Rapid Design and Test of Embedded Control Systems Using LABVIEW-FPGA Tool

Dr. Maher Algreer **Mohammad Tarik Mohammad**
E-mail: maher.algreer@gmail.com, E-mail:mohammed_t1990@yahoo.com
University of Mosul, College of Engineering, Computer Engineering Department, Mosul/ Iraq.

Abstract

Hardware-description Language (HDL) is typically used to synthesise the digital hardware of the control systems. Importantly, this requires a deep knowledge in digital hardware design; howeverthis is not essential for the design of the real time control systems. From this prospective, there is a great interest to employ a modern environment tool to simulate, design, validate and to rapidly implement the hardware to the target of the application. For this reason, this paper aims to presentthe methodology and effectiveness of using the LABVIEW-FPGA toolin embedded system design of digital control algorithms. As the model of the control system has been already simulated using the LABVIEW environment, therefore this will shorten the time of hardware implementation, where the designed control algorithm will directly translate into hardware resources by using LABVIEW-FPGA module. The methodology of hardware digital controller design is clearly explained using LABVIEW-FPGA modulebased SPARTAN-3EFPGA from Xilinx. The prototyped temperature control system using (CI-53003) is accommodatedas one of the examplesto demonstrate the embedded hardware design of digital control system. Experimental results clearly show the successful hardware implementation of the designed algorithm.

Index Terms- LABVIEW-FPGA, Industrial Control, Rapid Hardware Implementation, Temperature Control, FPGA, Embedded system, Real time system.

التصميم السريع وإختبار انظمة السيطرة المطمورة باستخدام LABVIEW-FPGA

ماهر الغرير **محمد طارق محمد**
قسم هندسة الحاسوب / كلية الهندسة / جامعة الموصل

الخلاصة

إن لغة وصف المكونات المادية تستخدم عادة في تصميم المكونات المادية للأنظمة المطمورة. إن عملية التصميم تحتاج الى معرفة عميقة في تصميم المعمارية للأنظمة المطمورة. ولكن ليس من الضروري ان تكون هذه المعرفة موجودة للباحثين في مجال تصميم انظمة السيطرة المطمورة. لهذا السبب هنالك اهتمام كبير لتوظيف واستخدام طرق بديلة وحديثة لتصميم تلك الانظمة من الدون الحاجة الى المعرفة بلغة التصميم VHDL. إن هذا البحث يبين كيفية استخدام احد الطرق الحديثة في تنفيذ انظمة السيطرة بشكل سريع وباستخدام بيئة LABVIEW-FPGA. إن طريقة التنفيذ المادي للمسيطر المطمور باستخدام بيئة LABVIEW-FPGA قد تم توضيحها في هذا البحث بشكل كامل و اوضح بالاعتماد على المكون المنتج من قبل شركة Xilinx (Spartan 3e). وقد تم توضيح طريقة التصميم من خلال احدى تطبيقات السيطرة وهو السيطرة على درجة الحرارة وباستخدام الجهاز (CI-53003). ان النتائج العملية التي تم استحصالها تبين بشكل واضح نجاح التنفيذ المكون المادي للمسيطر المطمور.

Received: 22 – 6 - 2013 **Accepted: 2 – 10 - 2013**

1- Introduction

Recently an enormity of the research effort has been devoted in the field of hardware implementation using Field Programmable Gate Array (FPGA) technology. This is from the attractive features that offered by the FPGAs devices, such as flexibility, parallelism, re-configurability and their ability to re-programmability[1]. For this reason, it can be found that many applications are successfully utilized the FPGAs devices in their systems. As illustrative examples, digital signal processing applications, digital communications, image processing, and one of the prime interest in this work is in the field of embedded control systems [2], where cost and complexity are the major concern in these applications.

FPGAs are typically programmed using text based languages, Hardware Description Language (HDL). Currently, other alternative modern development environment is attractively used for rapid hardware implementation based FPGAs. Xilinx system generator Simulink by MATLAB environment and LABVIEW-FPGA by National Instruments based LABVIEW environment. These two high level languages provide the opportunity to easily and quickly develop the highly complex algorithms without or with little committing to hardware design[2]. Importantly, the complex systems can be evaluated, verified, tested and then implemented with significantly shortened time[3]. Furthermore, the trend now is to endorse these tools to be competitive as compared with text based languages in term of performance and hardware resources. Recent development in LABVIEW-FPGA promoted the researcher and engineers to investigate the facility and tools that offer by LABVIW-FPGA, where more than 25 libraries are added to the LABVIEW-FPGA module for various applications [4]. Recent research presents many productive researches based LABVIEW-FPGA tool.Al-Naami*et al*. [5]developed a technique to detect the aortic stenosis disease based on the compact RIO platform and using LABVIEW-FPGA module. The applicability of LABVIEW-FPGA in industrial control applications has also been established in several research[1, 3, 4, 6,7].This paper demonstrates the effectiveness and methodology of using LABVIEW-FPGA tool in the application of digital industrial control systems. Particular attention has been given to the embedded temperature control system; however the hardware implementation procedure is still transferable to the other system. This paper is organized as follows: the development flow of the LABVIEW-FPGA is presented in section II, the system description in term of hardware and software is outlined in section III, while section IV demonstrates the experimental validation of the hardware implementation. Finally, conclusion is presented in section V.

I. LABVIEW-FPGA Development Flow

In order to make the hardware implementation more manageable, it is essential to develop a systematic approach that can be followed in the design process. Fig.1 (a), shows that there are several steps to develop the hardware implementation using LABVIEW-FPGA tool[8]. In the first step, the developer should be evaluated the requirements of the system, as illustrative examples, the sampling time of the system, the number of the input and output channels, *etc*. In the second step, the targeting mode of operation of the real time system should also be determined, two modes are considered here: host PC interactive modeand interactive mode. Fig.1 (b, c), shows the block diagram of the two modes. In the host interactive mode, the developed code will be distributed between two VIs(Virtual Instrumentations). The main code will be synthesised to hardware and downloaded to the FPGA target. The second part is running under

PC and is coded using all the available functions in the normal LABVIEW environment. Thus, more features will be added to the process in term of floating point operations. While in the interactive mode, there is only a single VI to develop the algorithm. The block diagram panel of the single VI will be complied tothe FPGA target. It is important to note that, the front panels for both modes are running under PC for monitoring and control process. Based on the selected mode, the FPGA platform can now be chosen, different criterion is involved in this step, such as execution time, the size of the memory, number of multiplier, system interface, *etc*. Then, the FPGA VIs is created, here the FPGA module palettes are utilized to build the designed algorithms. Next, the prototyped VIs is compiled and run under PC for system simulation and test. If the designed VIs achieved the desired performance, then the final step is adapted, otherwise a more enhancement in the designed VIs is required. This cycle will continue till the desired output response is succeeding. Finally, the automatic step is executed without any contribution by the developer. Fig.1 (d), illustrates the sequence involve during this phase. It is important to mention that there is a direct relation between the LABVIEW panels and the text based language such as VHDL, where the front panel represents the entity in the VHDL and the block diagram represent the architecture of the VHDL[8].

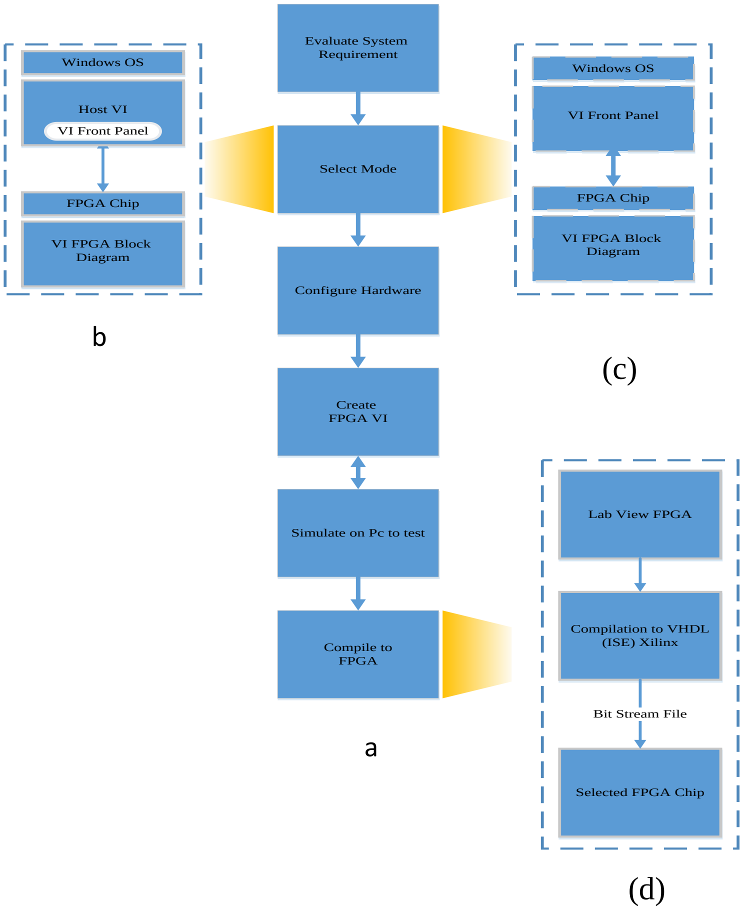

Figure (1): a) LABVIEW-FPGA development flow, b) Host interactive mode, c) Interactive mode, d) Compilation process

II. Hardware and Software Setup and System Description

The real time embedded control system based host PC consists into two parts (Fig.2). Here, the main role for the first VI (Host interactive block) is to acquire the output signal (sensed temperature signal), as well as is to send the action control signal to the process (PWM signal to the FAN). These operations are all implemented by the host PC based data acquisition card. The second VI (main VI) contains the main code that will be complied with the selected FPGA. It is imperative to emphasise that the design control system is working under real time operation, where both VIs are connected via a chain to transfer the data between the first and second VI.

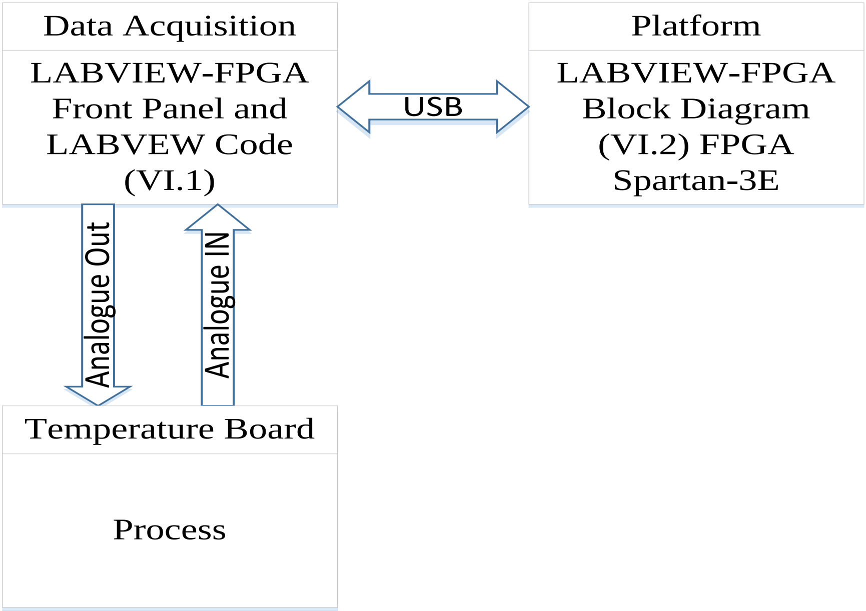
Data Acquisition
LABVIEW-FPGA Front Panel and LABVEW Code (VI.1)
USB
Platform
LABVIEW-FPGA Block Diagram (VI.2) FPGA Spartan-3E
Analogue Out
Analogue IN
Temperature Board
Process

Figure (2): Embedded host interactive mode

Fig. 3 depicts the overall block diagram of the real time control system. The design of the control system is also divided into two levels (Hardware level and software level). As shown in Fig. 3, the hardware level composes of three platforms. The first platform is the data acquisition card (DAQ) from the NI (PCI-6251).The analogue-to-digital converter (A/D) and the digital I/O of the PCI-6251 are directly connected to the second prototype system (CI-53003/ Temperature control module), to acquire the input/output signals. The operations of these devices are based on host-PC environment (first VI). Fig.4shows the wire connection of first VI. Signal conditioning circuits, such as AD590 (temperature transducer) is already embedded into the CI-53003, to sense the temperature signal. The sensed signal (feedback signal) is then passed to the DAQ. The converted signal via A/D converter is then transferred from the first VI to the second VI (LABVIEW-FPGA environment). Again, the second VI is represented the developed code that synthesis to hardware based FPGA. Fig.5 presents the second VI. Now, the FPGA, here XC3S500E / Spartan-3E board (third platform) will process the digital signal (loop error signal) via the designed control algorithm. Finally, the control action signal (PWM signal) is passed to the first VI to control the speed of the fan of the temperature module (CI-53003). The generated PW is directly transmitted to the CI-53003 through the DAQ device.

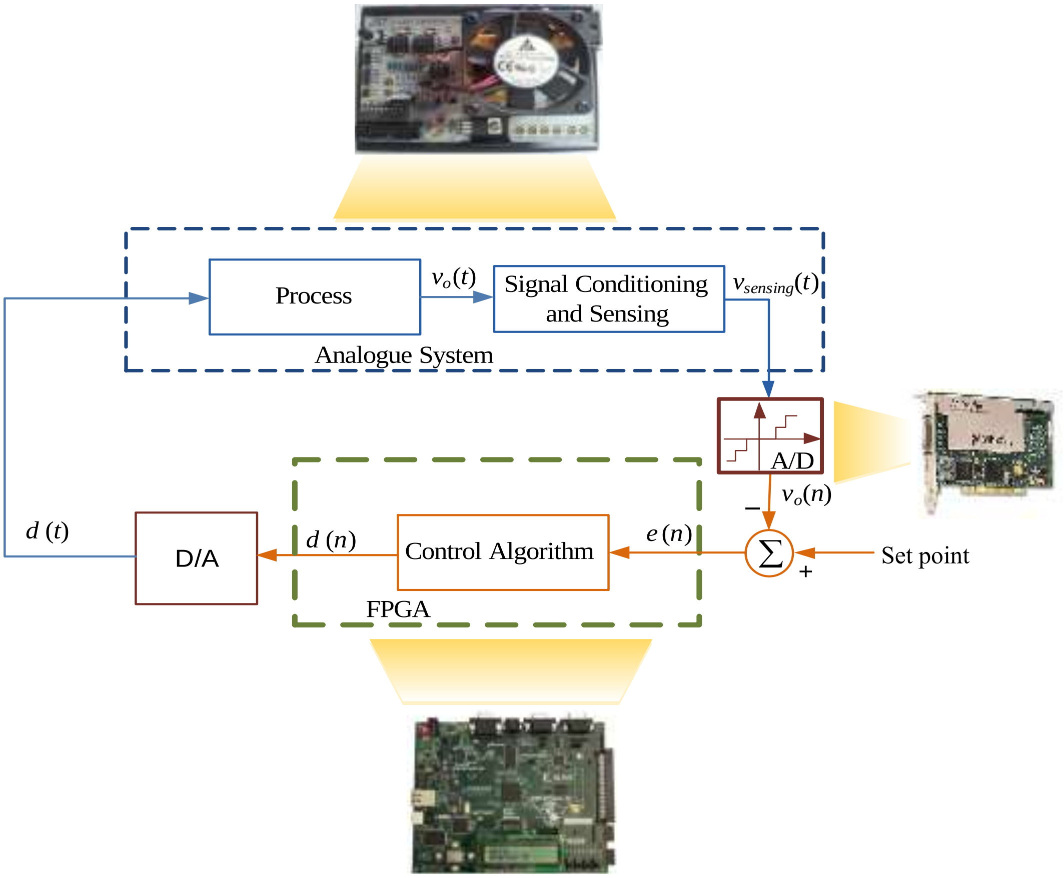

Figure (3): Real Time Feedback Control System

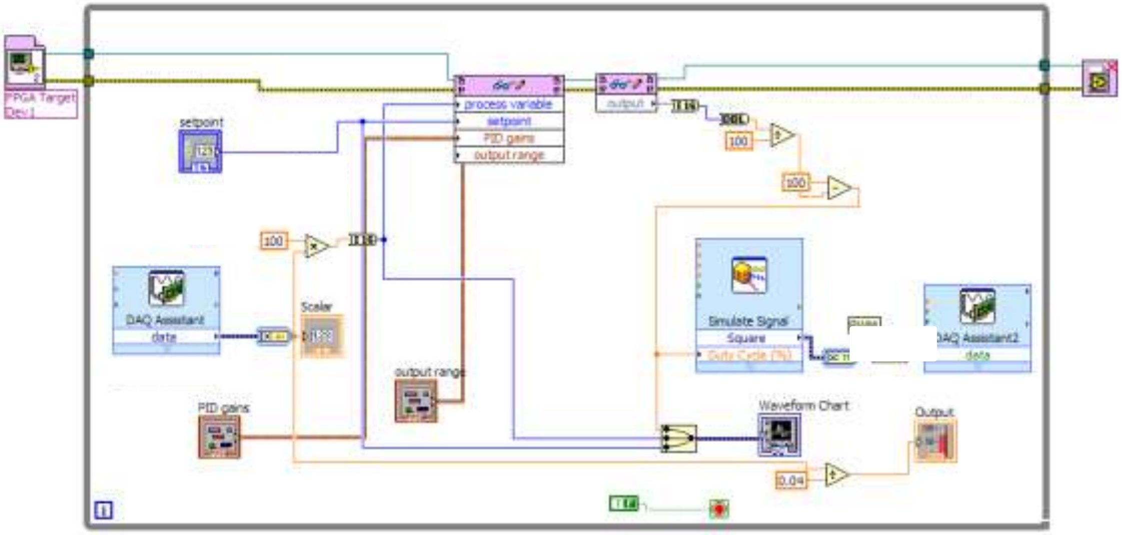

Figure (4): Schematic of the front Panel Based on Host PC

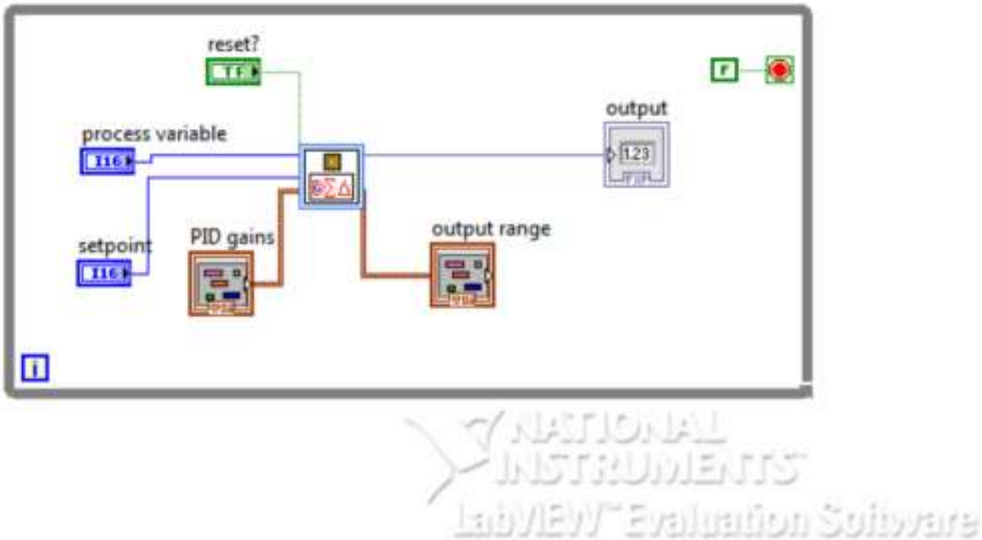

Figure (5): Block Diagram Panel Based on FPGA

III. Experiential Validation

This section illustrates the successful real time implementation of the design embedded control system using LABVIEW-FPGA module based Spartan-3E FPGA platform. Here, a digital PID controller is tuned using trial and error approach to regulate the temperature for a certain value. This can be accomplished by controlling the speed of the dc motor of the CI-53003 device. The PCI- 6521 DAQ is used to acquire the temperature signal and then to send the PW signal to the CI-53003 device. A gain, a PC host interactive mode has been utilized in this work.Now, to test the hardware implementation of the PID controller as well as to assess the overall dynamic performance, the reference signal; here the desired temperature signal is changing abruptly to different temperature level. It is imperativeto note that the characteristic of the temperature is not changing rapidly, therefore a small change in the temperature has been considered in the experimental results. This is clearly shown in Fig.6; here the temperature level is changed from 47-to-42 Celsius[1]. In addition, Fig.7 has shown the change of the reference signal from 40-to-47 Celsius. Therefore, the experimental results demonstrated that the designed controller has the ability to track the change in the temperature. Fig. 8 depicted the prototyped hardware setup. Finally, the hardware utilization for the digital PID-FPGA controller is presented in Table 1. It is clearly showing that a small amount of resources has been utilized in this block set (PID).

[1] Note: One Celsius is equal to 40 mv, here we multiply the converted value by 100, thus the real temperature value = sense temperature/4.

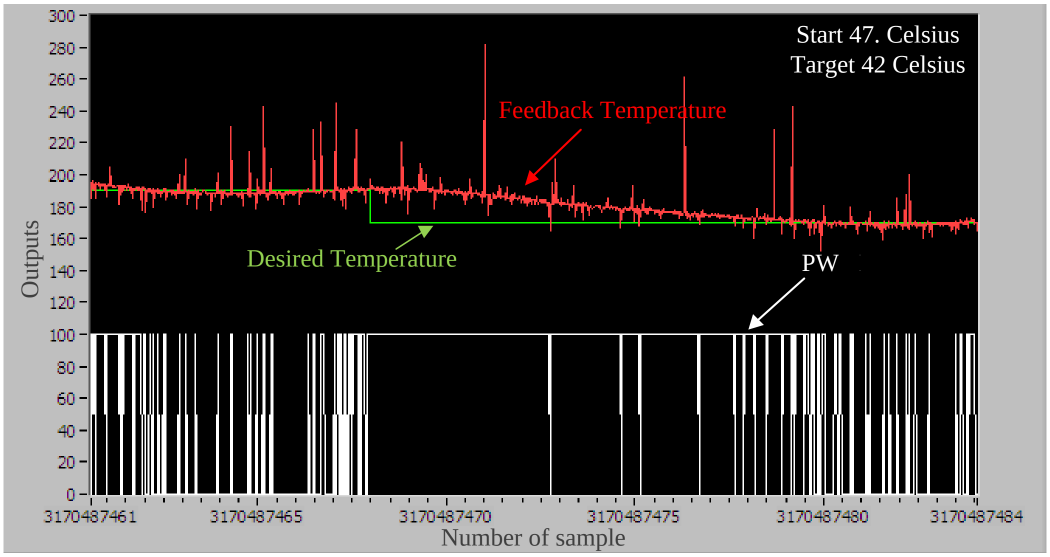

Figure (6): Response to the rapid temperature change from 47-to- 42 Celsius.

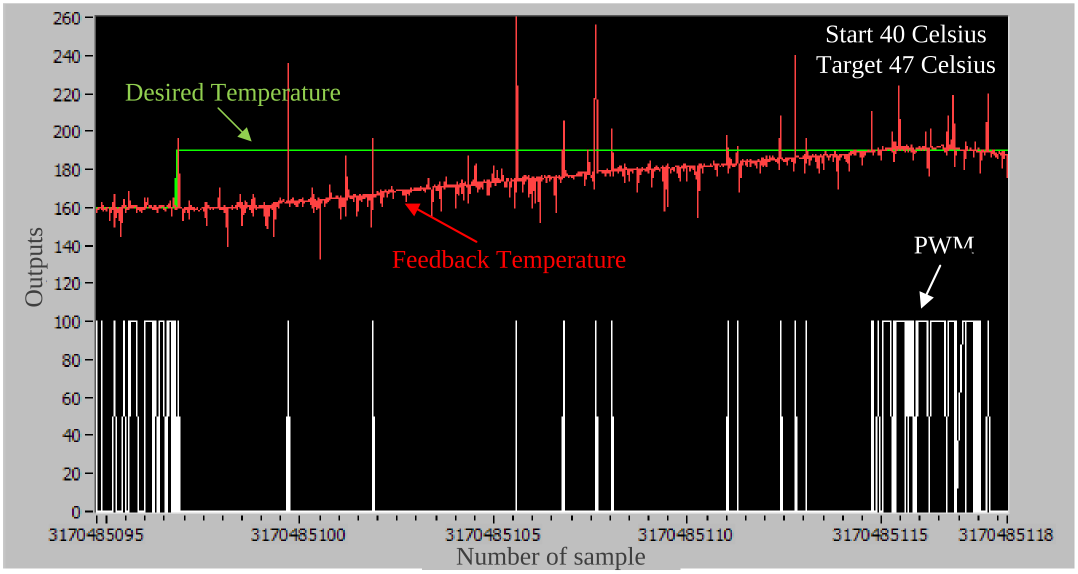

Figure (7): Response to the rapid temperature change from 40-to- 47 Celsius.

Table (1): Summary of the hardware resources for the single channel PID Controller

Device Type	**Percentage area occupied**
Number of Slices	1,080 out of 4,656 23%
Number of Slice Flip Flops	1,143 out of 9,312 12%
Number of 4 input LUTs	1,327 out of 9,312 14%
Number of bonded IOBs	84 out of 232 36%
Number of MULT18X18SIOs	3 out of 20 15%
The Achieved Maximum Frequency	53.720MHz

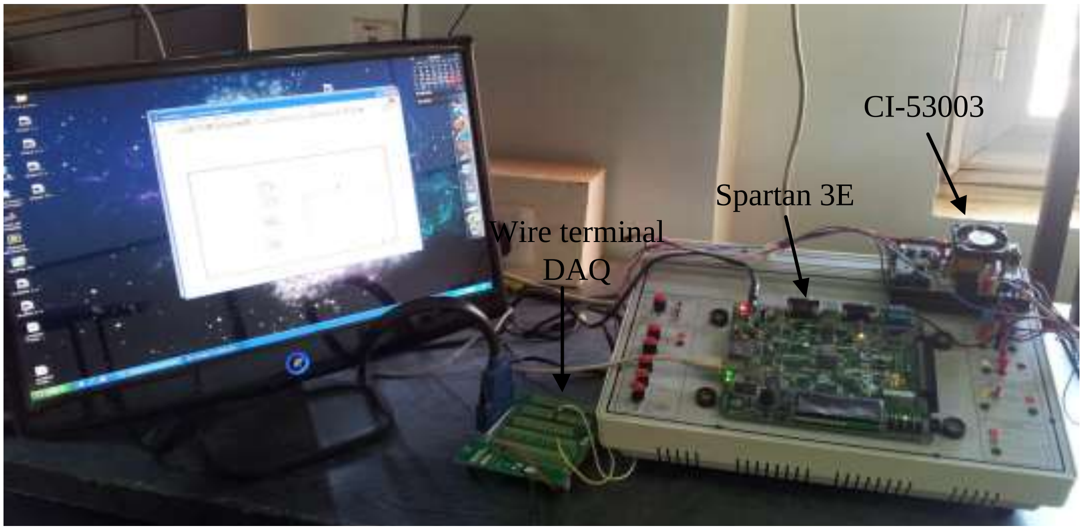

Figure (8): Prototyped hardware setup

IV. Conclusion

Text based languages is typically used to program the FPGAs, however this tool can become a barrier for rapid hardware implementation and for designing a truly real time control system. Therefore, this paper introduced an alternative method for rapid embedded control design using one of the superior modern tools, LABVIEW-FPGA from National Instruments (NI). The host interactive mode has utilized in the embedded system design, this including the acquiring of the analogue output signal, as well as the generating of the PW signal. The hardware implementation hasset up based on Sparatn-3E FPGA device. A temperature control system has selected to illustrate the procedure of embedded control design. The conventional digital PID controller is tuned and implemented to control the speed of the fan for successful temperature regulation. The methodology presented in this paper can be easily applied to many applications.Experimental results demonstrated the successful hardware implementation using LABVIEW-FPGAmodule.

References

[1] B. MacCleery and Z. M. Kassas, "New Mechatronics Development Techniques for FPGA-Based Control and Simulation of Electromechanical Systems", in *the International Federation of Automatic Control World Congress (IFAC 2008)*, 2008.

[2] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems-A review", *IEEE Transactions on Industrial Electronics*, Vol. 54, pp. 1824-1842, 2007.

[3] I. Catuna, E. Szopos, and M. D. Topa, "LabVIEW FPGA Implementation of Digital Reverberators", *Acta Technica Napocensis,* Vol. 51, 2010.

[4] W. Benrejeb and O. Boubaker, "FPGA modeling and real-time embedded control design via LabVIEW Software: Application for swinging-Up a pendulum", *International Journal on Smart Sensing and Intelligent Systems*, Vol. 5, pp. 576-591, 2012.

[5] B. Al-Naami, J. Chebil, B. Trabsheh, and H. Mgdob, "Developing custom signal processing algorithm with labview FPGA and compact RIO to detect the Aortic Stenosis disease", in *Computers in Cardiology, 2006*, 2006, pp. 193-196.

[6] M. L. Mears, J. S. Falcon, and T. R. Kurfess, "Real-time identification of sliding friction using LabVIEW FPGA", in *American Control Conference, 2006*, 2006, p. 6 .

[7] Y. Han, R. Tzoneva, and S. Behardien, "MATLAB, LabVIEW and FPGA linear control of an inverted pendulum", in *AFRICON 2007*, 2007, pp. 1-7.

[8] Natinoal Instruments. LABVIEW FPGA Module Training (May 2011). [online], Available: http://www.ni.com/white-paper/3555/en/.

The work was carried out at the college of Engineering. University of Mosul