major issue, a hybrid approach between the ATE and FPT was proposed to verify four load board categories before and after merging the two methods together. The experimental results showed a substantial test coverage increase (up to 100 %) categories of load boards.

Despite the fact that the previous works focus on the feasibility of combining FPT with other test methods as well as optimizing traveling paths between test nodes, test pad localization, and fault coverage, we are implementing an FPICT that integrates the test node localization features of a CMM, resulting in a sensorless solution.

III. Hardware Components of the Proposed FPICT

Our FPICT resembles the characteristics of a Flying Probe design and the operation features of a CMM. CMMs typically specify the position of a probe from a reference position in a three-dimensional Cartesian coordinate system in terms of its displacement. Inspired by its simple yet efficient design, we constructed our own tridimensional platform (axis X, Y, and Z) motorized by Steppers which are controlled directly from an Arduino Mega 2560 mainboard. In this section, we provide a detailed overview of our proposed FPICT system, which can be divided into mechanical and electrical components.

A. Mechanical Components

The device was fixed on a parquet board that was cut according to the following sizes: Length (L) = 430 mm; Width (l) = 200 mm, resulting in a total space of A = L x l = 430 x 200 = 86.000 mm2 allocated for testing. The main platform, from where the three axes (X, Y, and Z) gather their reference points was mounted on two 190 mm long metal rods. According to Fig. 1, our initial variant is built around three main axes that we will describe in the following lines:

- ***X-Axis*** – is located at the inferior part of the main platform with the Stepper motor being fixed on the parquet board a). The translation from one direction to another is realized via a smaller cogwheel that interacts with a 130 mm long rack. This allows complete translation freedom equal to the length of the entire rack until it reaches the first MELS illustrated in the bottom part of scenario c).
- ***Y-Axis*** – is mounted on top of the main platform and contains two metal rods (both 120 mm long), one of them being a screw that interacts with the Stepper motor on the other end, as seen in scenario a). Since the stepper motor rotates the screw in two distinct directions, the secondary platform (formed of a thick Plexiglas) will be translated according to the straight drill rule, acting like a nut on the screw. However, due to mechanical constraints, the translation freedom of the Y-axis was reduced to 50 mm until it reaches the second MELS presented in the top part of scenario c).
- ***Z-Axis*** – viewed as the most complex to build of the three-axis, adopts a two-layered Plexiglas structure and is mounted directly on the previous axis system. Composed of a 60 mm hexagonal nut and combined with a screw, it functions exactly on the same principle as the previously described Y-axis. The translation limit is set to ~ 20 mm, which is sufficient for the probe (nail) to touch the contact of the PCB.

B. Electrical Components

Each of the three Cartesian axes (X, Y, Z) described above is controlled by electrical equipment consisting of a main Arduino MEGA2560 microcontroller, 3 ULN2003A motor drivers, 3 28BYJ-48 Steppers and their MELS. We will detail each of the individual parts and further argue why the chosen setup is effective from the power consumption and cost perspectives.

a) Arduino MEGA2560 Microcontroller

Arduino Mega is an ATmega2560-based microcontroller board with 54 digital input/output pins (14 of which can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), 16 MHz crystal oscillator, USB connection, energy jack, ICSP header, and a reset button. With all the listed characteristics and notably because of the large number of digital pins, it provides an optimal solution

for complex projects. The board can operate on an external energy supply of 6 to 20 volts. However, if supplied with less than 7V, the 5V connector will provide less voltage while the board could become volatile. When using more than 12V, the voltage controller can overheat and damage the board. Therefore, the suggested range is between 7 and 12 volts. According to around 8 hours of measurements at the USB plug with a flowing current of 52-54 mA, the average usage of the Arduino Mega 2560 is rated at 0,27 Watts. For our project, we use a total number of 15 digital inputs/outputs to assign pins 22-25 to the X-axis, pins 26-29 to the Y-axis, pins 30-33 to the Z-axis and pins 46-48 to receive feedback from the MELS.

b) ULN2003A Motor Drivers

ULN2003 is part of the well-known ULN200X IC series and represents a relay driver IC made up of an array of Darlington transistors. It consists of seven open pairs of Darlington collectors with prevalent emitters. In addition, ULN2003A has the ability to simultaneously handle seven different relays. A single pair of Darlington is made up of two bipolar transistors and operates between 500mA and 600mA current. ULN2003 operates on 5V using Transistor-Transistor Logic (TTL) and Complementary Metal Oxide Semi-Conductor (CMOS) technologies. Its pin configuration provides an accessible design so that the input pins are on the left side of the IC while the output pins are placed on the opposite side. The chosen IC has a broad variety of applications being frequently used as relay drivers to drive different load types. ULN2003A can also be used to drive various engines (DC motors, Steppers), logic buffers, lamp, and line drivers LED displays, motor driver circuits, etc.

c) 28BYHYJ-48 Stepper Motors

The chosen 28BYHYJ-48 Steppers are lightweight engines that are generally incorporated into DVD drives, movement cameras, and other devices that require high accuracy for a set of specific functions. The engine has a 4-coil unipolar mount and each coil is rated at +5V, making it extremely simple to manage them with any traditional microcontroller. These motor types have a 5.625 ° /64 step angle, which means that the motor will have to take 64 steps to complete one rotation and cover a 5.625 ° level for each step. Usually, these stepper motors consume high current, thus requiring an IC driver like the ULN2003 that was listed earlier. As can be seen in Fig. 1, the engine of a stepper motor has four coils: one end of the coil is tied to + 5V (Red) and the other ends (Orange, Pink, Yellow, and Blue) are grouped together and linked to the header connector of the ULN2003A. The operational voltage is rated at 5V and hence it provides sufficient torque for moving the testing probe around the Device Under Test (DUT). Only when the coils are energized (grounded) in a logical sequence the stepper motor is able to rotate in a certain direction. The logical sequence can be implemented either by using a microcontroller or a dedicated digital circuit. These types of stepper motors can be used in a variety of applications such as CNC machines, security cameras, DVD players, car side mirror tilt systems and precise control machines such as our FPICT.

d) Mechanical Endstop Limit Switches (MELS)

A limit switch is known as an electromechanical element that contains an actuator that is mechanically connected to a set of contacts (terminals). During the test of a PCB, whenever the actuator interacts with a foreign object (e.g. metal object obstacle), the MELS device is triggered and starts sending a signal to the contacts (terminals) to decide if the electrical connection should be on or off (therefore, limit switches are practical and low-cost devices that allow the user to activate or deactivate a certain process when the MELS was stimulated by an external factor with the help of a lever-type of a switch). The lever switch is wired up so that it can pull the signal to LOW when it is activated. The micro board also has an LED that will light up when the switch is activated. In our case, the MELS is used to detect endpoints for all three axes of the FPICT. Usually, MELS can be used together with RepRap Arduino Mega Pololu Shield (RAMPS) boards but can also be combined with other microcontrollers such as the AtMega2560. The maximum working voltage is rated at 200V while the current can go up to as much as 2A. The MELS serves as a reference point from which the FPICT setup will start inspecting the DUT.

IV. Sensorless-Based Test Point Tracking

The proposed FPICT process is divided into several stages that are correlated with Fig. 2. It is worth mentioning that the FPICT procedure undergoes a number of preliminary preparations known as modules before launching the main test program.

a) Axis Calibration Module

This module sets the initial coordinates (0,0,0) of the Cartesian landmark that the device sketches spatially. Any mechanical imperfection of the constructed X, Y, Z-axes may influence the accuracy of the measurements made according to the values in module d).

b) Distance to Steps Conversion Module

The conversion is made in a unique way for each of the variables declared for testing the voltage and current. It is important to note that the voltage measurement procedure differs from the one performed for the current, in that the mobile probe is connected to a single analog input A0 of the Arduino Mega board and the voltage value is obtained by measuring the test point which is always connected to the ground point.

In this case, the test program requires only three local variables denoted by *Dist_X_mm*, *Dist_Y_mm* and *Dist_Z_mm*, representing the distance of each axis to the origin of the Cartesian system. In the case of current measurement, at least two test points from where the values are collected through the main program are required. As a result, a set of variables noted with *Dist_X1_mm*, *Dist_X2_mm*, *Dist_Y1_mm*, *Dist_Y2_mm*, and *Dist_Z_mm* are declared, where the pair (X1, Y1) designates the first coordinates target point, the pair (X2, Y2) refers to the second test point and Z is the same because the height from which the probe drops stays always constant throughout the test. Because we use Stepper motors to move the axes, the test program will have to translate the values of distances in steps according to the following formula (1):

$$DistXStep = DistXmm \times StepsPerMM \quad (1)$$

where *DistXStep* represents the number of steps obtained by multiplying the Cartesian distance *DistXmm* by the value of the distance in millimeters executed by a single step of the *StepsPerMM* engine (variables to which we will return to with further explanation in this section). Additionally,

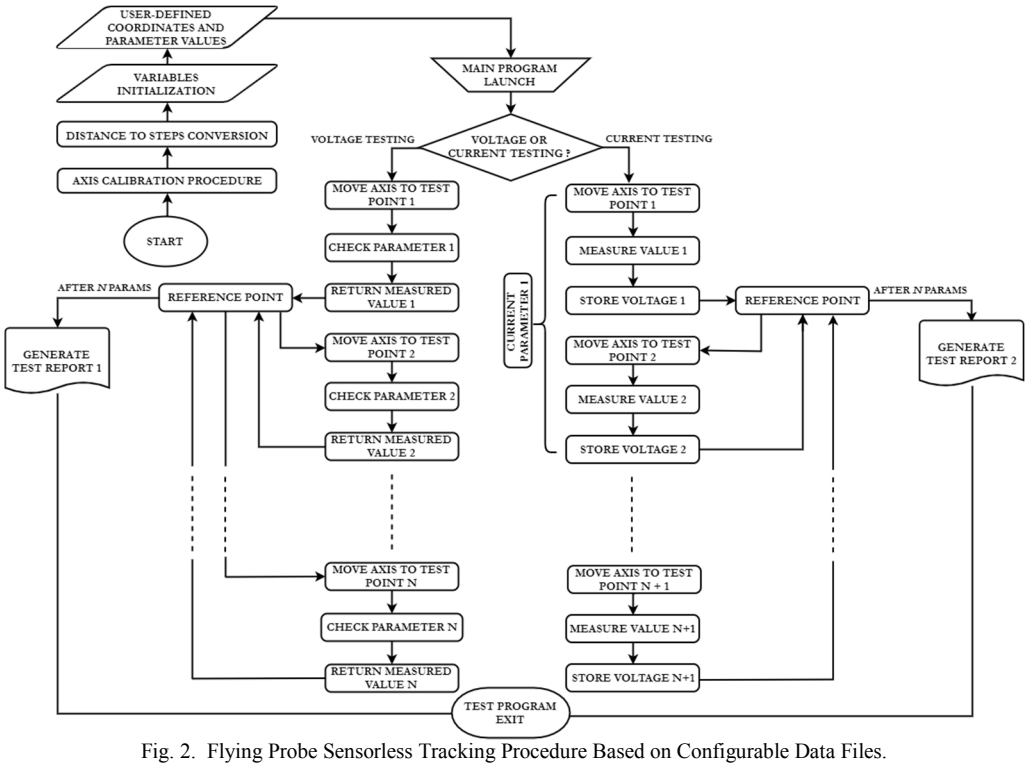

Fig. 2. Flying Probe Sensorless Tracking Procedure Based on Configurable Data Files.

because the Arduino Mega board has a built-in Analog to Digital Converter (ADC), implicit conversion of the parameters entered by the user in the same module is performed. The ADC on the Arduino is a 10-bit ADC, which means that it has the ability to detect 1.024 (2^{10}) discrete analog levels. Some microcontrollers have 8-bit ADCs (2^8 = 256 discrete levels), and others have 16-bit ADCs (2^{16} = 65.536 discrete levels). Thus, the converter generates a ratiometric value because the ADC assumes that 5V is 1.023 discrete levels and any value less than 5V (1.023 discrete levels) will be a ratio between 5V and 1.023 discrete levels. The result of the ADC in our case will be retained in a variable that appears in the relation (2):

$$CountExpectedVoltage = \frac{1023 \times ExpectedVoltage}{5} \quad (2)$$

where *CountExpectedVoltage* will count the measurements with the expected results from the tests performed.

c) Variables Initialization Module

This module covers two types of variables used: global and local. Due to the size and complexity of our code, we will list the most important variables. The global variables can be called anywhere in the code and allow a flexible modification by the domain expert:

- *STEPS_PER_REVOLUTION* – shows the total amount of motor steps for one complete rotation (360 degrees). According to the Stepper user manual, the recommended value is 2048.
- *MOTOR_SPEED1* – the value of the speed set for the first Stepper motor corresponding to the X-axis, with a default value of 15.
- *MOTOR_SPEED2* – the value of the speed set for the second Stepper motor corresponding to the Y-axis, with a default value of 15.
- *MOTOR_SPEED3* – the value of the speed set for the third Stepper motor corresponding to the Z-axis, with a default value of 13.
- *MM_PER_STEP* – the value in millimeters associated with a step executed by the Stepper motor. In order to determine this value, a trial and error experiment was used which resulted in 0.10 mm / step.
- *TOTAL_CURRENT_PARAMS* – the total number of parameters associated with the current measurement, with a set value of 4.
- *RES_CUR_MEASURE (RESCURMEASURE)* – the resistance expected in Ohms between the two test points (X1, Y1) and (X2, Y2) for measuring the current.
- *STEPS_PER_MM* – is a set value that approximates the number of steps performed per millimeter. The value found (determined by trial and error) was 100.

Local variables are the elements that underlie data processing such as the considered distances and the parameters targeted for verification. *Dist_X1_mm*, *Dist_Y1_mm*, *Dist_X2_mm*, *Dist_Y2_mm*, *Dist_Z_mm* are the distances from the reference point to the two test points associated with the measurement of the current, respectively *Dist_X1_step*, *Dist_Y1_step*, *Dist_X2_step*, *Dist_Y2_step*, the steps corresponding to the aforementioned distances. Variables *volExpectedL* and *volExpectedH* are float-type variables for setting a sensitive threshold for voltage measurements, while *CountExpectedL* and *CountExpectedH* will monitor the number of parameters outside the range of allowed values for each test. Additionally, *curExpectedL* and *curExpectedH* achieve the same minimum and maximum threshold but only for current measurement.

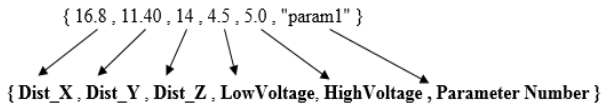

Fig. 3. Configuration file structure of a execution line for a voltage parameter.

d) Coordinates and Parameters Definition Module

The set of variables and parameters stated above will continue to be composed in the form of configurable structures that are visible to the average user. For an easier understanding, all the data that is needed to test the voltage and current were organized in cells, as can be seen in Table I.

The verification can be performed for various areas on the test board, which in our case is an Arduino UNO with two microchips of interest: Atmega328 and Atmega16u2. Both micro-devices must be powered at a voltage not exceeding 5V and generally not falling below 4V. Each deviation from the range of values considered critical (less than 4V and greater than 5V) according to the specialized catalog, can lead to a decrease of the performance (if it falls below 4V) or even to the failure of the board (if it exceeds 5V value). Additionally, the user will be able to enter the data needed to test the current parameters in the cells where the free space is represented by dots.

e) Main Program Launch

This module can be started directly from the Serial Monitor window in the Arduino IDE Suite Interface. Following the configuration of the preliminary values (Cartesian distances as well as the values of the voltage and current parameters), the main algorithm will move the calibrated probe from the considered reference point to the first test point. An example of a compact execution line for a voltage parameter during testing (which is very efficient also regarding memory) is presented in Fig. 3.

Once the test point is reached, the probe will collect the voltage value through the analog input A0 which is converted according to formula (2) into the binary system. Then, it is compared with the *LowVoltage* minimum value and the *HighVoltage* maximum value, in order to decide if the parameter is within the imposed limits (between 4V and 5V). Once the first value is measured, the moving probe will move to the reference point to continue the measurement for the next number of nodes. The same procedure is repeated for all the test points in Table I, up to the last value, in order to generate a complete report with the situation of each node (test point) separately.

According to formula (3), the mobile probe will have to collect the values of the voltages from the ends of the resistor and to divide their difference by the global variable declared in module c), namely *RES_CUR_MEASURE (RESCURMEASURE)*. The determined value of the current will be compared with the two values in the tolerance field, in order to check if it is within the limits imposed by the user.

$$curmA = \frac{voltage2 - voltage1}{RESCURMEASURE} \times 1000 \quad (3)$$

V. Experimental Setup and Results

The most fundamental resources needed when designing an effective FPT are probe positioning, measurements, test tools, development tools and time. These resources are taken into account also by our final prototype seen on the right side of Fig. 1 and were obtained by analyzing the experimental dataset summarized in Table II. The entire input dataset was determined by consulting the specialist catalog of the Arduino board for the optimal operating sizes (voltage and current). The Cartesian coordinates were measured using a digital caliper with the precision of hundreds of millimeters from the chosen reference point. In terms of power consumption, measurements were made with the multimeter, both with the laptop connected via USB to the Arduino Mega as well as just with the proposed FPICT alone (without a laptop). In the idle state, with the laptop connected, our FPICT device consumed 72mA at a 5V power supply voltage, resulting in power consumption of 0.36 W for all test scenarios. In the active state, with the X, Y, Z-axes in motion and the laptop connected, the current consumption increased to 686mA at the same supply voltage, thus achieving a power consumption of 3.43W.

The success of testing a parameter (voltage or current) is in principle given by the positioning accuracy of the mobile pin and the ability of the FPT probe to correctly read the value of the voltage or current on the already reached pin. As can be seen from Table II, the positioning accuracy when checking a single test point shows that our FPICT is capable of reaching maximum accuracy (100%), whereas, in the case of checking several nodes on the test board, the accuracy decreases in some proportion (91.40%), either from mechanical impediments that need to be revised or from the inability of the probe to take the voltage value correctly from the tested pin. The average accuracy obtained for all measurements was 95.70%, a relatively good percentage for a device built from low-cost components. The average time per test cycle (s) noted in Table II was determined for single test points and the entire test set composed of 5 parameters. Thus, 10.35 seconds were obtained for a fixed test point,

TABLE I. Example of the configuration structure for measuring voltage and current on the selected test points

Parameter Number	Cartesian Coordinates (mm)					Parameter Description						
	Voltage		Current		Voltage	Assigned Pins (for Voltage)	Targeted Microchip (for Voltage)	Targeted Microchip (for Current)	Low [V]	High [V]	Low [mA]	High [mA]
	Dist X1	Dist Y1	Dist X2	Dist Y2	Dist Z							
1	16.80	11.40	...	...	14	*7*	AtMega328	...	4.5	5	...	...
2	19.99	8	...	...	14	*20*	AtMega328	...	4.5	5	...	...
3	19.96	8	...	...	14	*21*	AtMega328	...	4.5	5	...	...
4	37.03	18.80	...	...	14	*31*	AtMega16u2	...	4	5	...	...
5	38	18.95	...	...	14	*32*	AtMega16u2	...	4	5	...	...

TABLE II. SINGLE POINT, MULTIPLE POINT AND MEASUREMENTS TESTING RESULTS

Test Type	No. of Test Samples (Parameters)	Precision Testing (%)	Average time per test cycle [s]	System Power Consumption [W]	
				Idle	*Active*
Single Point Testing	500	100	10.35	0.360	3.895
Multiple Point Testing	500	91.40	62.69		3.850
Measurement Testing	1000	**95.70**	1.53		4.027

with coordinates set from repeated tests and an average of about 1 minute for all 5 test points, each with different coordinates and distances from the reference point. The measurement time of the probe was estimated to be around one and a half seconds (1.53s).

VI. CONCLUSIONS

In this paper, we proposed a low-cost and portable FPICT device that was able to reach 100% precision in single-point testing, 91.40% precision in multiple-point testing and overall precision of 95.70% for the entire measurement testing. We believe that, due to the simplicity of our proposed FPICT and user-friendliness as compared to the ones found in the industry, students will find learning and practicing the testing of PCBs to be more fun and interesting experience. The proposed FPICT has several advantages, mainly that it is very easy to learn and use, especially because of the C written configuration files (e.g. which can be easily modified by the students in a laboratory), it has a friendly user interface and can be also quickly connected to any existent computing platform that has a USB port (Desktop PCs, laptops, tablets). Also, the proposed FPICT provides students easy access to study and experiment with the inner workings of an FPT when operating on a real PCB board, which otherwise would have been almost impossible, given the fact that the FPTs found in the industry are very expensive (i.e. thousands of dollars [2] compared to our FPICT which costs around 25 dollars and require no extra costly licenses).

REFERENCES

[1] UN Sustainable Development Goals. [Online]. Available: https://www.un.org/sustainabledevelopment/sustainable-development-goals/, last accessed 2020/03/16

[2] Huntron Workstation Tracker and Prober. [Online]. Available: http://shop.huntron.com/Workstation-License--Tracker-and-Prober_p_22.html, last accessed 2020/03/16

[3] T. Nguyen and N. Rezvani "Printed Circuit Board Assembly Test Process and Design for Testability", 9th International Symposium on Quality Electronic Design (ISQED), San Jose, CA, pp. 594-599, 2008

[4] P. Radev and M. Shirvaikar "Enhancement of flying probe tester systems with automated optical inspection", 2006 Proceeding of the Thirty-Eighth Southeastern Symposium on System Theory, Cookeville, TN, pp. 367-371, 2006

[5] Y. Hiratsuka, et al. Shin "A design method for minimum cost path of flying probe in-circuit testers", Proceedings of SICE Annual Conference 2010, Taipei, pp. 2933-2936, 2010.

[6] L. Bonaria, M. Raganato, G. Squillero and M. S. Reorda, "Test-Plan Optimization for Flying-Probes In-Circuit Testers," 2019 IEEE International Test Conference in Asia (ITC-Asia), Tokyo, Japan, 2019, pp. 19-24.

[7] S. C. Tan and S. T. W. Kit, "Fast retrievals of test-pad coordinates from photo images of printed circuit boards," 2016 International Conference on Advanced Mechatronic Systems (ICAMechS), Melbourne, VIC, 2016, pp. 464-467.

[8] Soh Ying Seah, et al. "Combining ATE and flying probe in-circuit test strategies for load board verification and test", 2009 IEEE Instrumentation and Measurement Technology Conference, Singapore, pp. 1380-1385, 2009.