ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

POSITION CONTROL OF DC SERVO MOTORS USING SOFT-CORE PROCESSOR ON FPGA TO MOVE ROBOT ARM

[1]A. Z. MANSOOR, [2]M. R. KHALIL, [3]O. A. JASIM

[1] Asstt Prof., Department of Electrical Technical Engineering, Technical College, Mosul, Iraq

[2] Asstt Prof., Department of Electrical Technical Engineering, Technical College, Mosul, Iraq

[3] Msc. Student, Department of Computer Technical Engineering, Technical College, Mosul, Iraq

E-mail: omareng_com@yahoo.com

ABSTRACT

A new method of controlling position of DC Servo motor using PicoBlaze soft-core processor on Field Programmable Gate Array (FPGA) is adopted. The soft processor is used to generate the PWM required to create motor motion and direction. The soft processor hardware system is build using VHDL, while the processor is programmed using assembly language. The type of control board used is XILINX FPGA SPARTAN-3E (XC3S500E). The system performance is debugged using chipscope technique and oscilloscope. The real time operation performance is done with five DC servo motors that are joined for testing motion of the robot arm. The results show that the position of the servo motors set was controlled efficiently.

Keywords: *Field Programmable Gate Array (FPGA),* Soft-core processor, *Very High Speed Integrated Circuit Hardware Description Language (VHDL), Pulse Width Modulation (PWM)*

1. INTRODUCTION

Servo motor is an Electro-mechanical device in which the electrical input signal controls the motor motion. It is designed for closed loop feedback systems. The output of the motor is coupled to a control circuit; as the motor turns, its speed and/or position are relayed to the control circuit. If the rotation of the motor is impeded for whatever reason, the feedback mechanism senses that the output of the motor is not yet in the desired location. The control circuit continues to correct the error until the motor finally reaches its proper point [1].

The input signal is pulse width modulation (PWM) in which the width of on pulse determines the position (angle) of the motor shaft and the amount of increment or decrement in the on pulse width determines the speed of rotation. The signal must be send every 20ms (50Hz).

The digital control techniques have predominated over other analog counts parts. The advantages of digital controllers are: Reconfigurability, Power saving options ,Less external passive components ,Less sensitive to temperature variation and High efficiency.

The available digital control hardware options include FPGA, Complex Programmable Logic Device (CPLD), Digital Signal Processor (DSP), and Micro Control Unit (MCU).

Micro-processor based control scheme have the advantages of flexibility, higher reliability and lower cost, but the demanding control requirements of modern power conditioning systems will overload most general purpose micro-processors and the computing speed of microprocessor limits the use of microprocessor in complex algorithms.

Present-day embedded systems use single-chip microcontrollers. Contemporary microcontrollers are available with 8-, 16- and 32-bit processing capability along with a peripheral set containing ADC, timer/counter and networks (I2C, CAN, SPI, and UART). For most applications the microcontroller-based board is adequate. For applications where there is a need to integrate custom logic for faster control and additional peripherals, the microcontroller or microprocessor board is augmented by a FPGA[2].

The FPGA is chosen because its circuit can be programmed easily in both hardware and software, Its capacity and speed compete and allow most functions in the control to be integrated on a single

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

chip, known as the System On Chip (SoC) design. It is compact and reliable, it is very low development costs, easy and very inexpensively to change designs. Considering all of its features, FPGA provides a good platform for controlling servo motors.

In this work, FPGA used to control the motion of the robot arm that contains five DC servo motors by PicoBlaze soft-core processor that generate pulse width modulation(PWM) signals to determine the position and speed of the motors.

2. LITERATURE REVIEW

Using FPGA based soft-core processor instead of a conventional architecture to control and drive high resolution motors like servos is a research area that has been receiving a considerable attention in the recent years. Soft-core processors can be used in various applications. A floating-point controller [3] and neuron-network for self-learning [4] have used original or customized PicoBlaze core for processing and controlling operation. The customizability and small size of PicoBlaze makes it an ideal soft-core processor for many applications. Controlling of DC servo motor [5], a robot hand [6], an autonomous mobile robot [7], a mobile robot[8] and an autonomous omnidirectional mobile robot [9] have used soft-core processor in the FPGA device for controlling operation. In this paper we proposed a controlling method of DC servo motors using PicoBlaze soft-core processor to move a robot arm in an efficient and accurate manner that saves the area of the FPGA device resources.

3. SERVO MOTORS

The type of DC servo motors used in this work is Hitec DC servo motors, five motors were used. The motors rotate to a limit up to 180 degree only in both clockwise and counterclockwise direction that is sufficient for robot arm free motion. These motors are more suitable for robot applications because of step accuracy, small size, light weight and high torque.

4. PICOBLAZE SOFT-CORE PROCESSOR

In this work a PicoBlaze soft-core processor that is a compact 8-bit soft core processor for Xilinx FPGA devices is used. It is provided as a cell-level HDL description (which is known as soft core) and can be synthesized along with other logic. PicoBlaze is optimized for efficiency and occupies only about 96 slices, which amount 2% of the resources in Spartan-3E (XC3S500E). It is compact and flexible and can be used for simple data processing and control, particularly for non-time-critical "house-keeping'' and I/O operations. The PicoBlaze processor can be easily integrated into a larger system and adds another dimension of flexibility in an FPGA-based design [10].

Many tasks can be done by either a customized hardware (VHDL) or a processor. The tradeoff is between the hardware complexity, performance and ease of development. There is no exact rule on which one to choose. Because developing software is usually easier than creating customized hardware, the processor option is generally preferable for non time critical applications. We can determine the feasibility of this option by examining the computation complexity. PicoBlaze requires two clock cycles to complete an instruction. If the system clock is 50 MHz, 25 million instructions can be performed in one second [10].

5. HARDWARE DESIGN

The PicoBlaze processor is fully designed by VHDL. It consists of KCPSM3 and single block RAM. Figure.1 illustrates the block diagram of picoBlaze processor.

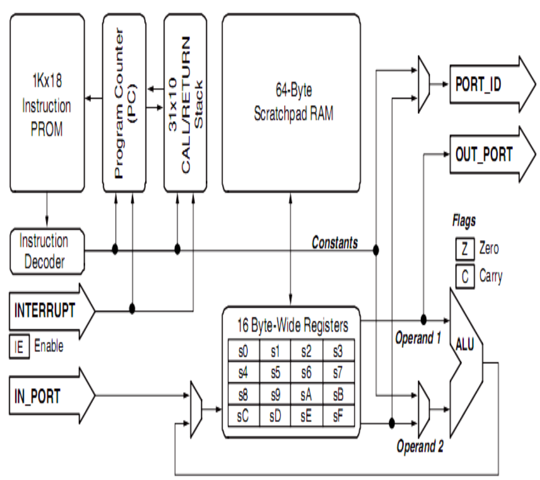

Figure 1. Block Diagram of PicoBlaze Processor with Single 1Kx18 Block RAM as the Instruction Store

The KCPSM3 is 8-bit processor connected together with single block RAM that is used to form a ROM store for a program of up to 1024 instructions [11].

Journal of Theoretical and Applied Information Technology
15th October 2011. Vol. 32 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure.2 illustrates the hardware design. The input and output decoders are designed outside the processor using VHDL language to check the port address, read strobe signal and write strobe signal are coming from the processor to determine the input and output port.

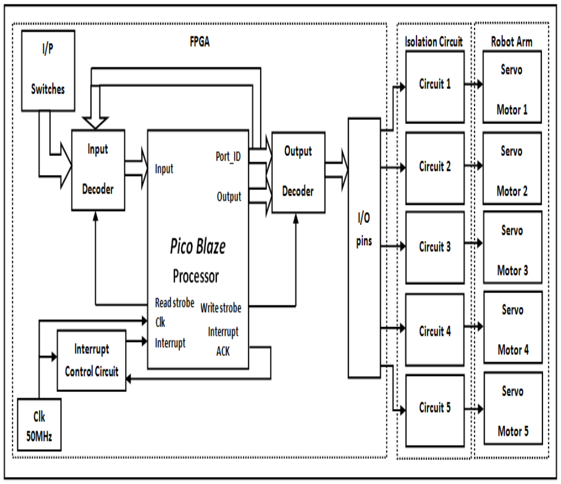

Figure 2. Block Diagram of FPGA Hardware Design and Overall Work

The interrupt control circuit is designed to interrupt the PicoBlaze. This circuit contains up counter to count from 0 to 499 and comparator to compare the counter value with 499 ,so this will interrupt the PicoBlaze every 500 clock. Since the system clock is 50MHz, this means that the interrupt occur every 10µs.

Interrupt period = Max. counter value/system clock

6. SOFTWARE POSITION CONTROL

The controller must generate a continuous pulse width modulation(PWM) for the DC servo motor at 20ms (50 Hz). The duration of the on pulse (logic high) determines the position of the servo shaft, the range of duration between 0.5ms (0º) and 2.5ms (180º). A short on pulse (0.5ms) moves the servo to the far left, while a long on pulse (2.5ms) moves the servo to the far right. Intermediate length pulses (1.5ms) correspond to intermediate positions. Figure.3 illustrates the timing constraints of the control signal. Since the rotation limits to 180º and the duration range is 2ms (2.5ms – 0.5ms=2ms) , so the amount of movement for each one degree equal 0.01111ms (11.1111µs).

Movement (1º) = 2ms/180º = 0.01111ms (11.1111 µs)

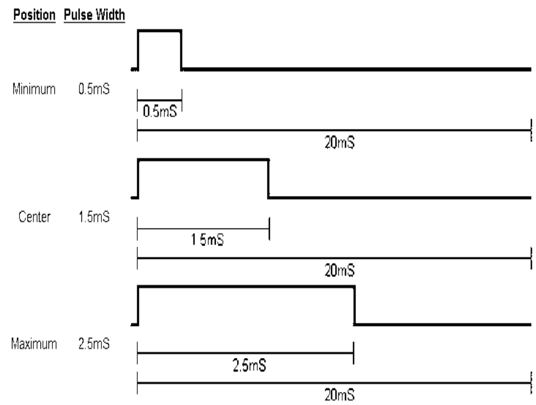

Figure 3. Timing Constraints of The Control Signal

7. SOFTWARE SPEED CONTROL

Speed is measurement of the time that the servo take to rotate a certain number of degrees. It is specified in degree/second.

The software speed control of the servo motor is determined by the amount of increment or decrement of an on pulse (logic high) between current on pulse and next on pulse.

If the amount of increment or decrement is too small (1µs = 0.09º), the motor will rotate very slow at minimum speed ,and if the amount of increment or decrement is too large (equal to the amount of desired position), the motor will rotate at maximum speed.

8. SOFTWARE PROGRAM

The PicoBlaze processor is programmed using assembly language that make it very easy and flexible for development. The instructions are stored in the single block RAM 1KX18. Because the processor execute the instructions sequentially and the PWM requires continuous pulses, interrupt should be used to generate continuous pulses. The resolution of PWM is 16-bit because interrupt event occur every 10µs and the total pulse repeated every 20ms (50Hz) so the software counter should has a value of 2000 (7D0H) that is 16-bit.

software counter = 20ms/10µs=2000

If the interrupt occurs every 1µs, this means that the processor should execute both interrupt service routine (ISR) and main program in 1µs for each.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Because PicoBlaze execute each instruction in 2 clock cycle and the system clock is 50MHz, so in 1µs it can execute 25 instructions only and this is too little. To control five servo motors to move the arm freely we need about (200-250) instructions to be executed. Execution of 250 instructions need 500 clock, so the interrupt should occur at least every 500 clock that mean 10 µs. This is sufficient resolution for our work. Figure.4 illustrates a flow chart of the software program. As it was mentioned, the minimum speed at 1µs(0.09º) . so at 10 µs : 10 * 0.09º = 0.9º. The minimum speed at 10 µs equal 0.9º.

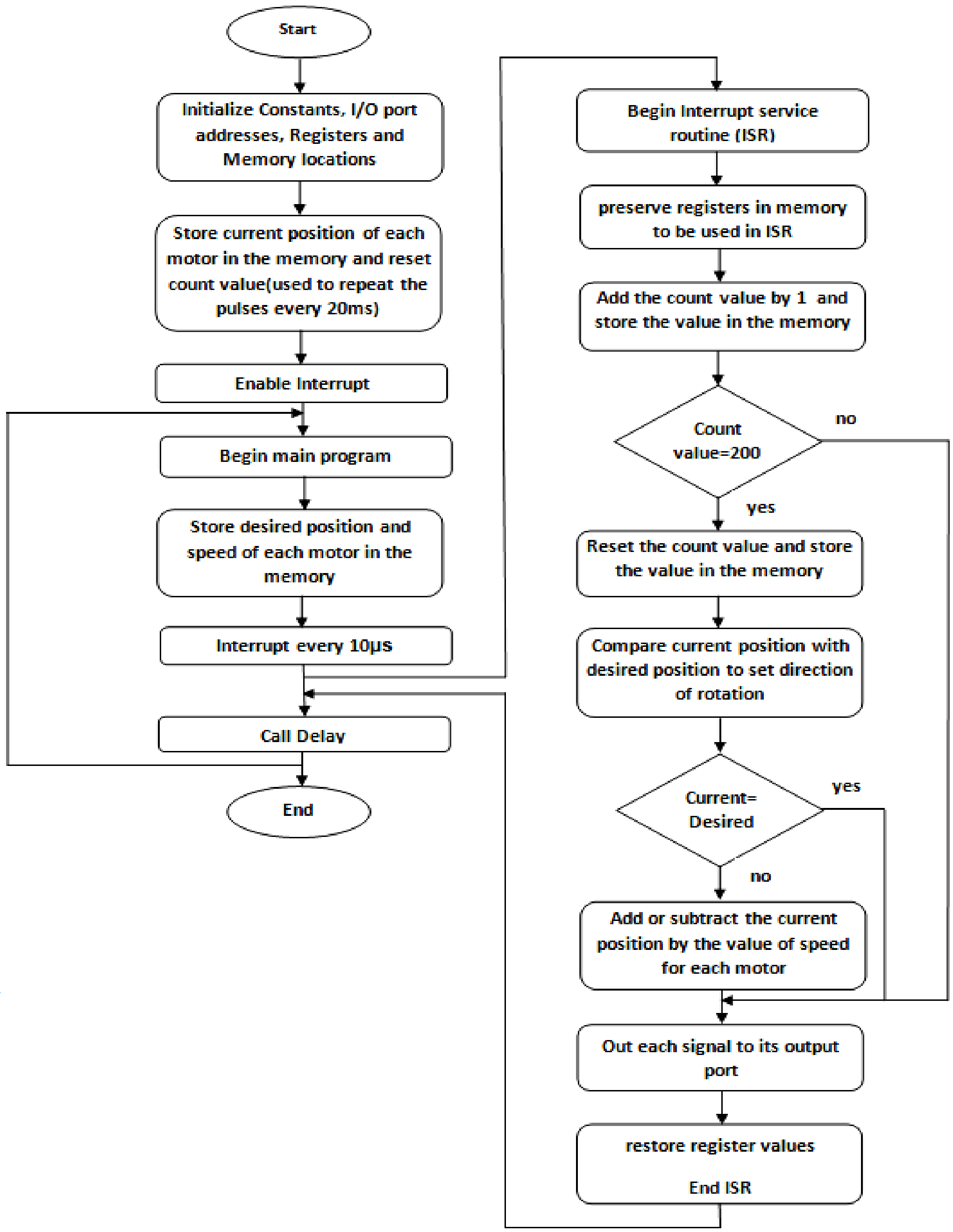

Figure 4. The Flow Chart of The Controlled Program

Journal of Theoretical and Applied Information Technology
15th October 2011. Vol. 32 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

9. ISOLATION CIRCUIT

The isolation circuit is important to isolate the FPGA I/O pins from being overloaded by excessive currents and to be isolated from the high voltage coming back from servo motor drive circuit because the motor voltage and current are higher than FPGA. This will provide safety for FPGA board from any high current and voltage reverse back to the board. The circuit is illustrated in figure.5.

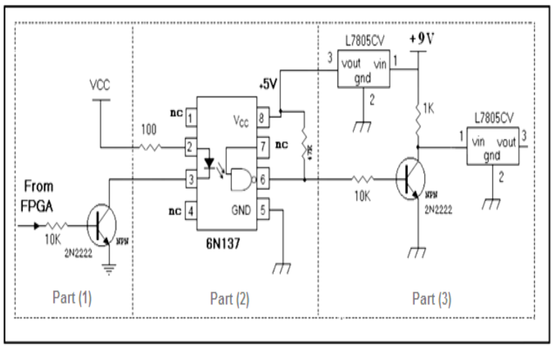

Figure.5 Isolation Circuit Diagram for One Servo Motor

It consists of three parts: part (1) used to protect the FPGA pins, part (2) is an optocoupler circuit that make the voltage level isolation, and part (3) which gives the appropriate level of voltage to the servo motor. The output of the circuit is connected to control signal for one servo motor, so it needs five isolation circuits one for each motor.

10.ROBOT ARM

The robot arm has 4 degree of freedom(4DOF). The arm uses five servo motors: 1 x HS-475HB in the base, 1 x HS-755HB in the shoulder, 1 x HS-645MG in the elbow, 1 x HS-475HB in the wrist, and 1 x HS-422 in the gripper.

11. RESULTS

Designing a control system using conventional architecture of hardware peripherals elements (designing by VHDL) or a soft-core processor in conjunction with hardware peripherals elements as explained in [13] and [14] to generate PWM signals takes a lot of resources from the FPGA device. So using compact soft-core processor like PicoBlaze to generate PWM signals without using additional hardware peripherals elements makes the design more reliable and efficient in reserving resources of the device when compared with conventional design since the designed system occupied only 4% from the FPGA Spartan-3E(XC3S500E) slices, additionally, the designed system in the FPGA explores the generation of PWM using software (assembly language) executed by PicoBlaze rather than using hardware peripherals circuits to generate PWM (designed by VHDL). This improves the dynamics of the PWM which are totally flexible and customizable PWM parameters by software definition. The position control with PWM technique is debugged in real time by using ChipScope software and oscilloscope. Figure.7 and 8 shows the results obtained using ChipScope software. Figure.7 display that the pulse width for each motor, for example is 1.5ms in "pwm_ch_2". While figure.8 displays that the period time of the "on" and "off" is 20ms. Figure.9, 10 and 11 present the results obtained by the oscilloscope. Figure.12 shows the overall structure of the designed control system . The results demonstrate that the position control of the servo motors is accurate and efficient.

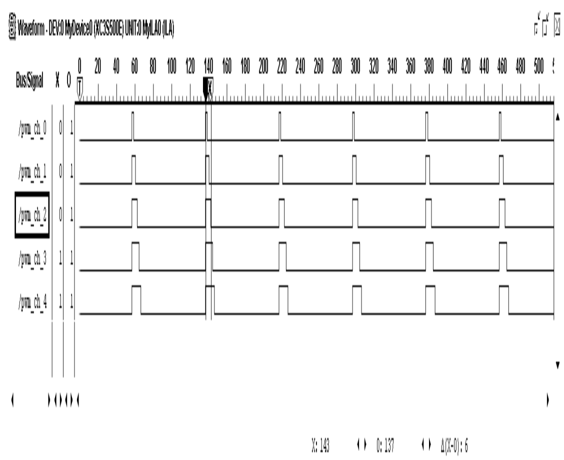

Figure.7 Practical Chipscope Results of On Pulse time

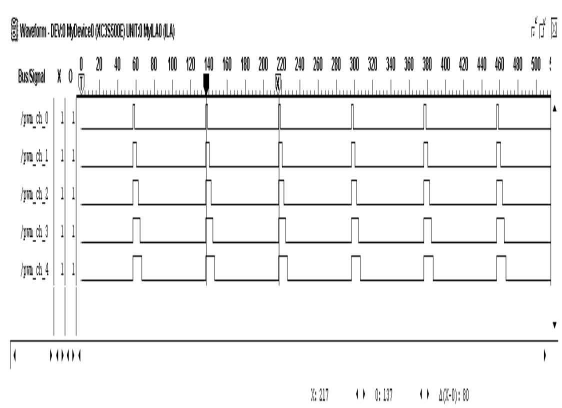

Figure.8 Practical Chipscope Results of On and Off Pulse Period

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

12. CONCLUSION

This paper demonstrated a hardware and software co-design of a controller that used to control a robot arm with five servo motors. PicoBlaze soft-core processor (controller) was used to control the position and speed for the servo motors by the PWM signals that were generated using processor interrupt. This advantage makes the system very convenient since it allows the processor to perform another tasks with the controlling of servo motors at the same time. Additionally, the system can be expanded simply in a case of needing to increase the number of motors or needing additional processing operations by adding another PicoBlaze processors to the design that make the system very efficient and flexible. An accurate position control of servo motors jointed to a robotic arm system was achieved successfully using PicoBlaze soft-core processor configured on FPGA Spartan-3E slices employing pulse width modulation (PWM) technique. Future work will focus on enhancement of drive control tasks and further application development by configuring multiple PicoBlaze processors system to control more number of motors and using a reference clock more than 50MHz with 16-bit resolution and decreasing the interrupt period to less than 10µs with attention to the dead band width period of the motors to increase accuracy of the servo shaft position.

REFRENCES:

[1] G.Mccomb and M.Predko ,"Robot Builder's Bonanza",2006 ,pp.393-413 .

[2] R. Dubey ,"Introduction to Embedded System Design Using Field Programmable Gate Arrays" ,2009 , pp. 1-14.

[3] J. Kadlec and R. Gook, “Floating point controller as a PicoBlaze network on a single Spartan 3 FPGA,” in MAPLD05, 2005.

[4] J. A. Starzyk, Y. Guo, Z. Zhu, “Dynamically reconfigurable neuron architecture for the implementation of self-organizing learning array,” in Proceedings of the 18th IPDPS, 2004.

[5] A. Das, K. Banerjee, "Fast Prototyping of a Digital PID Controller on a FPGA based Soft-Core Microcontroller for precision control of a Brushed DC Servo Motor" , IEEE , 35th Annual Conference of IEEE Industrial Electronics, 2009, pp. 2825-2830.

[6] Y.W.Liu, M.H.Jin, R.Wei, H.G. Cai, H. Liu, N. Seitz, R. Gruber, G. Hirzinger, “Embedded FPGA-based Control of the HIT/DLR Hand ”, IEEE/ASME, International Conference on Advanced Intelligent Mechatronics, Monterey, California, USA, 2005, 24-28 July, pp. 582-587.

[7] M. S. Lateef, “Design and Implementation of FPGA Based Adaptive Neural Controller for Intelligent Robot Navigation”, 2011, Msc. thesis, Engineering College, University of Mosul, Iraq.

[8] D. F. Wolf, J. A. Holanda, V. Bonato, R. Peron, E. Marques, “An FPGA Based Mobile Robot Controller”, IEEE, 3rd Southern Conference on Programmable Logic, 2007, pp. 119-124.

[9] H.C. Huang and C.C. Tsai, “FPGA Implementation of an Embedded Robust Adaptive Controller for Autonomous Omnidirectional Mobile Platform”, IEEE Transactions on Industrial Electronics, 2009, vol.56, iss.5, pp. 1604-1616.

[10] P.P.Chu - Cleveland State University, "FPGA Prototyping by VHDL Examples" , 2008, pp.323-418.

[11] Xilinx Inc. - UG129 (v2.0) , " PicoBlaze 8-bit Embedded Microcontroller User Guide ", January 28, 2010, pp.11- 124.

[12] K. Chapman, " KCPSM3 8-bit Microcontroller " , October 2003, Xilinx Inc.,pp.4-75 .

[13] K. Chaplin and K. Chapman, "JTAG Loader", 2004, Xilinx Inc., pp.4-10.

[14] P. R. Wilson ,"Design Recipes for FPGAs" , 2007,pp.57-79.

[15] V. A. Pedroni, " Circuit Design with VHDL ", 2004, pp.1- 363.

[16] Xilinx Inc., " Spartan-3E FPGA Family: Complete Data Sheet" , April 18, 2008,pp.1-234.

[17] Xilinx Inc. - UG230 (v1.1) , " Spartan-3E FPGA Starter Kit Board User Guide", June 20, 2008, pp.1- 166.

[18] Xilinx Inc. - UG029 (v10.1), "ChipScope Pro 10.1 Software and Cores User Guide", March 24, 2008,pp.1-194.

[19] Xilinx Inc. - XAPP477 (v1.0.1), " Embedded Processing and Control Solutions for Spartan-3 FPGAs" , August 11, 2003, pp. 1-4.

[20] W. Hongxing, L. Chengcheng, C. Diansheng, W. Jun, W. Tianmiao, "Research on Reconfigurable Robot Controller Based on ARM and FPGA" , IEEE International Conference of Industrial Informatics (INDIN), DCC, Daejeon, Korea, 2008 July 13-16, pp. 123-128.

Journal of Theoretical and Applied Information Technology
15th October 2011. Vol. 32 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

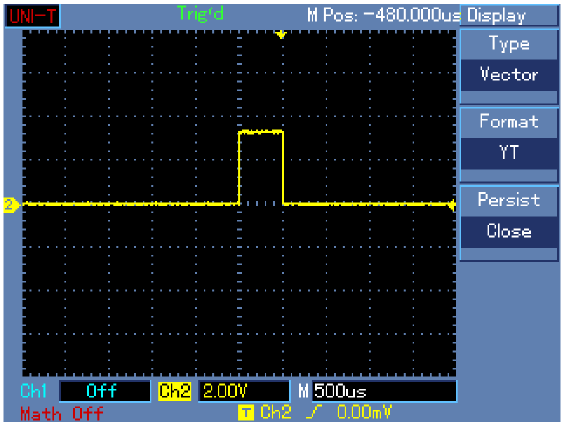

Figure.9 Practical oscilloscope results for the pulse that moving the motor to position (0º) with a pulse width of 0.5ms

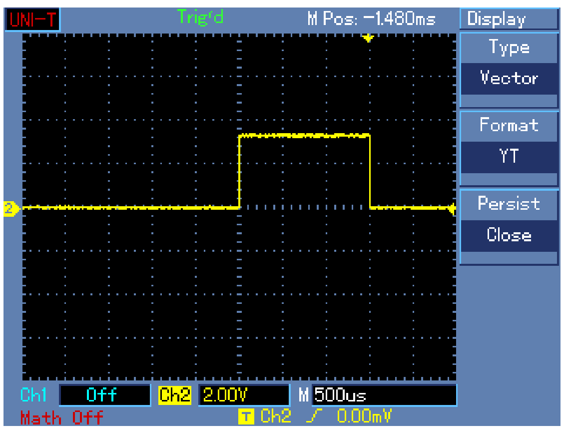

Figure.10 Practical oscilloscope results for the pulse that moving the motor to position (90º) with a pulse width of 1.5ms

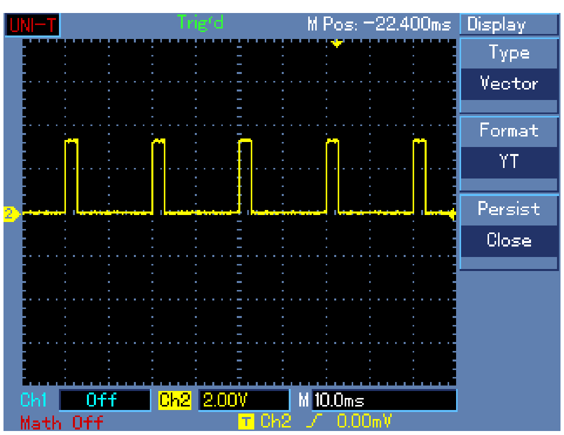

Figure.11 Practical oscilloscope results for a train of pulses that moving the motor to position (180º) with a pulse width of 2.5ms and the pulse repeated every 20ms


Figure.12 Overall work

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AUTHOR PROFILES:

Omar A. Jasim received the Bsc. degree in Technical Computer Engineering from Technical College of Mosul, Iraq, in 2005. Currently, he is a research student of master degree in Technical College of Mosul, Iraq. His interests are in control system design, Robots design and computer networking.

Mazin R. Khalil received the master degree in Electronic Engineering from university of Mosul, Iraq, in 1995. He is an Associate Professor at Technical College of Mosul, in Iraq. He published 20 scientific researches and two books. His interests are in control system design.

Dr. Abdul-kareem Z. Mansoor received the degree of the PH.D in Power Electronic Engineering from college of Automation-Technical university, Bulgaria, in 1993. Currently, he is an Associate Professor at Technical College of Mosul, in Iraq. His interests are in power system control design and motors control and drive.