An FPGA-based Testing Platform for the Validation of Automotive Powertrain ECU

Boyang Du and Luca Sterpone
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino, Italy

***Abstract*—Over the past decade, the complexity of electronic devices in the automotive systems increased significantly. The modern high level vehicles include more than 70 Electronic Control Units (ECUs) aimed at managing the powertrain of the vehicle, and improving passengers' comfort and safety. ECU microcontrollers aimed at the control of the fuel injection system have a key role. In this paper we present a new FPGA-based platform able to supervise and validate Commercial-Off-The-Shelf timer modules used in today state-of-the-art software applications for automotive fuel injection system with an accuracy improvement of more than 20% with respect to traditional approach. The proposed approach allows an effective and accurate validation of timing signals and it has two main advantages: can be customized with the exact timing module configurations to meet the exigency of new tests and allows effective modularity design test. As case study two industrial Time Modules manufactured by Freescale and Bosch have been used. The experimental analysis demonstrates the capability of the proposed approach providing a timing and angular precision of 10 ns and 10^{-5} degrees respectively.**

Keywords—Automotive; FPGA; ECU validation; eTPU; GTM.

I. INTRODUCTION

The today automotive development processes are characterized by an increasing complexity in mechanic and electronic. However, electronic devices have been the mayor innovation driver for the automotive systems in the last decade [1][2]. In this context, the requirements in terms of comfort and safety lead to an increasing number of on-vehicle embedded systems, with more and more software-dependent solutions using several distributed Electronic Control Units (ECUs).

Sophisticate engine control algorithms require performance enhancement of microprocessors to satisfy real-time constraints [3]. Moreover, the code generation, the verification and validation of the code itself, become key part in the automotive domain: the software component development processes have to be as efficient and effective as possible. Moreover, without a reliable validation procedure, the automotive embedded software can lead to a lot of errors and bugs, and decrease the quality and reliability of application software components.

Electronic devices managing the fuel injection in modern engines have a key role, in order to guarantee efficient and powerful vehicles [4]. The recent research works in the area are aimed at reducing the fuel consumption, while maximizing the power conversion and reducing air pollution emissions [5][6]. As reported in [7], a major challenge being faced in diesel technology is meeting current and future emission requirements without compromising fuel economy. Clearly, these goals could be reached through improvements in the engine electronic management. In this context, efficient fuel injection control is required [8].

Given this behavior, microcontrollers devoted to the engine management contain specific timer modules dedicated to generate, among the others, the signals used by the mechanical parts controlling the fuel injection in the cylinders [9]. The scope of these timer modules is to provide real-time generation of the signals, ensuring an efficient engine behavior. In order to achieve the correct level of synchronization between the engine position and the generated fuel injection signals, the automotive timer modules typically receive a set of reference signals from the engine; the most important ones are the *crankshaft* and the *camshaft* [10]. These signals are used to detect the current engine position, i.e., the angular position of the cylinders within the engine [11]. A precise detection of these information represents a key point for all the electronic engine management [12].

The correct programming of the timer modules is a key aspect in the automotive domain, due to the complexity of its programming code, and of the applications they have to manage. Consequently, efficient and precise validation methods and platforms are required. The current main methods to validate automotive engine applications are based on models [8][13][14], or on ad-hoc special purpose test equipment available on the market [15]. As timer module become more advanced, it raises the cost associated with validating these new modules, since extremely complex and expensive equipment must be adopted and traditional equipment are no longer able to keep up with constantly changing requirements of these systems.

In this paper, we present a new FPGA-based validation platform aimed at the validation of the applications in real-time running on the timer modules used in the modern vehicles. The purpose of this platform is to provide the developers of automotive applications with a flexible and efficient architecture able to effectively validate the code running in the most popular timer modules. More in particular, the proposed platform has the capability of generating the engine reference signals (i.e., the crankshaft and camshaft reference signals) that are typically used by the automotive microcontrollers to generate the fuel injection signals, and acquiring the signals generated by the timer module under test, verifying the synchronization between these signals and the provided engine reference signals. The proposed platform is useful to validate the functions of several

978-1-5090-3561-8/16/$31.00 ©2016 IEEE

timer modules running in different engine configurations (e.g., with different profiles of the crankshaft and camshaft signals). The platform can be customized with the exact instrument modules to meet the exigency of new test; it has flexible functionalities for diverse test bench purposes. Finally, it allows effective testing of modular designs and if compared with traditional approaches used to test state-of-the-art timer modules, it has an accuracy improvement of more than 20% [15] providing a timing and angular precision of 10 ns and 10^{-5} degrees respectively.

As case study, we use two important timer modules used today in the automotive domain: the *Enhanced Time Processor Unit (eTPU)* developed by Freescale [16][17], and the *Generic Timer Module (GTM)* developed by Bosch [18]. We choose these two modules since they are used, among the others, for the generation of the fuel injection signals in several engines; moreover, eTPU and GTM represent a good set of benchmarks, since the way in which they manage the automotive applications are different: in fact, in the eTPU several software routines share the same processing unit, while in the GTM several tasks can be directly managed by hardware parallel processing units.

The acquired results demonstrate the validity of the proposed approach, since using the proposed platform, it is possible to verify the synchronization between the inputs and the generated fuel injection signals with a very high degree of precision. Moreover, the proposed platform is able to verify the signals synchronization both in static engine conditions (i.e., constant engine speed), and in dynamic conditions (i.e., variable engine speed).

The rest of the paper is organized as follow: Section II overviews some previous work in the area of the automotive electronic control development and validation; Section III describes the main task performed by the today automotive timer modules embedded in the recent microcontrollers; Section IV details the proposed validation platform, while in Sections V the experimental results are shown. Finally, Section VI draws some conclusions and outlines the future works.

II. RELATED WORKS

With the development of electronic technology and the application of control theory in the automotive control [19], many research works have been developed with the purpose of improving the control of the fuel injection. The motivation of these research works is that, nowadays, the fuel injection system is the most important part of diesel engines, and its working state directly influences the performance, the consumption and the air pollution, as documented in [4][8][12].

In [4] the authors presented a new fuel injection intelligent control system, designed to improve the testing accuracy. The proposed system can automatically test the state of the injection pump, and it obtains all the parameters of the fuel injection system without human intervention by the use of PC and AT89C52 single chip microcomputer. Such system is designed and realized on the SYT240 fuel injection system test platform, which can automatically fetch and display the main parameters. Although the approach presented seems to be promising, it is strongly based on the usage of a dedicated test platform.

In [5] the authors faced the problem of improving the accuracy of the engine control electronic, and they affirmed that one potential way to do this is by using real-time in-cylinder pressure measurements. Consequently, the authors proposed an approach that derives the pressure information from the measurement of the ordinary spark plug discharge current. The motivation of this work is that, by monitoring the pressure of each cylinder, the electronic engine control can be optimized in terms of fast response and accuracy, thus enabling online diagnosis and overall efficiency improvement.

Another research work addressing the usage of cylinder pressure-based combustion control is presented in [7], where the authors explained that in case of multiple fuel injections, the timing and the width of the fuel injection pulses need to be optimized. More in particular, this paper presents several methods in which the cylinder pressure signal is used for multiple-pulse fuel injection management for a diesel engine capable of running in low-temperature combustion modes.

In [6] the authors explained that it is important to avoid discrepancies between the fuel amounts injected into the individual cylinders, in order to avoid excessive torsional vibrations of the crankshaft. Consequently, the authors presented a general adaptive cylinder balancing method for internal combustion engines; the proposed algorithm uses online engine speed measurements. The motivation of this work is that due to varying dynamics and ageing of components of the fuel-injection systems, there may be a significant dispersion of the injected fuel amounts.

In order to implement all the fuel injection optimization methods proposed above, in the real engine behavior, the engine angular position has to be identified as precisely as possible. In [16] the authors presented an example of engine position identification by using the eTPU module embedded in the MPC5554 microcontroller.

Another work addressing the problem of a precise angular engine position detection is reported in [12]; more in particular, the authors explain that, due to mounting and packaging tolerances, the magnetic field at the sensors position varies, resulting in angular measurement. Mounting and packaging tolerances cannot be avoided; consequently, the authors proposed a compensation method based on a new filter structure.

Summarizing, several research works have been developed in the last decade in order to optimize the fuel injection systems of the today's vehicles. In order to achieve this goal, the usage of specific timer modules, i.e., the eTPU and the GTM, is required; the main tasks typically managed by these modules are to acquire in a very precise way the engine angular position, and to generate, among the others, the signals aimed to control the cylinders' fuel injection. To do this, these modules have to be configured, using their specific programming code; this task is a difficult and important part of the fuel injection control systems development, since a small error can cause relevant problems in the engine behavior, thus compromising the efficiency of the entire system.

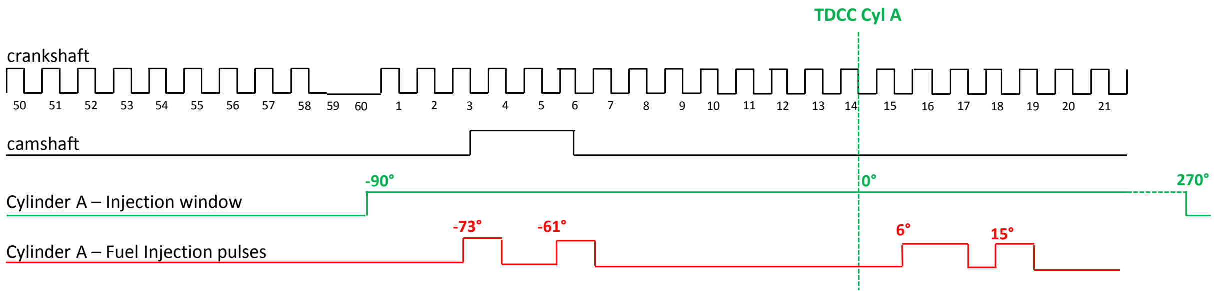

Fig. 1. Example of the main signals received and managed by the automotive timer modules, in order to efficiently supervise the engine behavior.

In this paper we propose an FPGA-based validation platform able to emulate the engine behavior (i.e., generate the crankshaft and the camshaft signals), and to acquire the fuel injection signals generated by the timer module under test; using this platform, it is possible to validate the timer module in a very precise way since the precision of the data acquisition is about 10^{-5} engine angular degrees, which is an improvement of more than 25% with respect to traditional methods [14][15]. The effective synchronization validation between the input and the output signals is performed by the developed software framework which compares the data achieved during the experimental analysis with the expected ideal values. To the best of our knowledge, this is the first work addressing the generation of a flexible and low-cost validation platform for the today's automotive microcontrollers. The main contribution of this work is to provide for the developers of the automotive applications a precise and flexible validation platform, useful to check the correctness of the developed software routines, thus ensuring an efficient system development. Moreover, using this platform it is possible to check the functioning of the real microcontroller, avoiding the unexpected misbehaviors due to the model-based validation of the developed applications.

III. Timer Modules in Automotive Applications

Today's automotive microcontrollers contain specific timer modules for managing the engine signals. In Fig.1, the most important signals related to the cylinder's fuel injection are shown. All the main tasks performed by the automotive microcontrollers are based on a precise detection of the engine's angular position (i.e., the precise position of the cylinders with respect to the crankshaft). This is done using the two reference signals coming from the engine, i.e., the crankshaft and the camshaft. The crankshaft, typically, is a square wave signal, where each falling edge transition represents a partial rotation of the crankshaft. For example, if the crankshaft phonic wheel [20] is composed of 60 teeth, each falling edge transition of the crankshaft signal indicates a rotation of 6°. Moreover, in a determinate position of the crankshaft signal, a *gap* (i.e., a missing tooth) is present: this gap is used as reference point to understand the correct engine angular position [25]. On the other side, the camshaft is a signal composed of few pulses synchronized with the crankshaft. Since the engines addressed in the context of this paper are 4-stroke engines, the complete 4-stroke sequence (i.e., intake, compression, power, and exhaust) takes two full rotations of the crankshaft. By only looking at the crankshaft signal, there is no way to understand if the crank is on its intake-compression rotation or on its power-exhaust rotation. To get this information, the camshaft signal is required; moreover, due to the 4-stroke configuration, the camshaft rotates at half the crankshaft speed (a rotation of 360° of the camshaft implies a rotation of 720° of the crankshaft); consequently, a signal generated once per rotation of the camshaft is sufficient to supply the required information. According to the features of these signals, the *Top Dead Cylinder Center (TDCC)* for each considered cylinder is identified [21]. The fuel injection pulses are electronic pulses that act on the fuel injector of each cylinder. These pulses have to be generated in a very precise angular position, where the reference point is the TDCC. The range in which these pulses can be generated is called *Injection Window (IW)*; typically, the width of the IW is 360°. In the context of this paper, we consider a maximum number of injection pulses equal to 16. The angular position of the beginning of an injection pulse is called *Start Of Injection (SOI)*, or *Start Angle*; moreover, the injection pulses can be programmed using a temporal displacement between them (*Dwell*). Finally, in the typical automotive applications, timer modules generate, also, a sequence of high frequency *pulse width modulation (PWM)* pulses, in order to trigger other engine sensors.

The generation of the fuel injection pulses is not the main goal of this paper and, consequently, we do not explain other details about this topic. The method proposed in the next section is focused on the verification of the precision and of the synchronization of the generation of the fuel injection pulses.

IV. The Proposed Validation Platform

The main purposes of the proposed platform are to generate the engine reference signals, and acquire the fuel injection pulses generated by the microcontroller under test, correlated with the provided engine reference signals itself. In Fig. 2 the overview of the proposed validation behavior is shown. It is composed of the FPGA-based validation platform and of the external controlling computer.

The validation platform is composed of the 32-bit Xilinx MicroBlaze processor core [22] and of a set of special purpose DSP peripherals designed to support the validation of the signals generated by the timer module under test. Both the MicroBlaze processor and the DSP peripherals are implemented in a FPGA device. The communication between the processor and the DSP are managed using a Processor Local Bus (PLB) interface. An external controlling computer is directly connected to the processor, in order to provide the input parameters required to the generation of the engine signals. Moreover, a RS232 interface is used to save the acquired data into a database contained in the external computer itself.

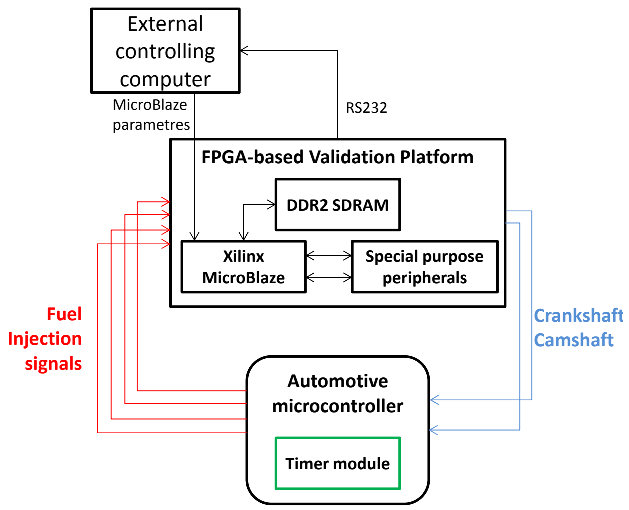

Fig. 2. The overview of the proposed validation platform

There are the two most important DSP peripherals developed in this context: the crank/cam generator which consists on the module used to generate the crankshaft and the camshaft signals according to the user parameters, and the measurement module which is used to sample and store the signals provided by the timer module under test.

A. *The Crankshaft and Camshaft DSP peripheral*

In Fig. 3 the composition of the crankshaft and camshaft DSP peripheral is shown. It is composed of three main sub-modules: *clock_div*, *Selector Crank*, and *Selector Cam*. The goal of this peripheral is to generate the crankshaft and camshaft signals in two different ways, according to the user requirements: in the former, the signals have to emulate the signals of an engine running with constant *rounds per minutes (rpm)*, while in the second case these signals have to emulate the dynamic behavior of an engine, i.e., non-constant rpm.

The clock_div sub-module receives in input from the MicroBlaze processor (through the use of *slave registers*) the 32-bit values *Delta_period* and *Num_cycles*; the former is a timeout value (expressed in number of clock cycles) in which the period of the crankshaft teeth has to remain constant. When this time is elapsed, the *request signal* is raised in order to communicate at the MicroBlaze processor (through the use of an interrupt) that the new speed parameters can be sent; this mechanism allows to generate dynamic or static crankshaft and camshaft signals. The second value received by the peripheral (i.e., Num_cycles) represents, instead, the actual speed of the engine: in particular, it is the duration, expressed in number of clock cycles, of half period of the crankshaft signal. The internal circuitry (i.e., the Selector Crank sub-module) causes a toggle of the output crankshaft signal every *Num_cycles* clock cycles. A similar approach is used to generate the output camshaft signal: the signal *Cam_Enable_2*, generated by the Selector Crank sub-module, reports at the Selector Cam sub-module when toggle the camshaft output signal. This has been done to ensure a very precise synchronization between the crankshaft and the camshaft signals generated by this peripheral.

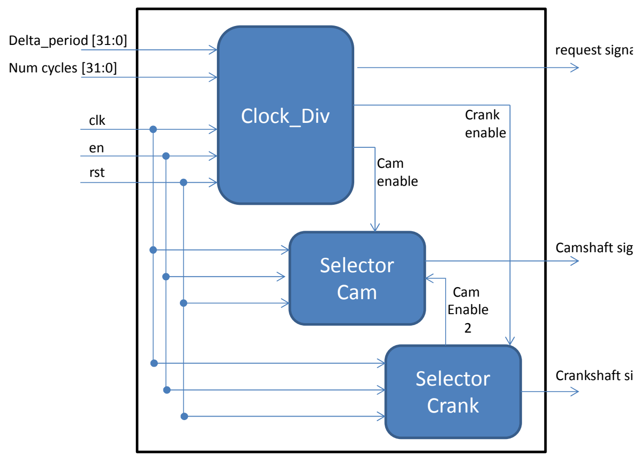

Fig. 3 The crankshaft and camshaft generator peripheral.

B. *The measure DSP peripheral*

The measure peripheral receives at input the injection pulses and the PWM pulses signals, both generated by the timer module under test; using the crankshaft and the camshaft signals (generated by the specific peripheral explained in the previous section) as absolute reference, it performs the required measurements of the input signals. This peripheral is able to measure the signals for one cylinder at a time.

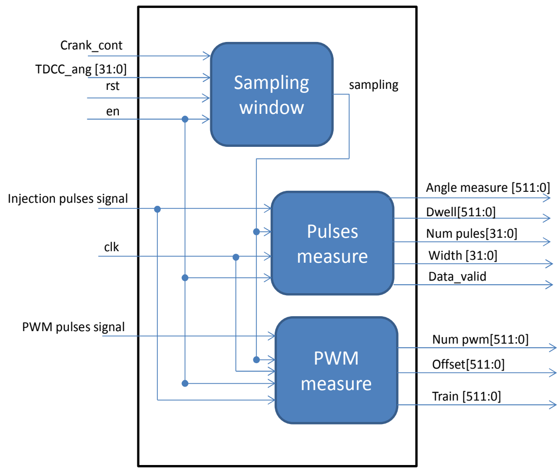

Fig. 4 The measure peripheral.

In Fig. 4, the main sub-modules composing the measure DSP peripheral are shown. The *sampling window* sub-module triggers the other two sub-modules: it receives in input (*TDC_ang* signal) from the MicroBlaze processor (through the use of a slave register) the number of the crankshaft tooth falling edge corresponding to the angular value of the TDCC referred to the cylinder to be monitored. It also receives in input the crankshaft signal generated by the other peripheral (*crank_cont* signal). By counting the falling edges of the crankshaft signal, the sampling window sub-module is able to produce in output the *sampling* signal, that remains active (i.e., logic value equal to 1) for 360° according to the programmed cylinder. This signal is connected to the *pulse measure* and to the PWM *measure* sub-modules, in which it is used as "clear" signal for the internal counters.

The pulses measure sub-module counts the number of incoming pulses (*injection pulses signal*) and exploiting a couple of latches, it stores in a set of internal signals the time stamps at which the rising and falling edges occur. To do this, an internal counter is used, where the frequency of the counter itself is the same of the frequency of the *clk* signal; moreover, this counter is reset at the beginning of each injection window. As soon as the sampling signal becomes 0, all the stored results are transferred to the respective 512-bit outputs. The length of the output signals is due to the fact that inside a programming window, there should be at most 16 injection pulses; consequently, we have to store 16 parameters, each one represented by 32 bit. The output data of this sub-module are: (1) *angle measure*, which contains the measured start angle time stamps of the injection pulses; (2) *dwell*, which contains the measured time between the current pulse and the previous one; (3) *num pulses*, which indicates the total number of detected pulses inside the injection pulses; (4) *width*, which contains the measured time duration of each injection pulse; and (5) *data valid*, that signals when all the measured values have been copied in the output signals; this value is used to signal at the MicroBlaze processor (through an interrupt) that the current data could be transferred in the SDRAM.

A similar approach has been used for the PWM measure module, in which the features of the PWM signal generated by the timer module under test are measured. This module is activated once the injection pulses signal is detected. The output data of this module are: (1) *offset*, which contains the measured offset from the fuel injection pulses falling edge to the first PWM rising edge; (2) *train*, which contains the measured PWM train duration (in terms of number of clock cycles), from the first rising edge to the last falling edge; and (3) *num_pwm*, which contains the number of counted PWM pulses.

C. The MicroBlaze processor tasks

Considering the special purpose peripherals described in the previous sections, the MicroBlaze processor has three main tasks. The first is to provide the parameter data to the crankshaft and camshaft generator peripheral. The second performed task is, whenever the data valid signal is received from the measure peripheral, transfer the acquired data (contained in the output signals of the measure peripheral) into the SDRAM; this allows to acquire the same data (about the fuel injection and the PWM signals) in different time instants, in order to characterize in a very precise way the measured signals produced by the microcontroller under test. Finally, when the required number of measurements have been acquired, the MicroBlaze processor takes care of sending (through a RS232 interface) the data contained in the SDRAM to the external controlling computer.

D. The output data format

All the measurements acquired by the peripherals described in the above sections, are based on temporal intervals and are contained in a file that we call *data_raw.txt*.

Since the main purpose of the proposed validation platform is to verify the synchronization of the fuel injection pulses with respect to the engine reference signals, we need to translate the acquired data about the pulses start angle from the FPGA time domain (i.e., the number of clock cycles measured by the peripheral) to the engine angle domain. To do this, the following formula has been used:

$$StartAngle\ [\text{deg}] = \frac{(6 * StartAngle\ [ClockCycle\ Num] * RPM)}{ClockCycle\ Period\ [s]}$$

where the degrees of each crank falling edge transition are 6, and RPM are the current rounds-per-minute of the engine. Clearly, this formula can be used in the case of constant engine rpm; in case of dynamic engine behavior, the formula has to take in consideration the different RPM values during the measurements interval.

V. EXPERIMENTAL RESULTS

In Fig. 5 the experimental flow is shown. A Matlab parser translates the data acquired by the proposed validation platform. Then, these data are compared with the file containing the expected values (called *ideal_values.txt*); this file is generated by an ideal data generator written in C language. As a result, several report files are obtained; these files contain the details about the measures, including the average error, the maximum error and the standard deviation of each feature of each injection pulse generated by the timer module under test. As case study, we used two timer modules: the eTPU and the GTM; in the following section we explain the main features of these modules and the obtained measurements results.

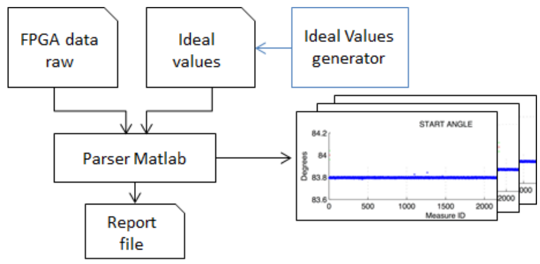

Fig. 5 The experimental flow.

A. Enhanced Time Processor Unit (eTPU)

The Enhanced Time Processor Unit (eTPU) [16][17] by Freescale, is an effective timing co-processor available in the automotive domain; it is used to efficiently managed I/O processing in advanced microcontroller units. From a high level point of view, the eTPU has the characteristics of both a peripheral and a processor, tightly integrated between each other [23]; essentially, it is an independent microcontroller designed for timing control, I/O handling, serial communications, and engine control applications [17]. More in particular, the eTPU is mainly used to decode the engine angular position, and, consequently, to control actuators such as the fuel injectors and the spark plugs, thanks to the high flexibility of the dedicated programmable hardware.

In the context of this paper, we used the eTPU module embedded in the microcontroller SPC5644AM; moreover, we used the automotive functions set available in [24].

B. Generic Timer Module (GTM)

The Generic Timer Module (GTM) [18] is a recent hardware module provided by Bosch. It is composed of many sub-modules with different functionalities. These sub-modules can be interconnected together in a configurable manner in order to obtain a flexible timer module for different application domains. The scalability and configurability is reached by means of the architectural structure of the module itself: a set of dedicated sub-modules is placed around a central routing unit, which is able to interconnect the sub-modules according to the programmed configuration specified in the running software [18]. The GTM is designed to run with a minimal CPU interaction and to unload the CPU itself from handling frequent interrupts service requests.

In the context of this paper, we used the GTM module embedded in the microcontroller SPC574K72; moreover, we directly implemented a set of automotive functions useful to generate the fuel injection pulses using only the GTM module.

C. The used Xilinx FPGA board

In order to implement the proposed validation platform, the Xilinx Virtex-5 XC5VLX50T FPGA has been used. This FPGA is embedded in a Digilent Genesys board. The working frequency of the MicroBlaze processor implemented in the FPGA itself in 100MHz; also the clock frequency provided at the special purpose peripherals is 100MHz. This allows us to obtain time measurements with a precision of 10ns, and angular measurements with a precision of 10^{-5} degrees.

D. Main obtained results

The main purpose of this section is to give the reader an idea of the effective features of the proposed validation platform, highlighting its capability of making measurements with extreme precision.

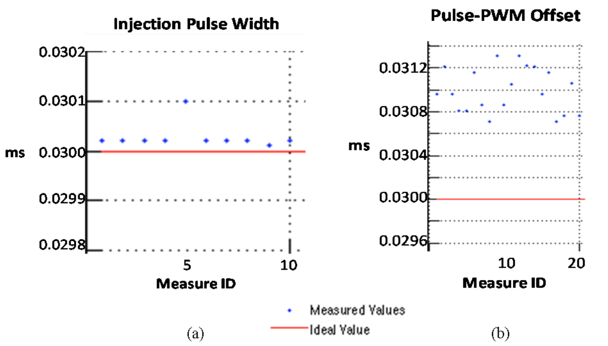

Fig. 6 Measures of injection pulse width (a), and of the offset between the injection pulse and the pwm signal (b); both the signals are generated by the GTM.

In Fig. 6 two graphs are shown: the first (Fig. 6.a) reports the measurements of the width of an injection pulse, while the second (Fig. 6.b) shows the measurements of the PWM offset; in this case the module under test is the GTM. In Fig. 7 a graph reporting the measurements of the start angle of an injection pulse generated by the eTPU module is shown. By looking at this graph, it is possible to understand if the injection pulse is generated in the correct position, i.e., in a determinate angle position, according to the crankshaft and the camshaft signals. In this case, the precision of the measurements is 10^{-5} degrees.

As it is possible to notice by the graphs reported in this section, using the proposed validation platform it is possible to understand if the injection pulses are correctly generated. This allows to understand if the software applications running in the timer module are correct (ensuring a real-time behavior) or contain software bugs. Using this platform, thus, the developers of automotive applications can verify if the applications they developed are able to efficiently manage the fuel injectors.

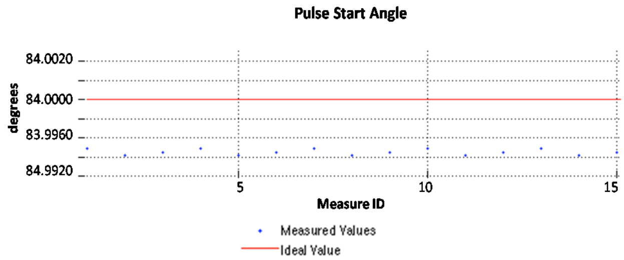

Fig. 7 Measures of the Start Angle of an injection pulse generated by the eTPU module.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper we present a new platform for the validation of timer modules used in automotive applications. The high flexibility, combined with the capability of extreme precise measurements, makes the platform very suitable to be used by the developers of automotive applications during the software development. As case study, we used two important timer modules employed in the today's vehicles.

As future work, we plan to extend the proposed FPGA-platform in order to inject faults in the input signals provided to the timer module under test, checking the correctness and synchronization of the generated output signals.

REFERENCES

[1] E. Armengaud, A. Steininger, M. Horauer, "Towards a Systematic Test for Embedded Automotive Communication Systems," IEEE Transactions on Industrial Informatics, vol. 4, n.3, 2008.

[2] M. Steger, C. Tischer, B. Boss, A. Muller, O. Pertler, W. Stolz, S. Feber, "Introducing PLA at Bosch Gasoline Systems: Experiences and Practices," Springer Software Procuct Lines, Lecture Notes in Computer Science, vol. 3154, pp. 34 – 50, 2004.

[3] Y. Kanehagi, D. Umeda, A. Hayashi, K. Kimura, H. Kasahara, "Parallelization of automotive engine control software on embedded multi-core processor using OSCAR compiler," IEEE Cool Chips XVI, p. 1 – 3, 2013.

[4] F. Juan, M. Xian-Min, "Research on fuel injection intelligent control system," IEEE Conference on Industrial Electronics and Applications (ICEA), pp. 2782 – 2785, May 2009.

[5] A.D. Grasso, S. Pennisi, M. Paparo, D. Patti, "Estimation of in-cylinder pressure using spark plug discharge current measurements," European Conference on Circuit Theory and Design (ECCTD), pp. 1 – 4, 2013.

[6] F. Ostman, H.T. Toivonen, "Adaptive Cylinder Balancing of Internal Combustion Engines," IEEE Transacrions on Control System Technology, vol. 19, n. 4, pp. 782 – 791, 2011.

[7] I. Haskara, W. Yue-Yun, "Cylinder Pressure-Based Combustion Controls for Advanced Diesel Combustion With Multiple-Pulse Fuel Injection," IEEE Transactions on Control Systems Technology, vol. 21, n. 6, pp. 2143 – 2155, 2013.

[8] Q. Lui, H. Chen, Y. Hu, P. Sun, J. Li, "Modeling and Control of the Fuel Injection System for Rail Pressure Regulation in GDI Engine," IEEE/ASME Transactions on Mechatronics, vol. 19, n.5, pp. 1501 – 1513, 2014.

[9] J. Larimore, E. Hellstrom, S. Jade, J. Li, "Controlling Combustion Phasing Variability with Fuel Injection Timing In a Multicylinder HCCI Engine," American Control Conference (ACC), pp. 4435 – 4440, 2013.

[10] T. A. Johansen, O. Egeland, E. A. Johannessen, R. Kvamsdal, "Free-piston diesel engine timing and control - toward electronic cam- and crankshaft," IEEE Transactions on Control System Technology, vol. 10, n. 2, pp. 177 – 190, 2002.

[11] S. Hainz, E. Ofner, D. Hammerschmidt, T. Werth, "Position Detection in Automotive Application by Adaptive Inter Symbol Interference Removal," IEEE 5th conference on Sensors, pp. 1103 – 1106, 2006.

[12] S. Hainz, D. Hammerschmidt, "Compensation of Angular Errors Using Decision Feedback Equalizer Approach," IEEE Sensor Journal, vol. 8, n. 9, pp. 1548 – 1556, 2008.

[13] F. Li, T. Shen, X. Jiao, "Model-based design approach for gasoline engine control Part I: Modeling and validation," 32nd Chinese Control Conference (CCC), pp. 7774 - 7779, 2013.

[14] E. Alabastri, L. Magni, S. Ozioso, R. Scattolini, C. Siviero, and A. Zambelli, "Modeling, analysis and simulation of a gasoline direct injection system," in *Proc. 1st IFAC Symp. Adv. Automot. Contr.*, 2004, pp. 273–278, 2004.

[15] National Instrument Data-Sheet, "Counter/Timer Overivew", pp. 386 – 393, 2013.

[16] W. Dafang, L. Shiqiang, J. Yi, Z. Guifan, "Decoding the Engine Crank Signal Referring to AUTOSAR," Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on, pp. 616 – 618, March 2011.

[17] Enhanced Time processor Unit (eTPU). Available [Online]: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=eTPU

[18] Generic Timer Module (GTM) Product Information. Available [Online]: http://www.bosch-semiconductors.de/media/en/pdf_1/ipmodules_1/timer/bosch_product_info_gtm_ip_v1_1.pdf.

[19] A. Ohata and K. R. Butts, "Improving model-based design for automotive control systems development," in *Proc. 17th World Congr.*, ,pp. 1062–1065, 2008.

[20] X. Ying, Y. Qiangqiaang, L. Fuyuan, "Research of Crankshaft Grinding Wheel Dresser Based FANUC NC System," 3rd International Symposium on Information Processing (ISIP), pp. 189 – 192, 2010.

[21] T. Yamanaka, M. Esaki, M. Kinoshita, "Measurement of TDC in Engine by Microwave Technique," IEEE Transaction on Microwave Theory and Techniques, vol. 33, n. 12, pp. 1489 – 1494, 1985.

[22] Xilinx MicroBlaze details. Available [Online]: http://www.xilinx.com/tools/microblaze.htm

[23] C. Rodrigues, "A case study for Formal Verification of a timing co-processor," 10th Latin American Test Workshop (LATW), pp. 1-6, 2009.

[24] Freescale automotive funcions set. Available [online]: http://www.freescale.com/webapp/etpu/

[25] Freescale Application Note. Available [Online]: http://cache.freescale.com/files/32bit/doc/app_note/AN3769.pdf