FPGA Based Robotic Arm With Six Degrees of Freedom

Shri Lakshmi Pravalika
Student, Department of Electronics and Communication Engineering
Saveetha Engineering College, Chennai, Tamilnadu, India

Dr. Alex Noel Joseph Raj
Associate professor, Department of Electronics and Communication Engineering
Saveetha Engineering College, Chennai, Tamilnadu, India

Abstract- The purpose of the project is to build a multipurpose robotic arm which can perform operations with higher accuracy with 6 degrees of freedom. The arm will exhibit higher speed of operation and lower delay in processing due to the use of FPGA for processing and control. This will improve speed of operation and reduce the latency of response when used for applications such as bionic arm when compared to present methods.

Keywords – FPGA, Robotic Arm, Degrees of freedom

I. INTRODUCTION

A robotic arm is a mechanical arm, which can be autonomous or controlled manually, having multipurpose manipulator programmable in three or more axes and can be used to perform a variety of tasks with great accuracy and speed. The ROBOTIC ARM with six Degrees Of Freedom (DOF) as that of a human arm can perform all tasks (including the tasks performed by a human) with ease and comparatively faster, simpler with fewer movements. In the present scenario Robotic arm has fewer degrees of freedom typically less than five, but for a successful replication of a human ARM we require 6 DOF[1][2]. The replication of a human arm would be ideal to use it as a bionic arm and to perform other industrial automations in a simpler manner.

In our project we constructed a Robotic ARM with servo motors. The selection of the servo motors was due to their ability to move to a finer angle, High Torque during loaded and unloaded conditions versus DC motors and stepper motors have low holding torque with less precision. These attributes made the servo motor an effective choice for the construction of the ARM. In the project a user interface (GUI) (front end GUI) was created using VB.NET which is used to control the ARM. The GUI contains a slider control for the control of the servo motors. On moving the slider in the windows form, corresponding signals are sent in order to move the servo motors. The VB.NET interface is responsible for sending the data to the microcontroller (MC) via serial port. The data is at first encoded and then start and stop bits are added with check bits to ensure the reliability of data. Then the data is received by the microcontroller, where it is decoded and checked for reliability of transmission using check bits, start and stop bits.The Micro controller has limitations regarding the number of PWM channels it can generate. Due to the requirement of several PWM channels, another method was adopted to resolve this issue. A Xilinx based SPARTAN 3 FPGA (Field Gate Programmable Array) was used to resolve our limitations [3]. The flexibility of the FPGA alongside with the enormous number of ports with programmable gate arrays which could give multiple PWM channels was an ideal choice for end implementation.

The decoded information is converted into digital data and is updated along with the address. The FPGA reads the data and then, generates the PWM and updates the value to the corresponding joint motor. The FPGA then checks for the PWM pulses by taking an internal feedback from the generated data's and then checks the values with pre defined values. It then sends the feedback (ACK packet) to the MC which then forwards it back to the GUI for successful validation. The robot also features isolators which are used to isolate the MC and the motor drivers to avoid reverse voltage protection to the FPGA and other circuits. The signal from the isolation circuits are amplified to the desired voltage level and sent to the motors.

The entire ARM is constructed on a ROBOT which can be moved via positioning control through GUI [2]. Electromagnetic Interference produced by the motors affected the entire electronics. Steps such as twisted pair cables were replaced by shielded twisted pair cables and all electronic equipment was covered with electrostatic shield. The ROBOT offer the following two modes of operation

- Automatic Mode - In this mode the ROBOT and the ARM can be pre-programmed to perform certain task or tasks in a sequential manner without any human intervention.
- Manual Mode - In this mode the ROBOT can be operated via Wi-Fi or through Internet [5][6].

In the Automatic mode, the ROBOT features a camera, which is used for image processing, for visual feedback and motion control. The presence of dual cameras helps the ROBOT for stereoscopic view and thereby helps to find an approximate estimate of distance between objects. It also helps in determining the distance from the ROBOT to the object.

II. GRAPHICAL USER INTERFACE (GUI)

The Windows application in Microsoft Visual studio was selected for the GUI due to its ability to easily support Input/output operations via serial port (RS232).In the manual mode the GUI accepted user instructions via slider and then converted into suitable data. To ensure reliable transmission the packet transferred contains the following attributes.

START	HEADER	DATA	PARITY	FOOTER	END

START- Indicates the start of transmission.
HEADER - Contains information regarding PWM Data or Address.
DATA- Contains the PWM DATA or ADDRESS.
PARITY- It contains the parity element.
FOOTER- Contains the length of the packet.
END- Indicates the end of transmission.

To move a motor to 180 degrees, a data of 255 must be sent from the computer to the MC in order to generate the PWM signals. Each motor is assigned with a specific ADDRESS. For example if a motor has its ADDRESS as ‘6’. The header information of ADDRESS “6” is added with the data 255. Now the parity bit is added with the data. Then the length of the data is calculated and the footer information is added in the footer field. Then the START and END fields is added to the packet. The microcontroller on receiving the packet splits the individual fields and determines the data and checks for parity and verifies the length of the data and the correctness of Data. On successful verification an Acknowledgment Packet (feedback) is sent to the User Interface. The User interface then waits for the Acknowledgment packet from the FPGA. If the FPGA receives the data and generates the exact PWM signal then an ACK packet is sent to the GUI for verification. Thus the validation of data and its successful implementation is performed and the result is updated in the GUI.
On failure of receiving the data, a NACK packet is sent by the corresponding device (MC or FPGA) to the GUI. On receiving such a packet the GUI attempts to send the ADDRESS and data again for a sequence of 5 attempts after which it reports an error to the End User for probable connection problems.

In the Automatic mode the GUI is pre-programmed to do specific functions and has minimal no of interactions with the END USER. The presence of dual camera helps in stereoscopic vision as well as gives a view of the surroundings of the robot. In the Manual mode it helps the user to operate the ARM. In the Automatic mode, each and every operation obtains feedback through the hardware. Visual feedback is obtained only when programmed or for tasks requiring extreme precision.

III. MICROCONTROLLER

Microcontroller used in the Robot for processing and control of the DATA and ADDRESS packets was Atmega16. The Atmega 16 is an AVR family micro controller produced by ATMEL. They have considerable processing power and enough memory and are easy to program. It has separate buses for program and data memory for faster access. The features of Atmega 16 were ideal for the intended purpose. The micro controller communicates with the user interface via serial port, with a baud rate of 9600bps. The micro controller is responsible for the communication between the user interface and the FPGA.

The MC on receiving the packet from the GUI splits the packet into header, data, parity, footer. It uses the start and end fields to recognize the start and end of the transmission. Then the header field information is used to determine whether the received data is either ADDRESS or PWM DATA. Then the parity field and footer field is used to verify the correctness of data. On successful validation an ACK packet is sent to the GUI. The fields of the ACK packet contain.

START	ACK	END

START –Indicates the start of transmission.
ACK –Contains ACK of the micro controller or FPGA in case of feedback from FPGA.
END –Indicates end of transmission.

When the data cannot be verified, a NACK packet is sent. The NACK packet contains

START	NACK	END

START - Indicates the start of transmission.
NACK - Contains NACK of the micro controller or FPGA in case of feedback from FPGA.
END - Indicates the end of transmission.

The MC after successful validation of the packet, sent by UI, sends the ADDRESS to FPGA through a parallel bus. It then waits for the FPGA to send an ACK signal. On successful reception on ACK signal it updates the Bus with the data value. The FPGA reads data value and then computes and implements the PWM data and checks and verifies the feedback. If updated it sends an ACK signal or else a NACK signal is sent. On receiving a NACK signal from FPGA, the microcontroller retransmits the data via bus for 5 trials. On failure to obtain an ACK signal after 5 trials, will lead to the transmission of a NACK packet from the MC to the GUI. On successful reception of the ACK signal from FPGA an ACK packet is sent. The feedback ensures the implementation of the PWM signal, and also helps to debug the problem in a simple manner.

IV. FIELD PROGRAMMABLE GATE ARRAY

4.1. Overview-

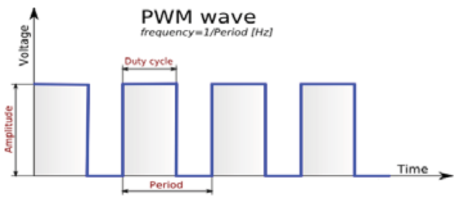
A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by the user. The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC)[1]. FPGA can be used to implement any logical function that an ASIC could perform. The ability to update the functionality after shipping, partial of a portion of the design and the low non-recurring engineering costs relative to an ASIC design, offer advantages for many applications.

4.2. SPARTAN 3-

The Spartan-3 Starter Board provides a powerful, self-contained development platform for designs targeting the new Spartan-3 FPGA from Xilinx [3]. It features 200,000 gates Spartan-3, on-board I/O devices, and 1MB fast asynchronous SRAM, making it the perfect platform to experiment with any new design, from a simple logic circuit to an embedded processor core. The board also contains a Platform Flash JTAG-programmable ROM, so designs can easily be made non-volatile. The Spartan-3 Starter Board is fully compatible with all versions of the Xilinx ISE tools, including the free Web Pack. The board ships with a power supply and you can add a programming cable at checkout.

4.3. PWM-

Pulse-width modulation (PWM), as it applies to motor control [4], is a way of delivering energy through a succession of pulses rather than a continuously varying (analog) signal. By increasing or decreasing pulse width, the controller regulates energy flow to the motor shaft. The motor's own inductance acts like a filter, storing energy during the "on" cycle while releasing it at a rate corresponding to the input or reference signal. In our case the servo motors which should be made to rotate according to desired angle is controlled using PWM signal. PWM signal used for controlling servo motors has constant pulse width of 20ms (50 Hz) whose ON time is varied according to desired position. For the entire servo motors PWM width is constant. Initially motor shaft is adjusted to 90 degrees and forward and reverse movements are controlled by slider application in visual basic.



Duty cycle = T_{on} / Time period

Figure 1. PWM Waveform

Degree of rotation (deg)	ON time (ms)
0	0.70000
1	0.70625
2	0.71250
3	0.71875
.	.
.	.
90	1.5000
.	.
.	.
180	2.3000

Table 1. ON Time for each Degrees of rotation

Following are ON time associated with degree of rotation,

ON time for zero degree position of shaft: 0.7ms (3.5% DC), ON time for 180 degree position of shaft: 2.3ms (11.5% DC). The range (0.7ms to 2.3ms) is divided by 256 (8 bit output) outputs from slider.
Initial position (downwards) = 0 (00000000) = 0.7ms. Intermediate position (middle) = 128 (10000000) = 1.5ms.
Final position of slider (upwards) = 255(11111111) = 2.3ms
Therefore, (2.3-0.7)*10^-3/ 256 =6.25us, so for 1 degree movement, ON time should be incremented by 6.25us

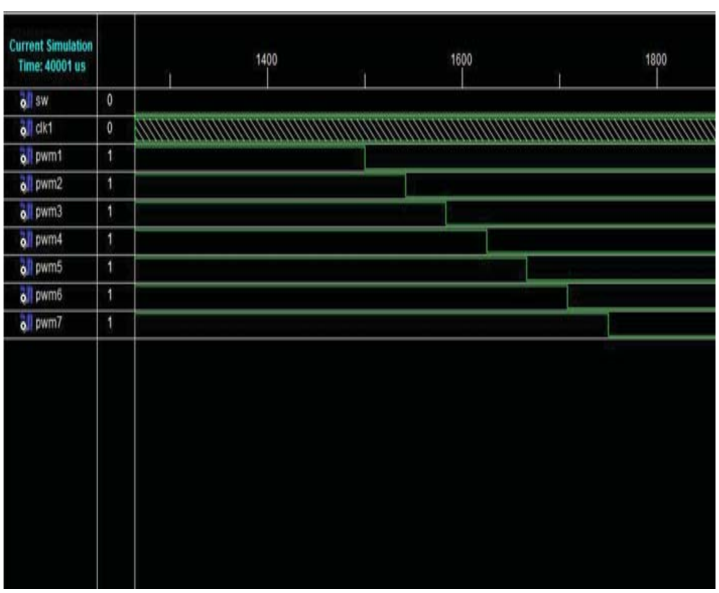


Figure 2(a). PWM Waveform generation in FPGA

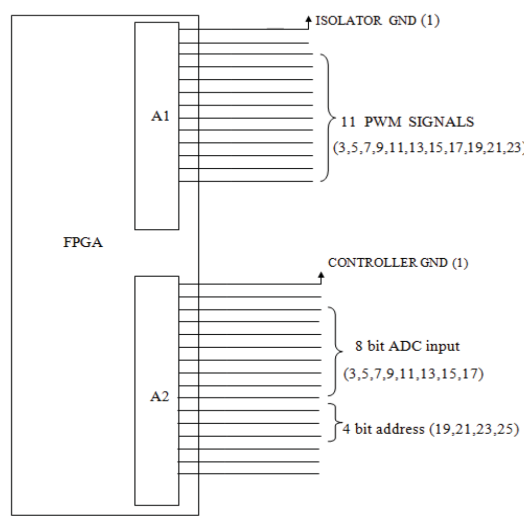


Figure 2(b). Connection diagram with FPGA

The various PWM signals are generated for each individual motors is shown in Figure 2(a). The PWM signals in the Figure 2(a) indicate their various ON and OFF times mentioned in the Table1. Hence for each degree a corresponding PWM signal is generated. Figure 2(b) illustrates the connection diagram of the FPGA.

4.4. Operations in FPGA-

The FPGA receives the data and address through the bus. The FPGA sends an ACK signal for each received ADDRESS and it also sends an ACK signal after each successful implementation of PWM DATA received to the MC. It sends an NACK after failure of implementation. The FPGA uses the PWM DATA and ADDRESS to update the PWM value generated for that address. For instance, a PWM value of 255 on updating channel 10 (Previous PWM value 0) would result in a 180 degree turn by the Servo motor on channel 10. Thus to change the angle of the Servo motor we need to change the PWM value in the DATA field.

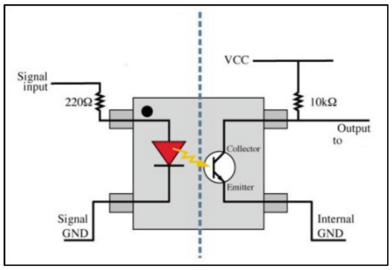


Figure 3(a). Opto Isolation circuit diagram

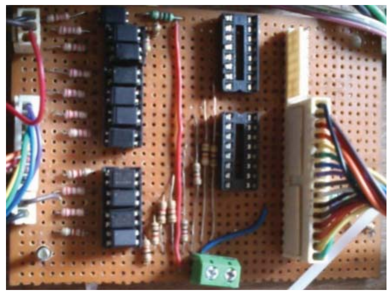


Figure 3(b). Opto isolation circuit

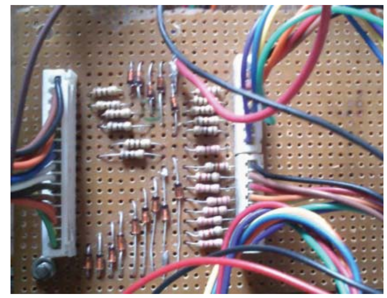


Figure 3(c). Signal Conditioning Circuit

V. OPTO ISOLATION CIRCUIT

5.1. Overview-

The Isolation Circuit is essential for isolation between FPGA and the Servo Motors. Since FPGA works at 3.3v it cannot be used to drive the motors as it would cause reverse current, which consequently would damage the FPGA. Hence Isolation circuit is important to completely isolate them from each other. We used an 817A opto coupler IC for the Isolation Circuit. The circuit is shown in Figure 3(b). The Isolation circuit also provided the necessary amplification to drive servo motors. Since the voltage produced by the FPGA was in the order of 3.3V and the voltage required for servo operation was 6.0V the Isolation circuit provided the necessary amplification. Based on the output from the FPGA the circuit maintains the digital logic state of '1' or '0' for corresponding generation of PWM signal for the servo.

5.2. Signal conditioning Circuit-

A signal conditioning circuit as shown in Figure 3(c), is an electrical regulator designed to automatically maintain a constant voltage level. In our Design the signal conditioning circuit is used for connecting the FPGA and the Atmega 156 micro controller for bidirectional transfer of DATA, ADDRESS and control packets such as ACK, NACK. The Atmega 16 micro controller operates at 5V for logic '1' whereas a FPGA operates at 3.3V for the same. Hence voltage from MC is reduced to 3.3V in order to avoid malfunctioning of the FPGA due to excessive voltage.

5.3 Block Diagram-

The block diagram explains the hierarchy of the system. The bi-directional lines indicate bi-directional data flow and control. The single arrow indicates a master to slave relationship. Based on the user selection the Manual or Automatic the ARM would operate accordingly. The Opto isolation circuit is used to shield the FPGA and the MC (Atmega16) from reverse current as shown in Figure 4(a).

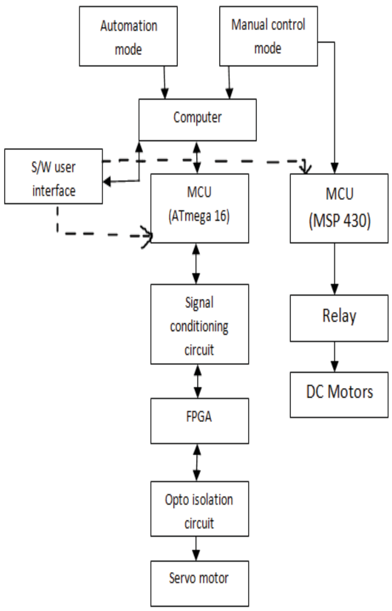


Figure 4(a).Block diagram

Figure 4(b) The complete robot

Figure 4(c).Robotic Arm

VI. CONCLUSION

The paper presents a study of the construction of a Robot ARM in order to replicate a human arm. The robotic arm with the robot was tested in different terrains and its operation was noted. The arm was subjected to temperatures between 16 and 45 degree Celsius and there was no deterioration in performance measured. The payload capacity was determined. The arm could lift objects up to 3Kgs and within the diameter of 10 cm. It lists out the various methods used for data communication between electronic devices and precautions taken to avoid errors. Circuits in the ARM were affected by electromagnetic radiation due to the electronic devices and surroundings which caused significant errors which deteriorated the performance. This problem was solved by using shielded twisted pair cables to protect the signals from being affected by electromagnetic interference (EMI). Further all electronic devices were shielded to avoid being affected by EMI. It also formulates certain guidelines to make an all purpose ARM which could operate at tested conditions mentioned. Future work is concentrated on (a) building micro-servos to accommodate fingers and to increase the Pay load capacity up to 10Kgs or higher (b) build multiple ROBOT Arm's and to make them coordinate with each other to accomplish tasks in an ordered manner and (c) incorporate Artificial Intelligence to the constructed ROBOT.

REFERENCES

[1] U. Meshram, P. Bande, R.R. Harkare, "Hardware and Software Co-design for Robot Arm Position Control using VHDL and FPGA", International Conference on Multimedia, Signal Processing and Communication Technologies, IMPACT '09, pp.8-11, 2009.
[2] Hao , W. Guan, Leck , Y.Lee, Hun, L.Chot,"6-DOF PC-based Robotic Arm (ROBOARM) with efficient trajectory planning and speed control", 4th International conference on Mechatronics (ICOM), pp. 1-7, 2011.
[3] J. Losansky, M. Rentzsch, H. Guldner, "Intelligent measurement and control platform using SPARTAN3 FPGA", In Proc.of European Conference on Power electronics and application, August, 2005.
[4] V.Ramakrishnan , N.S. Gopal , R. Ashok, S. Moorthi, "FPGA based DC Servo motor control for remote application of movements for a surgical arm", 10th Regional IEEE Conference.(TENCON), pp. 671- 675. Nov. 2011.
[5] O.Fukunda , T.Tsuji , M.Kaneko , A.Otsuka, "A human-assisting manipulator teleoperated by EMG signals and arm motions", IEEE Transactions on Robotics and Automation, Vol.19, Issue-2. pp. 210-222 April 2003.
[6] W.Wei , Y.Pan , K.Furuta, "Internet based Tele-control system for wheeled mobile robot", International conference on Mechatronics and Automation, Vol.3, pp.1151-1156, 2005.