Design, Implementation and Control of a Robotic Arm Using PIC 16F877A Microcontroller

Arian Faravar

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
February 2014
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Hasan Kömürcügil
Supervisor

Examining Committee

1. Prof. Dr. Hakan Altınçay
2. Prof. Dr. Hasan Kömürcügil
3. Asst. Prof. Dr. Ahmet Ünveren

ABSTRACT

This thesis focuses on design, implementation and control of a five degree of freedom (DoF) robotic arm using servo motors. The control of robotic arm is achieved by a PIC 16F877A microcontroller. The main duty of microcontroller is to generate pulse width modulation (PWM) signals which are applied to servo motors for achieving the desired rotation. Each servo has a different specification. Therefore, a PWM pulse could have a different effect on servos. Most of the time, it is crucial to apply the exact PWM pulses for achieving the desired rotation. The main advantage of controlling the servo motors with PWM signals is that they can be programmed to have an initial position and to rotate with an exact degree with respect to the requirements. In this study, six servo motors are employed to realize the robotic arm. Four servos are utilized to control the body motion including base, shoulder and elbow and two smaller servos are employed for the motion of end effector.

In this thesis, a general formula is derived for finding the pulse duration (pulse width) so as to achieve the desired rotation in each servo motor. The main advantage of this formula is that it can be used for any servo motor with different specification. The operation of designed robotic arm has been experimentally verified. Simulation and experimental results are presented and discussed.

Keywords: Microcontroller, Servo Motor, PWM, Robotic

ÖZ

Bu tez serbestlik derecesi beş olan robot kolunun servo motorlar kullanılmak suretiyle tasarımı, uygulaması ve denetimi üzerine odaklanmıştır. Robot kolunun denetimi PIC 16F877A mikroişlemcisi tarafından başarılmıştır. Mikroişlemcinin esas görevi istenen dönme hareketinin gerçekleştirilmesi için servo motorlara uygulanacak olan darbe genişlik modülasyonlu (DGM) sinyalleri üretmektir. Her servo motorun farklı bir çalışması vardır. Dolayısıyla, bir DGM darbesinin servo motorun üzerinde farklı bir etkisi olabilir. Çoğu zaman, istenen dönme hareketini gerçekleştirmek için, doğru DGM darbelerini uygulamak çok önemlidir. Servo motorların DGM sinyalleri ile control edilmesinin esas avantajı, servo motorların programlama yardımıyla başlangıç pozisyonunu elde etmesi ve istenen açı ile dönmelerinin gerçekleştirebilmesidir. Bu çalışmada, robot kolunun gerçekleştirilmesinde toplam altı servo motor kullanılmıştır. Bunlardan, dört servo motor taban, omuz ve dirsek hareketlerini içeren gövde hareketinin kontrolünü sağlamak için kullanılmıştır. Geriye kalan iki küçük servo motor ise sondaki efektör hareketinin denetimi için kullanılmıştır.

Bu tezde, servo motorlar için istenen dönme hareketini yapacak olan darbe süresinin bulunmasını sağlayan genel bir formül türetilmiştir. Bu formülün esas avantajı, farklı özelliklere sahip servo motorlara uygulanabilir olmasıdır. Tasarlanan robot kolunun çalışması deneysel olarak yapılmıştır. Benzetim ve deneysel sonuçlar sunulmuş ve tartışılmıştır.

Anahtar Kelimeler: Mikroişlemci, Servo Motor, DGM, Robotik

DEDICATION

To my lovely family, fiancé and all those who supported me through the preparation of this thesis.

Arian Faravar

ACKNOWLEDGEMENT

I would like to thank my thesis supervisor, Prof. Dr. Hasan Kömürcügil for his continuous support and precious guidance throughout this study.

I sincerely thank the committee members of my thesis defense jury, Prof. Dr. Hakan Altınçay and Asst. Prof. Dr. Ahmet Ünveren for their helpful comments and on this thesis.

Last but not least, my deepest thank goes to my family for their support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

With the growth of technology, the need of new devices grows accordingly. Computer and electronic sciences is mostly premier in raising the new technologies. Of course the new technology could affect different engineering fields. For instance, if the robotics and artificial intelligence are considered, it reveals that the technology with its high potential, affected many different fields of studies. Therefore related fields of study could be combined to generate new technologies that can be used in wide fields.

The robots play important roles in our lives and are able to perform the tasks which cannot be done by humans in terms of speed, accuracy and difficulty. Robots can be employed to imitate human behaviors and then apply these behaviors to the skills that leads the robot to achieve a certain task [1]. They do not get tired or face the commands emotionally, and since they are designed by humans. They can be programmed and expected to obey and perform some specific tasks. In some cases the use of a robotic hand becomes remarkable. Robotic is applied in different forms and fields to simulate human behavior and motions [2]. There are different types of robots which are discussed in chapter two.

Our daily life is virtually affected by robots [3]. The idea of robotic is to create practical and useful robots that facilitate our daily tasks. Because of the independency of the robots, they have longer life time comparing with the humans and can be helpful in industry, dangerous tasks and nursing homes [4].

Most of tools, vehicles, electronic devices and cuisine are built and prepared with the help of industrial robots. For instance, there are industrial robot assembly lines which help in many cases that can operate more accurate and faster than humans [5]. Recently, robots operate in almost all human labors mostly in the fields which are unhealthy or impractical for workers. This fact causes the workers to have more free time to spend on skilled professions including the programming, maintenance and operation of the robots which are essential [3].

There are situations where a robot is a replacement for human because the human does not have the capability to work under the specific conditions, such as working in the space, under the water and etc, unless the person is equipped with some expensive special clothing and equipment. Therefore, while designing a robot, considering the factors such as concept and techniques, artificial intelligence and cognitive science are essential in order to obtain an effective design [6]. The other situation is when the robot is used to ease the actions done by the human or the human is handicapped.

Obviously, building a robotic arm is not a new idea, but still the design and the specifications can differ from other designs. For instance, the circuitry, degree of freedom (DoF), algorithm, program, attachments, equipment, accuracy and speed, completely depend on the designer’s tact.

The challenge is to be able to perform some physical tasks close to a human's hand actions, such as replacement and grabbing, under the conditions where a human hand is not a particular solution. Therefore, a robotic arm can be designed to perform the required actions which can be controlled by the humans. The robotic arm has a main processor which is using a PIC microcontroller.

1.1 Robot Arm Details and Summary

The robotic arm will be controlled via the designed controller and it will be able to grab, pick up and move objects according to their weights and shape. The manipulator design is mostly expected to pick up cubes and the geometric shapes like a box. Depending on the numbers of joints, DoF differs, but generally robotic arms operate using 4 or 5 servo motors. The servo motors are popular for their desirable characteristics for robotic application [11].

1.2 Problem Statement and Proposed Solution

This study intends to investigate the design, implementation and control of a 5 DoF articulated robotic arm using servo motors and PIC 16F877A microcontroller. The advantage of this microcontroller its low cost and in-circuit programmability [10]. A pulse could have a different effect on servos with different specifications. Therefore, most of the time it is crucial to be able to give the exact PWM pulse in order to rotate a servo to a specific rotation. The main advantage of controlling the servo motors with PWM signals is that they can be programed to have an initial position and to rotate with an exact degree with respect to the requirements [11].

Since each servo motor has a different specification in terms of pulse width range as discussed above, a simulation for each servo is made in Proteus and MATLAB, and the results are compared with the experimental results in order to obtain a general formula to find the exact PWM pulse required to rotate the servo to the expected degree. With the help of the proposed formula, it is possible to find the exact duration of the PWM signal which is required to meet the expectations.

Simulation results and real-time robot arm behavior are then compared with the results presented in [10].

Chapter 2

REVIEW OF EXISTING ROBOTS

Robots are used in different fields such as industrial, military, space exploration, and medical applications. These robots could be classified as manipulator robots and cooperate with other parts of automated or semi-automated equipment to achieve tasks such as loading, unloading, spray painting, welding, and assembling.

Generally robots are designed, built and controlled via a computer or a controlling device which uses a specific program or algorithm. Programs and robots are designed in a way that when the program changes, the behavior of the robot changes accordingly resulting in a very flexible task achieving robot. Robots are categorized by their generation, intelligence, structural, capabilities, application and operational capabilities. In this study robots are reviewed according to their structural properties [11].

- Linear Robots (including Cartesian and gantry)
- Cylindrical Robots
- Parallel Robots
- Spherical Robots
- SCARA Robots

- Articulated Robots

2.1 Linear Robots

A robot which has linear actuators cooperating with linear motors linked to a linear axis is known as a linear robot (also known as gantry or Cartesian). This link can be fixed or flexible connections between the actuators and the robot. The linear motor is attached directly to the linear axis [12].

Robots which use two motors in controlling a linear axis defined gantry robots. Each motor has a limited distance orthogonal to the linear axis [13]. Ball screws follow the same principles which either use linear motors or rotary motors. This kind of robots usually achieve tasks such as palletizing, unitizing, stacking, order grasping, loading, and coordinate measuring.

The manipulator (also known as end-effector) of the linear robots is connected in an overhead way that allows the robot to move along the horizontal plane easily, where each of these movements are perpendicular to each other and are basically defined as x, y for horizontal axis and sometimes z in case of having a vertical axis.

Any action taken by x and y-axis need to have position accuracy of less than ±5.0 mm and repetition of ±1.0 mm. Figure 1 shows Hercules x-y gantry robot made in USA. Figure 2 shows a cartesian horizontal reach and its symbol where d_1, d_2 and d_3 represent x, y and z axis respectively [14].

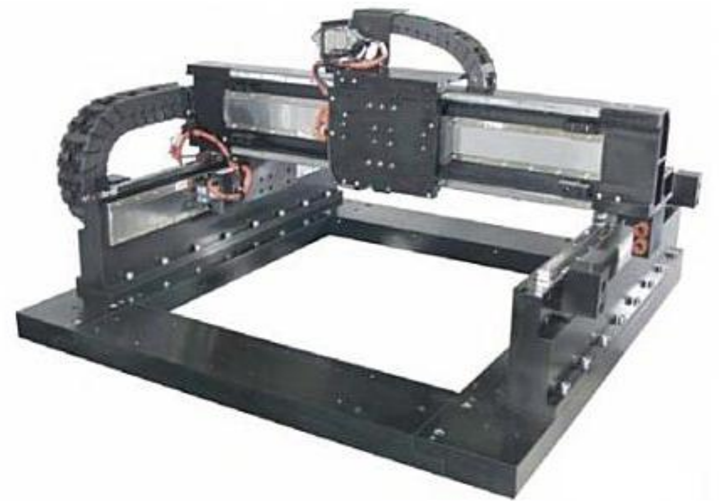


Figure 1. Hercules x-y gantry x-y robot

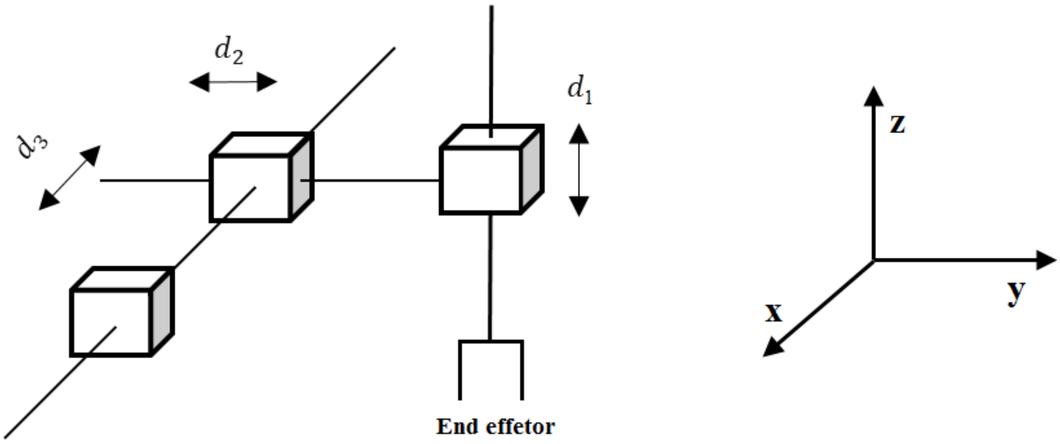


Figure 2. Cartesian horizontal reach and its symbol (on the right)

The mathematical expression regarding the movement of the linear robot can be written as

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \tag{1-1}$$

Where $q\epsilon\mathbb{R}^{n\times 1}$ and it represents the value of each joint. In equation (1-1) there exists n joints. The robot moves according to the given manipulator position. The desired movement of the robot can be represented by a vector as follows [15].

$$q_d = \begin{bmatrix} q_{d_1} \\ q_{d_2} \\ \vdots \\ q_{d_n} \end{bmatrix} \tag{1-2}$$

Where q_d is the desired position of the end-effector. If we define $\widetilde{q_\iota} = q_{d_i} - q_i$, as the desired position for the end-effector, the positions (x, y) are functions of the angles can be expressed as follows [15]

$$x_i = f_i(q) \qquad \text{for } i = 1,2, \dots, k \tag{1-3}$$

In our case the value of $i = 2$, which implies for the number of end-effectors since we are dealing with only (x, y) positions. If we let q_1 and q_2 to be the joint displacements, the forward kinematics will be as follows [15].

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} q_1 \\ 0 \end{bmatrix} ; \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{1-4}$$

Where the first vector is considered by the value of the first displacement q_1 and the second vector considers the movement of q_1 and q_2 with respect to the x and y-axis [15].

The advantages and disadvantages of linear robots are as follows [14].

Advantages:

- Large workspace
- High speed and stiffness
- Good performance
- Good for multiple machines and lines
- Good handling with large loads

Disadvantages:

- Large structural frame
- Complex mechanical properties for linear sliding movements
- Energy inefficiency
- Large floor space requirement
- Limited workspace
- Common workspace restriction

2.2 Cylindrical Robots

Cylindrical robots have two prismatic joints: one rotary joint for positioning task and the end-effector of the robot forms a cylindrical workspace. The main idea of the cylindrical robots is to mount a horizontal arm which moves in forward and backward directions. The horizontal arm is linked to a carriage which goes up and down and is connected to the rotary base. Schematic cylindrical robot and its symbol are shown in

Figure 3 [14]. Since both of the units move on the base, the workspace is annular space of the cylinder.

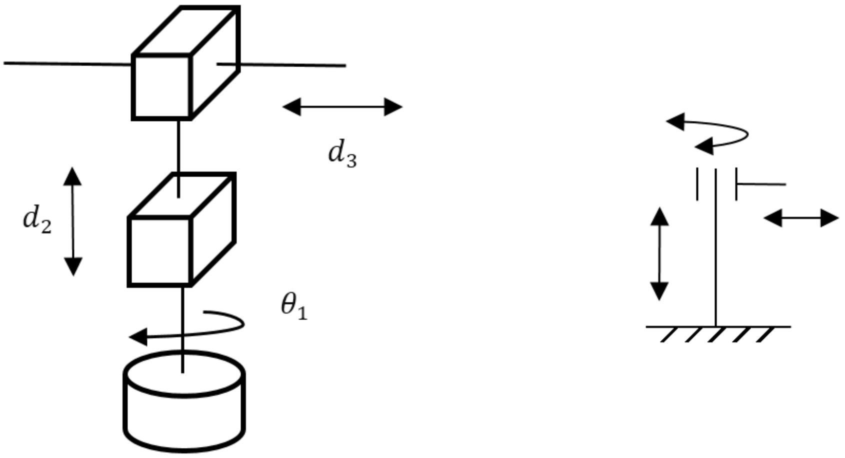


Figure 3. Cylindrical robot and its symbol (on the right)

When the arm of the robot has a revolute and two prismatic joints, it can operate in z-axis and each point that can be reached by this robot can be represented by the cylindrical coordinates. As shown in Figure 4, the robot can move in and out in z direction, can elevate in y direction and can rotate in θ direction. The arm can move in directions between the specific upper and lower boundaries [16].

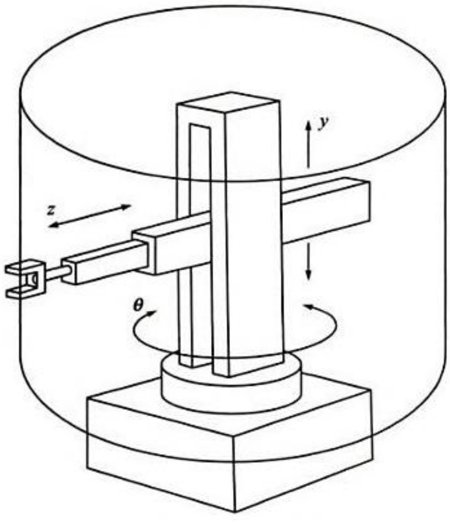


Figure 4. Cylindrical Robot Arm Configuration

It is possible to have additional joints for the end-effector so that the horizontal arm can extend. This kind of robots have a wide use in electronics manufacturing [17]. Figure 5 shows a cylindrical robot designed by Seiko instruments [18].

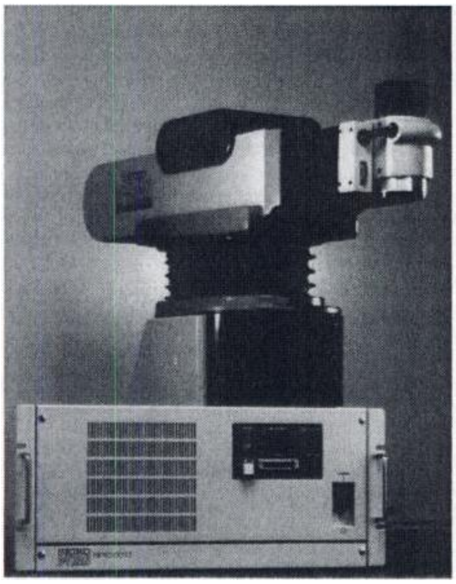


Figure 5. Cylindrical Robot

The advantages and disadvantages of cylindrical robot are as follows [17]

Advantages:

- No collisions while moving
- Two linear axis result in simpler design

Disadvantages:

- Large structural frame
- Incompatible with other robots

- Inaccurate on the end resolution compared to Cartesian robot

2.3 Parallel Robots

A parallel robot has an end-effector with n DoF which is connected to a fixed base. The connection is done by at least two independent kinematic chains which provide the movements of the robot. A generalized parallel manipulator has a closed-loop kinematic chain mechanism where the manipulator is linked to the base. In Figure 6, potentially the first parallel robot which was patented in the USA is shown [20].

There exists different definitions and types of a parallel robots yet the most common properties are as follows [19]

- The end-effector should be supported with at least two chains. Each of the chain should have at least one simple actuator.

- The number of actuators should be equal to the number of DoF of the end-effector.

- There should not exist any mobility of manipulator when the motors are locked.

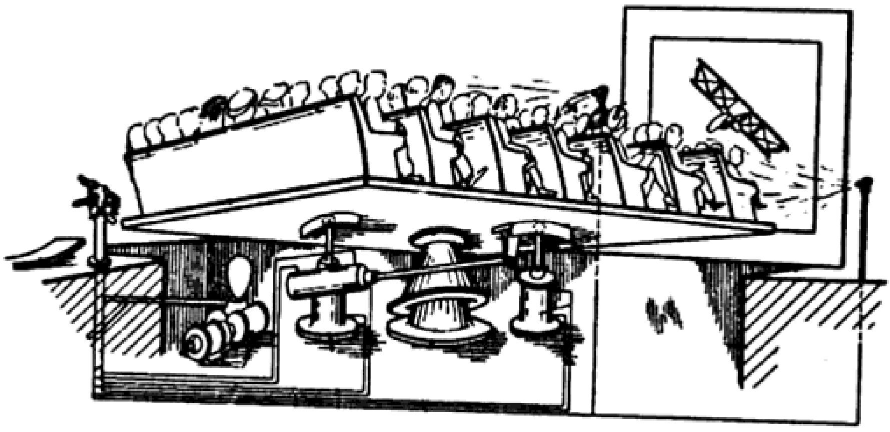


Figure 6. Potentially the first spatial parallel mechanism

With having two platforms, parallel robots generally provide relative movement between the moveable platform and the fixed one. Parallel robots have become essential in both industry and academic and likewise, the recent researches on mechanism theories, mobility analysis, dynamics and kinematics modeling, electronics and optimizations have made remarkable developments on the parallel robots [20].

There are different parallel robot configurations but two kinematic designs have become popular. First design is a parallel robot shown in Figure 7, which has tripod with three axis connecting the end-effector to the movable platform and the base, and has a wrist with 2 or 3 DoF [21]. The second design is a hexapod with six axis which results in full spatial movement as shown in the Figure 8 [21].

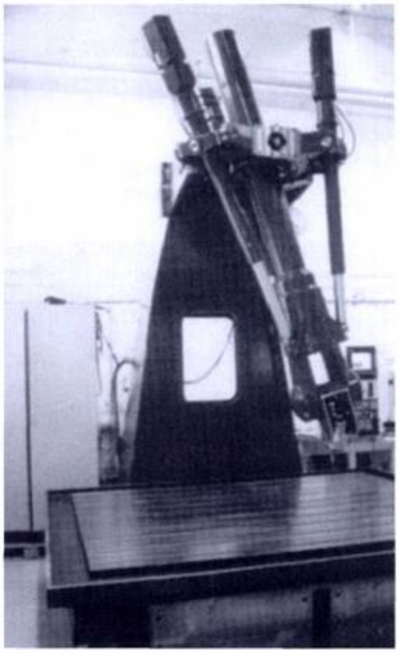


Figure 7. Tripod with three axis parallel robot

Figure 8. Hexapod with six axis parallel robot

Because of the interconnections of the links and axis, this robot has the following advantages [22]

- High stiffness
- High accuracy
- High load capacity
- High damping

Parallel robot disadvantage [22]

- Limited kinematic dexterity

2.4 Spherical Robots

The spherical robot (also known as polar robot) is huge in terms of size and has a telescopic arm. Spherical robot basic movements are rotation at the base and angularly up and down at the arm [23].

Spherical robots have at least two movable joints and one fixed joint. The schematic diagram and symbol of the spherical robot are shown in Figure 9. The motion of the spherical robot consists of the following three movement steps; the first movement defines the rotation of the base along vertical axis. The second movement defines the rotation of the arm and finally the third movement defines the in and out motion. The workspace of the spherical robot depends on the volume of globe of the sphere [14].

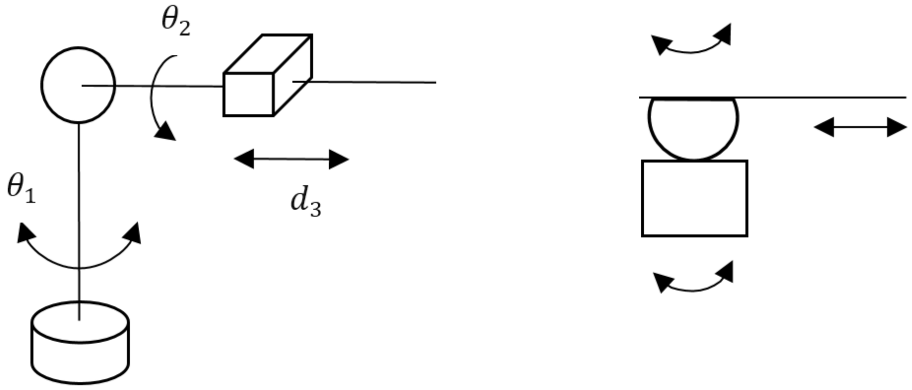


Figure 9. Schematic diagram of spherical robot and its symbol (on the right)

The workspace of the robot is the space between two concentric hemispheres. When the arm is fully retracted, the reach of the arm is the inner hemisphere and when the arm is fully straightened, the reach is the outer hemisphere [24]. A typical spherical robot configuration in shown in Figure 10 [23].

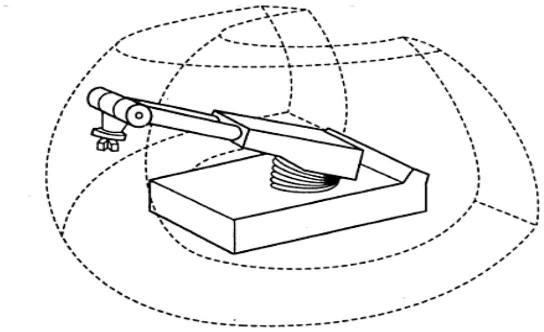


Figure 10. Spherical Robot Configuration

Advantages of spherical robot are as follows [14]

- Light weight
- Simple kinematics
- Compatible with other robots especially with ones in a common workspace
- Sharp joints level
- Good resolution due to perpendicularity of the end-effector’s errors

Disadvantages of spherical robot are as follows [14]

- Need of variable torque due to the large size
- Challenging counter balancing
- Chance of having collision with obstacles due to bounded ability to avoid collisions
- Large position errors due to the rotation and proportional radius

2.5 SCARA

Selective Compliance Assembly Robot Arm (SCARA) was first designed and invented in early 1960s in Japan. SCARA robots are perfect for the applications which require high speed and repetitive point to point movements. This is why SCARA is used widely in assembly operation [14]. Special end-effector movement makes SCARA ideal for the tasks which require uniform motion and accelerations in a circular form [22].

SCARA consists of two parallel rotary joints and a prismatic joint. The rotary joints can move along the horizontal plane and the prismatic joint moves along the vertical plane. One of the special characteristic of SCARA is that the robot is smooth while operating on x and y-axis but very strong versus the z-axis. Figure 11 shows the schematic diagram of SCARA.

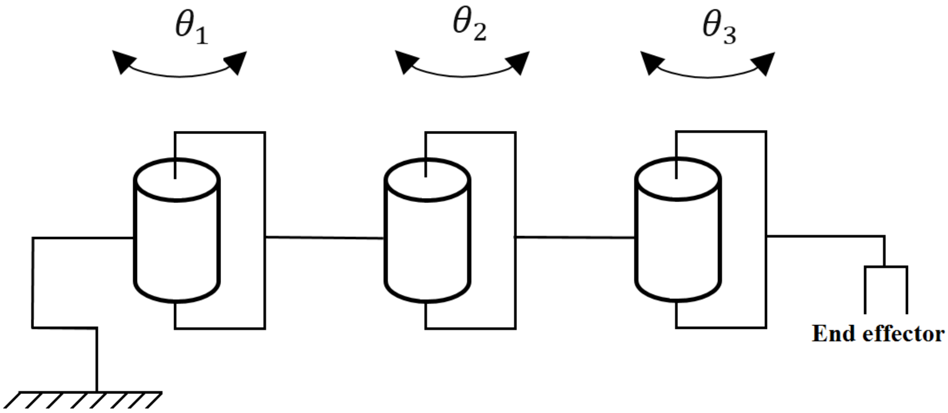


Figure 11. Schematic Diagram of SCARA

An especial design of SCARA which was built in Sweden which is known as Spine robot, is designed in a way that it consists of several discs which are linked together via two pairs of hydraulic cable actuators. Each robot has at least four cables in each rigid part which results in total of eight cables with four DoF.

The robot operates when these cables are pulled by the cylinders which are placed in the base of the spine robot [14]. The schematic diagram of spine robot is illustrated in Figure 12.

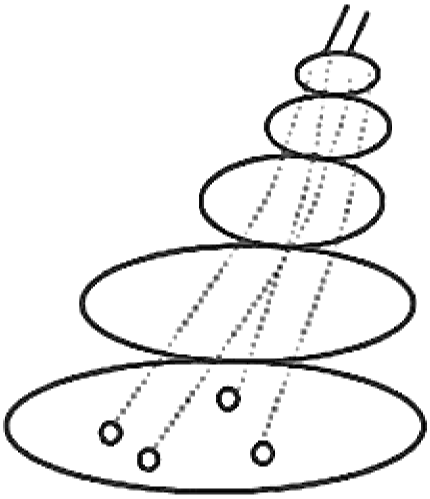


Figure 12. Schematic Diagram of Spine Robot

SCARA arm is able to pick up a part vertically from a horizontally placed table and move along the horizontal plane to a desired point and accomplish the assembly task by lowering the arm and placing the part at its proper location. Figure 13 shows a typical SCARA robot [24].

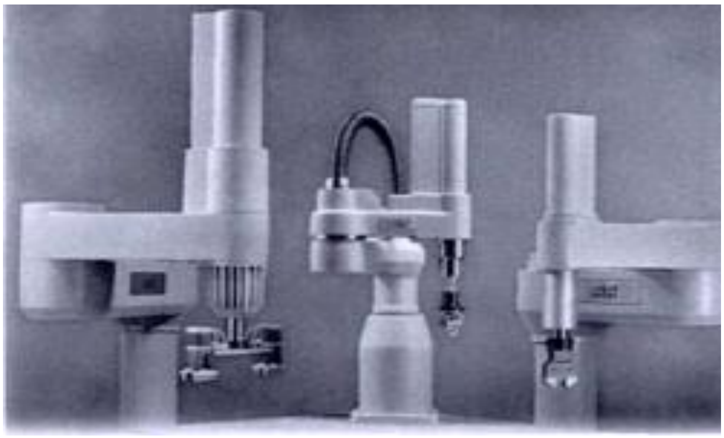


Figure 13. SCARA robot

2.6 Articulated Robots

Articulated robots (also known as revolute robots) have three fixed axis connected to two revolute base. All joints of an articulated arm are revolute and most likely represent the human arm. Figure 14 shows the schematic diagram and symbol of an articulated robot [14].

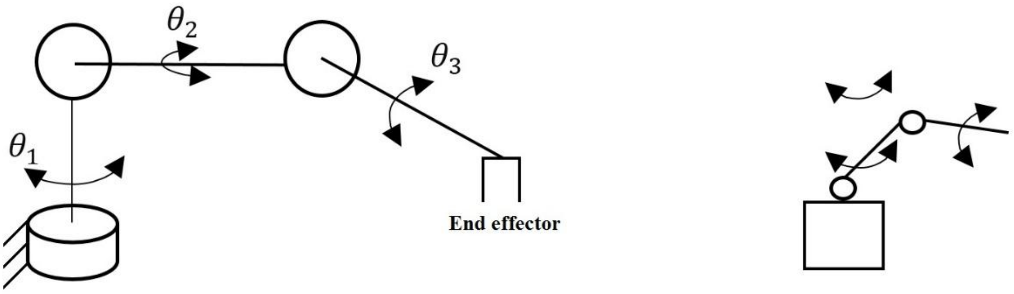


Figure 14. Schematic diagram of an articulated robot and its symbol (on the right)

The moving rigid objects are called links, revolute joints are called hinges and prismatic joints are called sliding joints. Each joint defines the relative motion of the other two object it links which determines the subset of the whole configuration space. Each configuration subset is a different position for each link. These subsets are simple to measure by considering a distance or an angle with each joint.

A robotic arm can be said to be a typical example for articulated robot. An important matter which should be considered is that the dimension of the configuration space increases with the number of joints however the operation speed is limited due to the different payloads at the manipulator and nonlinear environment [25].

Almost 80% of the registered robots are articulated and up to 20% are linear robots. Figure 15 shows a typical articulated robot [11].

Figure 15. Articulated Robot

The advantages and disadvantages of articulated robots are [14]

Advantages:

- Superb structural flexibility
- Compatible with other robots operating in common workspace
- High rotation speed

Disadvantages:

- Low accuracy and resolution because of rotary joints and positional errors
- Counter balancing difficulties due to the large and variable torque
- High chance of collision
- Dynamic instability due to higher moment of inertia and gravity

With investigating the properties and specification of each robot, a table can be illustrated to show the main differences between these robots in a general form. If a robot is to be chosen and used for a specific task, this comparison can lead to have a better understanding and decision. Table.1 shows a general comparison between fundamental robot arms [16].

Table 1: Comparison of fundamental robot arms

CONFIGURATION	ADVANTAGES	DISADVANTAGES
Cartesian (Three linear axes) *x*: base travel *y*: height *z*: reach	• Easy to visualise • Rigid structure • Easy off-line programming • Easy mechanical stops	• Reach only to front and back • Requires a large floor space • Axes are hard to seal • Expensive
Clyndrical (1 rotating and two linear axes) *Θ*: base rotation *Φ*: height *z*: reach	• Can reach all around • Rigid y,z-axes • Θ- axis easy to seal	• Cannot reach above itself • Less rigid Θ-axis • y-z-axes hard to seal • Wont reach around obstacles • Horizontal motion is circular
Spherical (2 rotating and one linear axes) *Θ*: base rotation *Φ*: elevation angle *z*: reach	• Can reach all around • Can reach above or below obstacles • Large work volume	• Cannot reach above itself • Short vertical reach
Articulated (3 rotating axes) *Θ*: base rotation *Φ*: elevation angle *Ψ*: reach angle	• Can reach above or below obstacles • Largest work area for least floor space	• Difficult to program off-line • Two of more ways to reach a point • Most complex robot

Chapter 3

BUILDING A ROBOTIC ARM

3.1 Building the kinematic structure

The first step of designing a robot is to decide the dimension and workspace configuration according to the requirements. The next step is to decide the specification of each actuator [26]. The structure of the robot is built with compacted wooden sheets in order to decrease the overall weight of the robot. The compacted wooden sheets are also strong enough to keep and hold the whole parts tightly together. The arm is attached to a base which is the most bottom part of the robot. It is important to mention that the base ought to have considerably heavy weight in order to maintain the general balance of the robot in case of grabbing an object.

Although the idea of using stepper and gear motors is brilliant, but physical movement of the robot is done by using servo motors. The advantage of the servos is that they can be programmed to return to their initial position. Since the servo motors operate using the signals received from the microcontroller, they could be programmed according to the requirements. However, this characteristic of the servo motors is actually a disadvantage, because the chance of sending and receiving a wrong signal is high which causes the servo to operate incorrectly.

The developed robot in this study is a stationary articulated robotic arm with 5 DoF with only revolute joints which includes base, shoulder, elbow, gripper pitch and gripper spin.

All parts of the robot including the parts for shoulder, elbow, gripper and etc, were printed on the compact board and cut accurately. Some carpentry processes where applied to the sheets to make the necessary holes and cuts to connect the parts to each other and to keep the actuators tightly.

The gripper of the arm is designed in a way which uses a single actuator and follows a basic physical gear concept. This means that when the mini servo actuates, it turns the gear which is attached to it causing the gripper to expand and contract. Figure 16 shows the template of the gripper with its magnitudes. The design of the base, shoulder and elbow with their measurements are shown in Figure 17 [27].

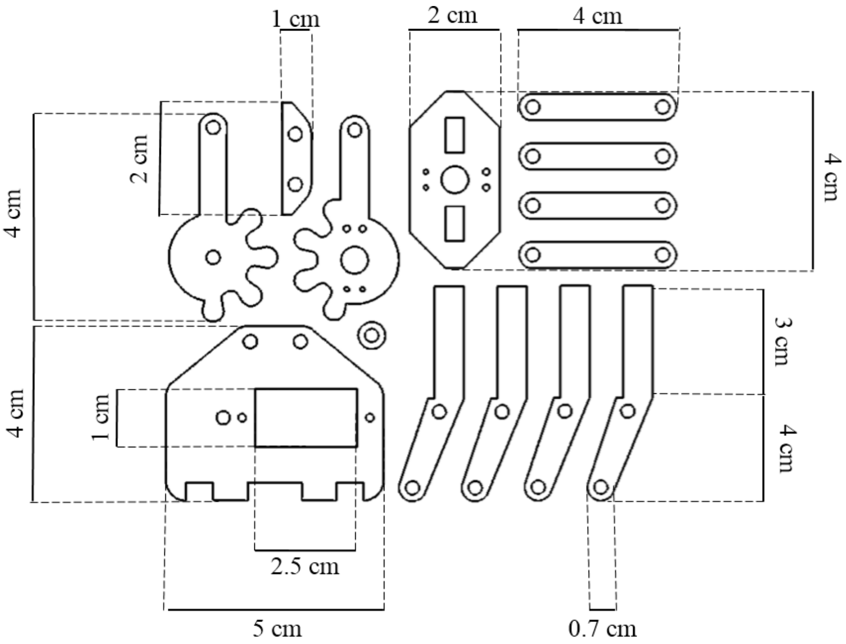


Figure 16. Gripper Template

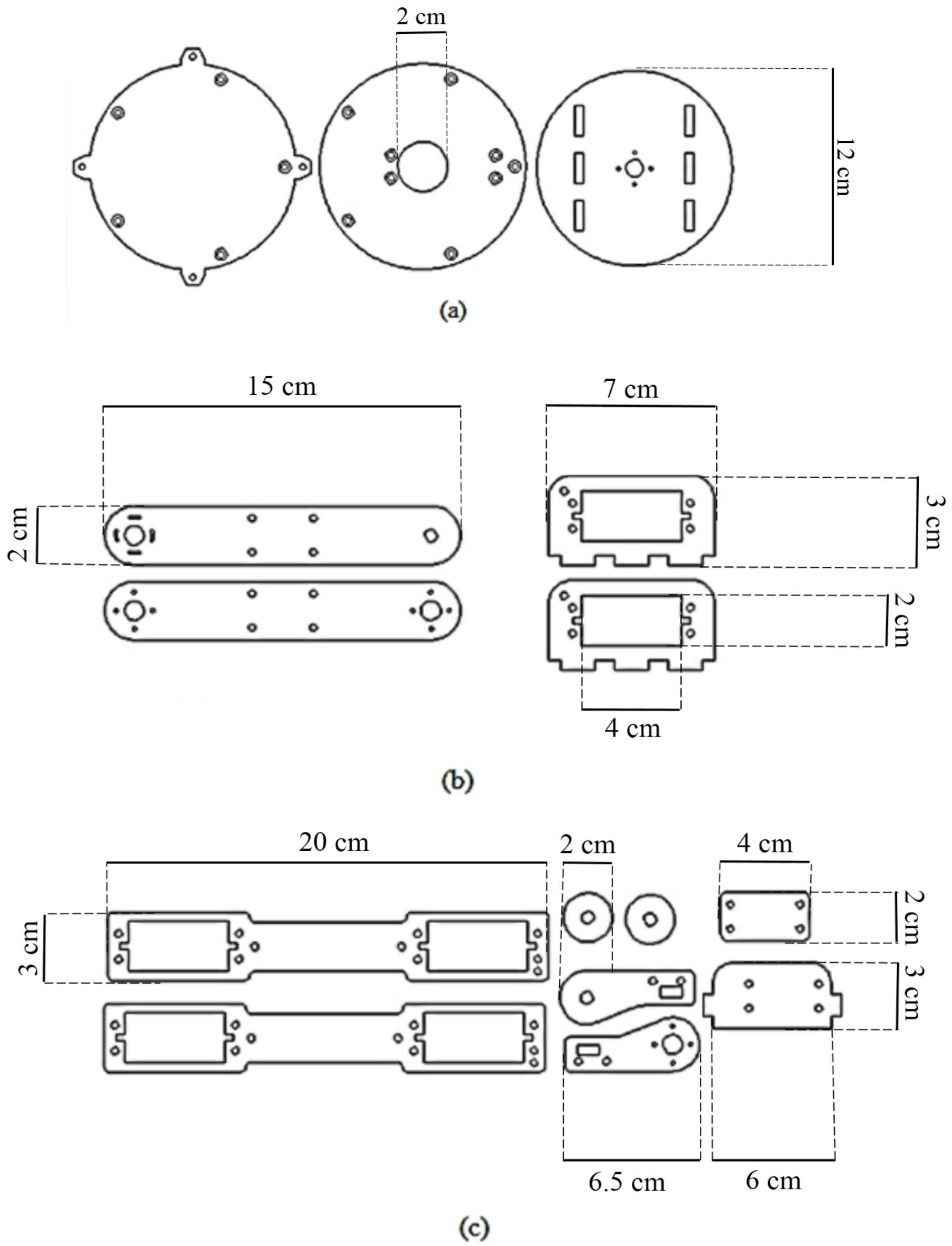


Figure 17. Template of (a) base, (b) shoulder, and (c) elbow

The dimensions can differ for different designs, but this should be mentioned that the dimensions given in this study are chosen with respect to the servo motors which are used in the robot. The power, torque and size of the servo motors can affect the dimensions. For instance if the servo motor used in the elbow is changed with a less powerful servo, the length of elbow should be decreased accordingly, because the servo may not have enough power to pull the elbow up.

On the other hand, if a longer elbow is required in order to enlarge the workspace of the robot, the height of shoulder and elbow from the base should be changed respectively in order to maintain the physical balance of the robot. In general, if one part of the structure is changed in dimension, the change should be applied for all parts of the robot accordingly in order to eliminate the instability problem.

All the parts were cut and drilled properly according to the design template. Then, all parts were painted and the robot was assembled. Figure 18 shows the final look of the robotic arm.

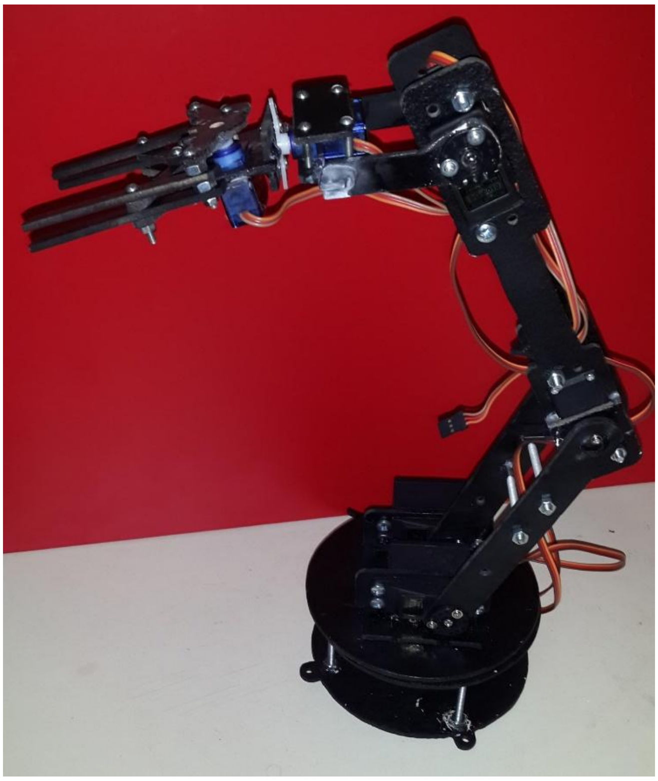


Figure 18. Structure of the robotic arm

3.2 Mathematical model of the kinematics

Each robot design involves mathematical modelling of the kinematics, structure design, electronic design and software design [28]. The robotic arm has a total of five axis. Three major axis which correspond to the base, shoulder and elbow are needed to move the arm to the desired spot, and two minor axis which correspond to the gripper pitch and gripper spin. The design has six rotary joints. Although we consider the number of joints as five because two joints that move the shoulder, rotate in the same direction with the same speed. Therefore they are counted as one joint. Figure 19 shows the precise link coordinate diagram of axes and joints [29].

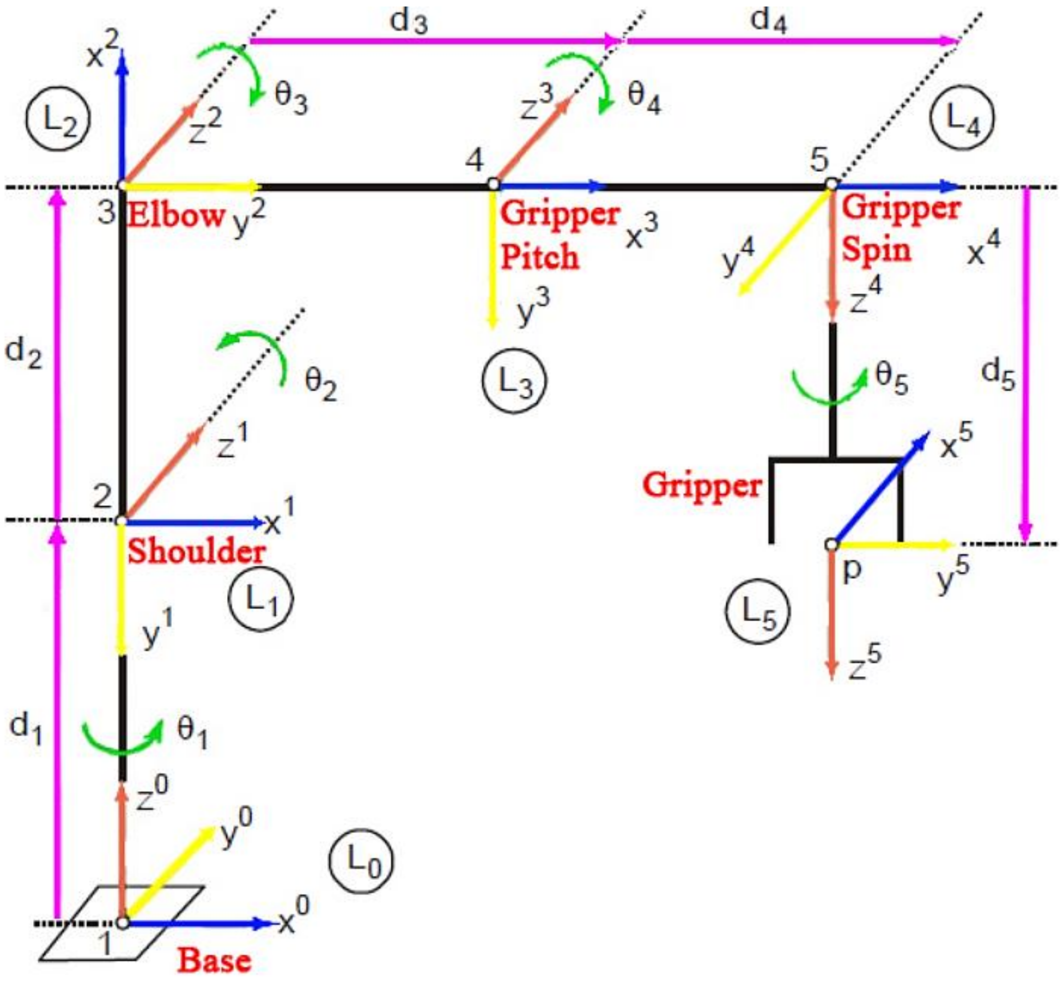


Figure 19. Link coordinates of axes and joints of the developed robot

The parameters and kinematic equations of the robotic arm can be written as follows [29]

L_0 to L_5 – Six unit frames

d_5 – Gripper length

q_1 to q_5 – Joint variables ($q = \theta$)

P – Gripper spin

d_1 – Height of shoulder from base

x^0 to x^5 – Motion of base, shoulder elbow, gripper pitch and spin in x direction

y^0 to y^5 – Motion of base, shoulder elbow, gripper pitch and spin in y direction

z^0 to z^5 – Motion of base, shoulder elbow, gripper pitch and spin in z direction

$1,2,\ldots,5$ – Rotary joints

a_2, a_3, a_4 – Link lengths

Figure 20 shows the axes and joints on the robotic arm.

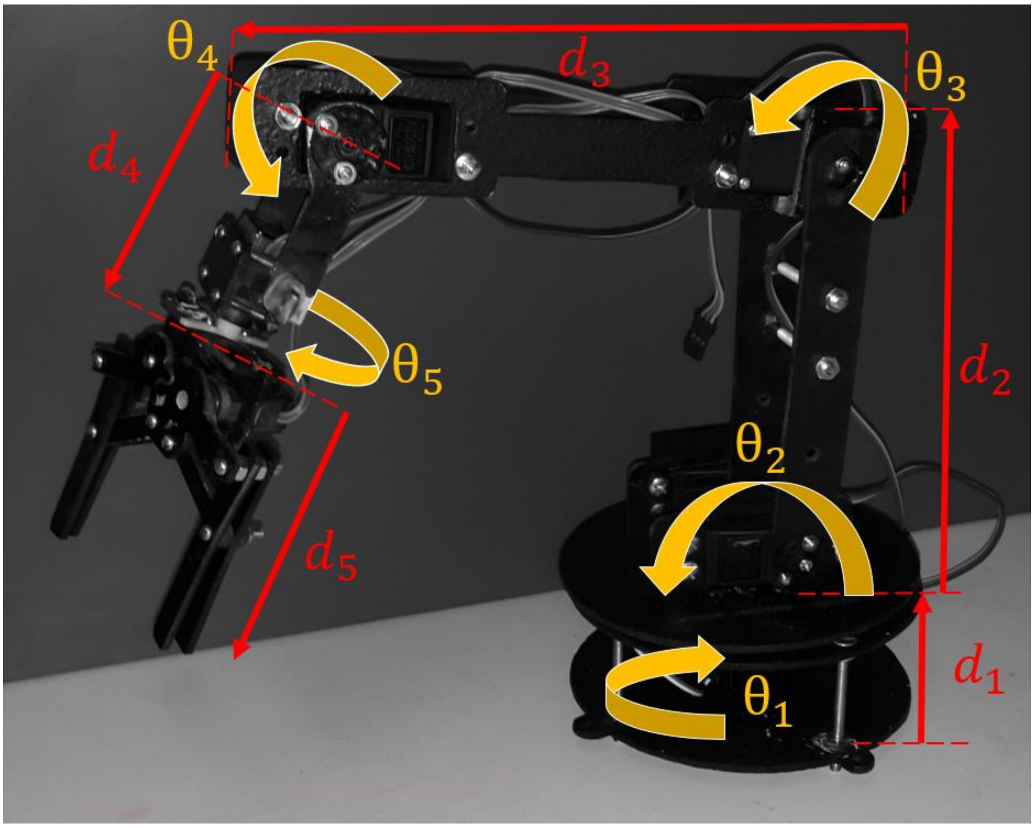


Figure 20. Axis and joints on the robotic arm

Base:

$$q_1 = \theta_1 \tag{3-1}$$

where $\theta_1 = \tan^{-1}\left(\frac{y^0}{x^0}\right)$ and $[-2\pi \leq \theta_1 \leq 2\pi]$ for full clockwise and counter clockwise rotation.

The servo motor which is responsible for the rotation of the base (S1) is connected to Port D.0 of the microcontroller as shown in Figure 28.

Shoulder:

$$x_{shoulder} = \sqrt{(x^1)^2 + (y^1)^2} \tag{3-2}$$

$$z_{shoulder} = z^1 - D_1 \tag{3-3}$$

$$x_{shoulder} = x_{shoulder} - D_2 \tag{3-4}$$

$$h = \sqrt{x_{shoulder}{}^2 + z_{shoulder}{}^2} \tag{3-5}$$

where $\theta_2 = \tan^{-1}\left(\frac{z_{shoulder}}{x_{shoulder}}\right) - \cos^{-1}(\frac{(L_l)^2+(L_u)^2-h^2}{2L_l L_u h})$ and $[\frac{\pi}{6} \leq \theta_2 \leq \frac{2\pi}{3}]$ for ($L_l =$ Lower Link , $L_u =$ Upper Link, h = Height of shoulder from base) for the rotation from 30° to 120°.

Two servos which are responsible for the motion of the shoulder (S2 and S3) are connected to Port D.1

Elbow:

$$\theta_3 = \cos^{-1}(\frac{(L_l)^2+(L_u)^2-h^2}{2L_l L_u h}) \tag{3-6}$$

where $[\frac{7\pi}{6} \leq \theta_3 \leq \frac{\pi}{3}]$, for the rotation from 60° to 210°.

The responsible servo motor for the rotation of elbow (S4) is connected to Port D.2 of the microcontroller.

Gripper Pitch:

$$\theta_4 = -\theta_2 - \theta_3 \tag{3-7}$$

where $[-\frac{\pi}{4} \leq \theta_4 \leq \frac{\pi}{4}]$, for the rotation from -45° to 45°.

The servo motor for the gripper pitch (S5) is connected to Port D.3.

Gripper Spin:

$$\theta_5 = \cos^{-1}(\frac{(L_l)^2+(L_u)^2-h^2}{2L_lL_uh}) \tag{3-8}$$

where $[-\frac{\pi}{4} \leq \theta_5 \leq \frac{\pi}{4}]$, for the rotation from -45° to 45°.

The corresponding servo motor for the gripper spin (S6) is connected to the Port D.4 and the gripper servo (S7) is connected to Port D.5 which is responsible for opening and closing the end-effector.

3.3 Hardware implementation

In the current robotic arm, a PIC 16F877A microcontroller is chosen as the main processor. Since it has a good range of interfaces, including analogue and digital pins, pulse width modulation (PWM) and in-circuit debugging. This microcontroller can generate PWM signals on ten pins (PORT D and two pins from PORT C) [30]. The rest of the electronic equipment, and sensors, are decided according to the tasks which the robot is expected to achieve. The circuitry of the robot is first designed and tested on a bread board and then the actual printed circuit board (PCB) is built in the final step and the electronic devices are soldered on the PCB.

3.3.1 PIC 16F877A Microcontroller

PIC 16F877A microcontroller has 40 pins and is a popular microcontroller capable of doing complex tasks. This microcontroller has 8192 × 14 flash program memory which consists of 368 bytes of RAM and 256 bytes of non-volatile EEPROM memory. 33 pins are dedicated for input/output pins and 8 multiplexed analog/digital converters with 10 bits resolution. This microcontroller also has specifications such as PWM generator, 3 timers, analog capture and comparator circuit, universal synchronous receiver transmitter (USART), internal and external interrupt capabilities. Figure 21 shows the pin configuration of the PIC 16F877A microcontroller and the pins which are used for PWM generation are marked [31].

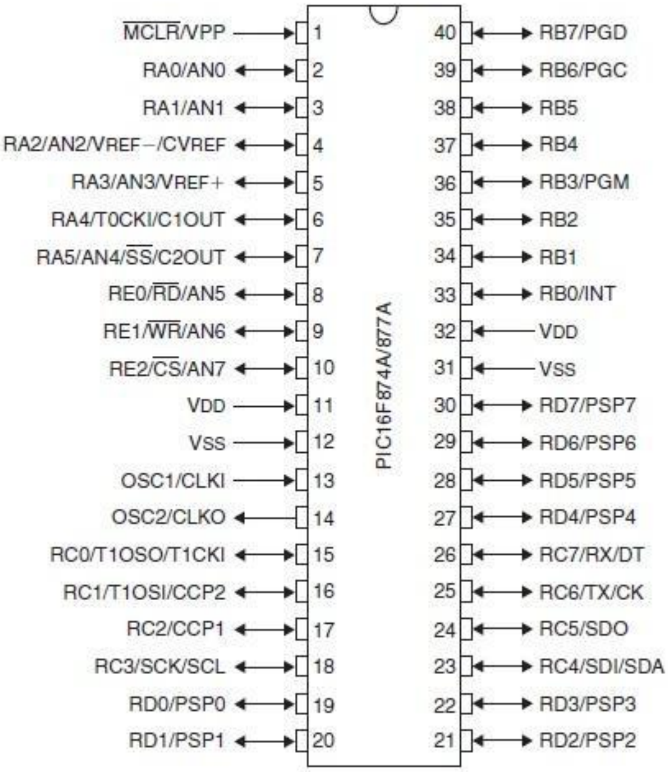


Figure 21. PIC 16F877A pin configuration

3.3.2 Servo Motor Principles and Interface with Microcontroller

Quick response and stability are remarkable properties of servo motors [32]. The servo motors used in this study operate with DC and are suitable for PWM control. The motor has series of gears attached to it and the potentiometer which does the feedback process. This feedback indicates the position of the servo which is derived from the corresponding servo potentiometer voltage [33]. A typical servo motor’s shaft is limited to rotate from 0 to 180° but it is possible to modify the motor for a continuous rotation. In the robotic arm assembled in this study, the servo motor which is used in the base is modified for a full rotation.

Servo motors are controlled with PWM signals. In this study, the PWM signals are generated with the microcontroller. The effect of pulse width on the direction of servo motor is shown in Figure 22. A pulse duration of 2ms rotates the servo clockwise at its full speed and 1ms pulse rotates the servo counter-clockwise with full speed. Giving the servo motor a pulse with 1.5ms width stops the motor or can be set to return to its initial position via a feedback control [31].

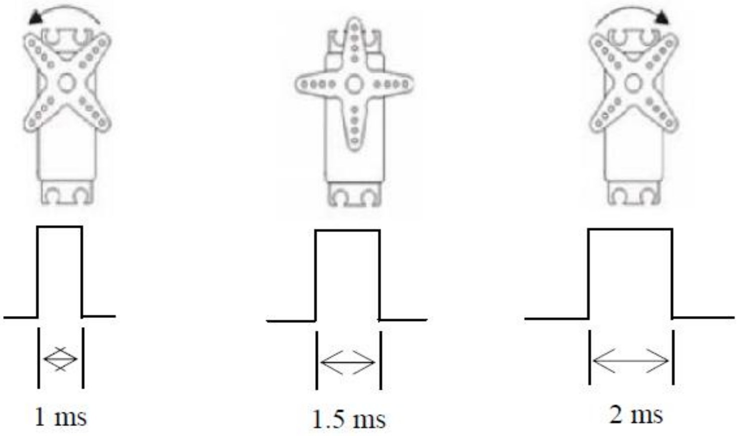


Figure 22. Effect of pulse width on direction of servo motor

The servo motor circuit is shown in Figure 23 [10].

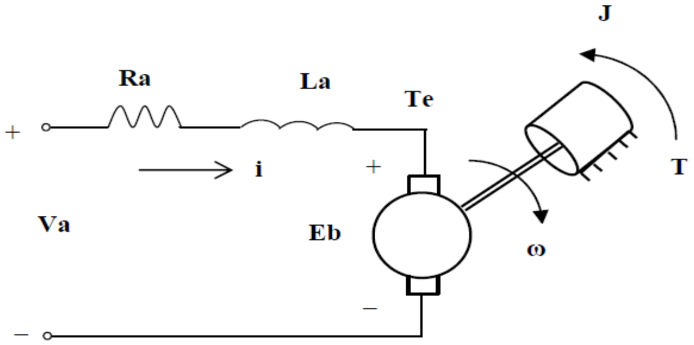


Figure 23. Servo motor circuit

The mathematical expression of the servo circuit can be written as follows [10]

$$V_a = R_a i + L_a \frac{d_i}{d_t} + E_b \tag{3-9}$$

$$J \frac{d\omega}{dt} = T_e - T \tag{3-10}$$

$$E_b = K\omega \, , T_e = Ki \tag{3-11}$$

where i is the armature current, V_a is armature voltage, R_a is armature resistance, L_a is armature inductance, K is torque, ω is rotor angular speed, T_e is electromagnetic torque, T is total load torque and J is the rotor inertia [10]. It is important to keep in mind that the built-in oscillator of PIC 16F877A microcontroller operates with 1MHz, therefore the oscillator should be adjusted accordingly.

For this robot the oscillator is adjusted by connecting a 4MHz crystal and two 22μF capacitors to oscillator pins (OSC1 and OSC2) of the microcontroller in order to increase the processing speed of the microcontroller, hence to let the servos to operate properly with respect to their input PWM signals.

Two types of servo motors are used in building the robot arm where each of which has a different specifications. The specifications of servo motors used (Tower Pro SG-5010) in base, shoulder, elbow and gripper pitch of the robot and servo (Tower Pro SG-90) used in gripper spin and gripper itself are shown in the Table-2 and Table-3, respectively [34, 35].

Table 2: Specifications of Tower Pro SG-5010 Servo Motor

<table>
<tr><th colspan="5">BASIC INFORMATION</th></tr>
<tr><th><i>Modulation</i></th><th><i>Torque</i></th><th><i>Speed</i></th><th><i>Weight</i></th><th><i>Dimensions</i></th></tr>
<tr><td>Analog</td><td>4.8V:
10.00kg-cm
6.0V:
11.00kg-cm</td><td>4.8V: 0.17sec/60^{o}
6.0V:0.14sec//60^{o}</td><td>38.0 gr</td><td>Length:
40.1mm</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Width:
20.3mm</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Height:
43.2mm</td></tr>
<tr><th colspan="5">ADDITIONAL SPECIFICATIONS</th></tr>
<tr><th colspan="2"><i>Rotational range</i></th><th colspan="2"><i>Pulse cycle</i></th><th><i>Pulse width</i></th></tr>
<tr><td colspan="2">180^{o}</td><td colspan="2">20ms</td><td>600-2400µs</td></tr>
</table>

Table 3: Specifications of SG-90 Servo Motor

<table>
<tr><th colspan="5">BASIC INFORMATION</th></tr>
<tr><th><i>Modulation</i></th><th><i>Torque</i></th><th><i>Speed</i></th><th><i>Weight</i></th><th><i>Dimensions</i></th></tr>
<tr><td>Analog</td><td>4.8V:
2.20kg-cm
6.0V:
2.50kg-cm</td><td>4.8V: 0.11 sec/60^{o}
6.0V: 0.10 sec/60^{o}</td><td>14.0 gr</td><td>Length:
23.1mm</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Width:
12.2mm</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Height:
29.0mm</td></tr>
<tr><th colspan="5">ADDITIONAL SPECIFICATIONS</th></tr>
<tr><th colspan="2"><i>Rotational range</i></th><th colspan="2"><i>Pulse cycle</i></th><th><i>Pulse width</i></th></tr>
<tr><td colspan="2">180^{o}</td><td colspan="2">20ms</td><td>400-2400µs</td></tr>
</table>

3.3.3 Implementation

The electronic circuit of the robot arm is simulated and tested on Proteus simulation software which will be discussed in the next chapter. Considering the results and behavior of the system derived from the simulation of the user button control diagram as shown in Figure 24, the actual circuitry is designed and implemented on bread board. Figure 25 shows the implementation of figure 24 on bread board. It should be mentioned that since the shoulder servos operate simultaneously, S2 and S3 are both connected to the portD.1 of the microcontroller to receive an identical PWM. As seen in Figure 24, each servo motor is operated with a button. Each button causes the microcontroller to generate the necessary PWM signal and send it to the corresponding servo motor.

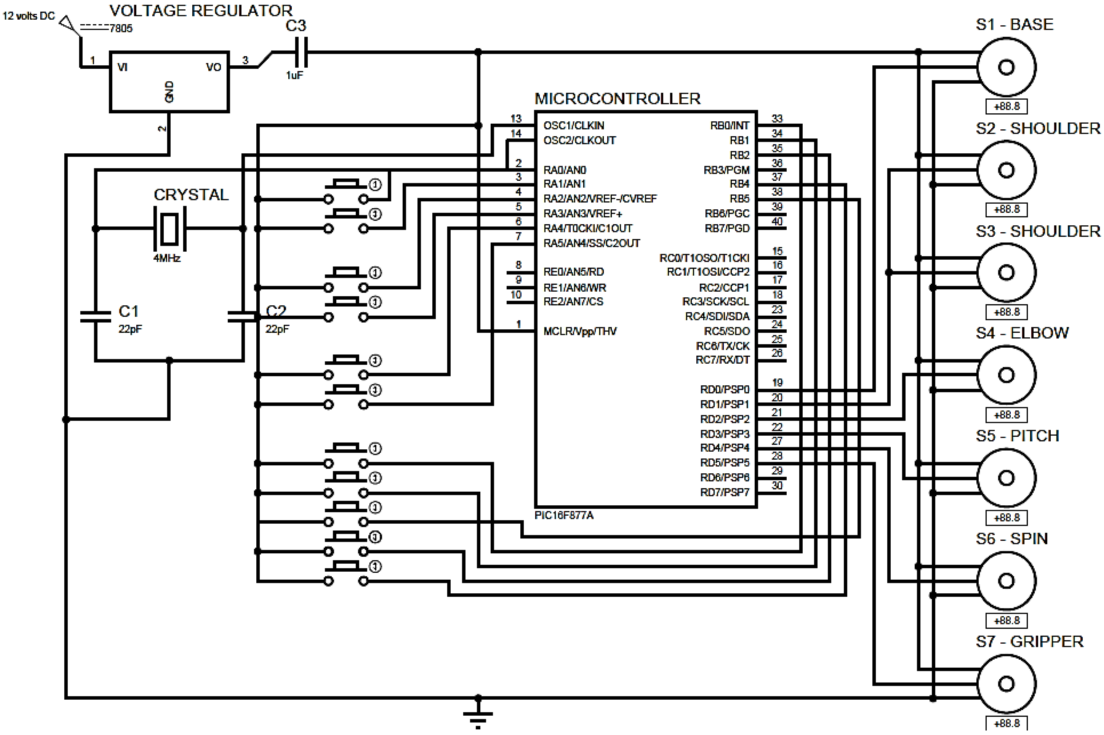


Figure 24. Schematic diagram of user-controlled robot arm circuit

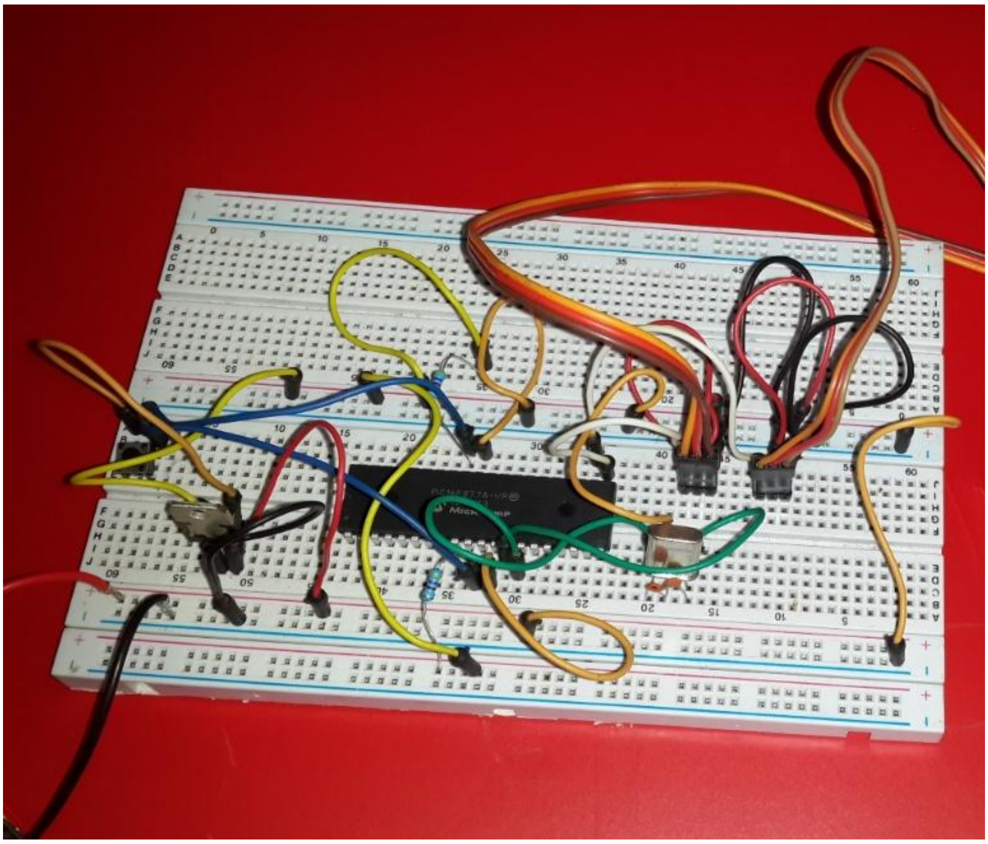


Figure 25. Hardware implementation of the robotic arm on bread board

3.4 Flowchart and Software Implementation

The behavior of the robotic arm can be shown in a flowchart in a generalized way as shown in Figure 26. As illustrated in the flowchart, the system starts with the initialization of the robot and the behavior when a button is pressed mentioned earlier in Figure 24. When a button is pressed, the corresponding PWM is generated by the microcontroller. If the proximity sensor is not activated, the robot can continue to grasp and finish the job. Otherwise the corresponding servo stops and the robot can continue and finish the job when the sensor is deactivated.

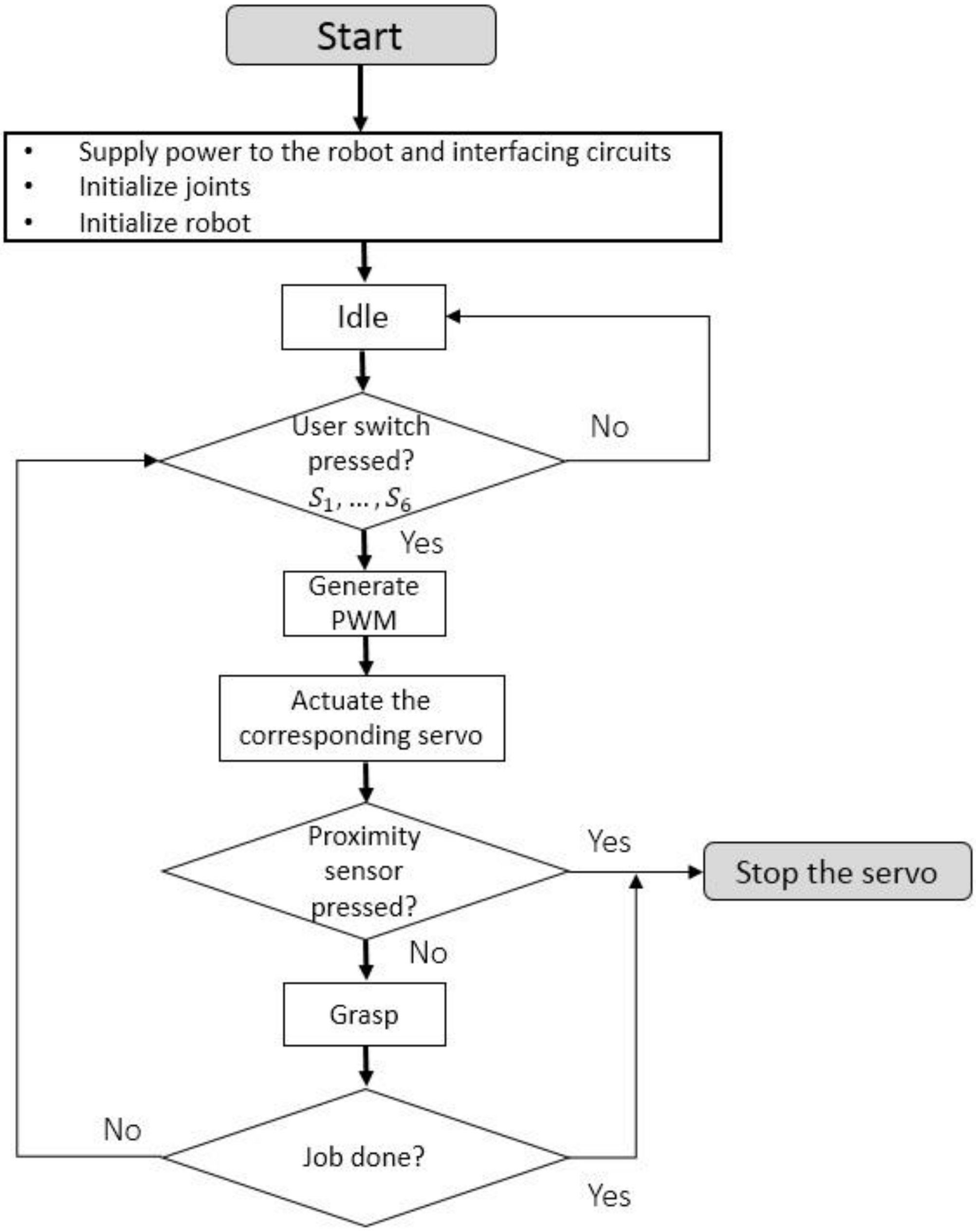


Figure 26. Robotic Arm Flowchart

All the procedure shown in the flowchart are processed in the microcontroller. The microcontroller is programmed to meet the user requirements. It can be programed to make decision by engaging artificial intelligence (AI) or it can be programmed to be dependent to a user.

There are different approaches to program a microcontroller. There are variety of programming languages and compilers such as assembly, basic, and C++. In this study, PIC BASIC PRO (PBP) compiler is used to generate HEX files for programing the microcontroller.

When the HEX code is ready, it can be written on the EPROM of the microcontroller [36]. The most common commands of the PBP language are PULSIN, PULSOUT, SERIN, SEROUT [37]. These commands basically generate pulses as an output or can store the value of a pulse in the memory of the microcontroller as the input for the next steps.

The microcontroller is connected to a PC via a PIC programmer. Figure 27 shows the PIC programmer loaded with PIC 16F877A. PIC programmer has a specific software which loads and writes the HEX file on the microcontroller. The PIC programmer used in this study is a NUP-113 version 3.0. Figure 28 shows the interfacing the microcontroller with PC via the PIC programming device and the programming software.

When the microcontroller is programmed, it is removed from the programmer and placed on its proper location on the bread board to actuate the robot. This step should be done carefully otherwise inappropriate connections on the pins may damage the microcontroller which in this case a new microcontroller is required.

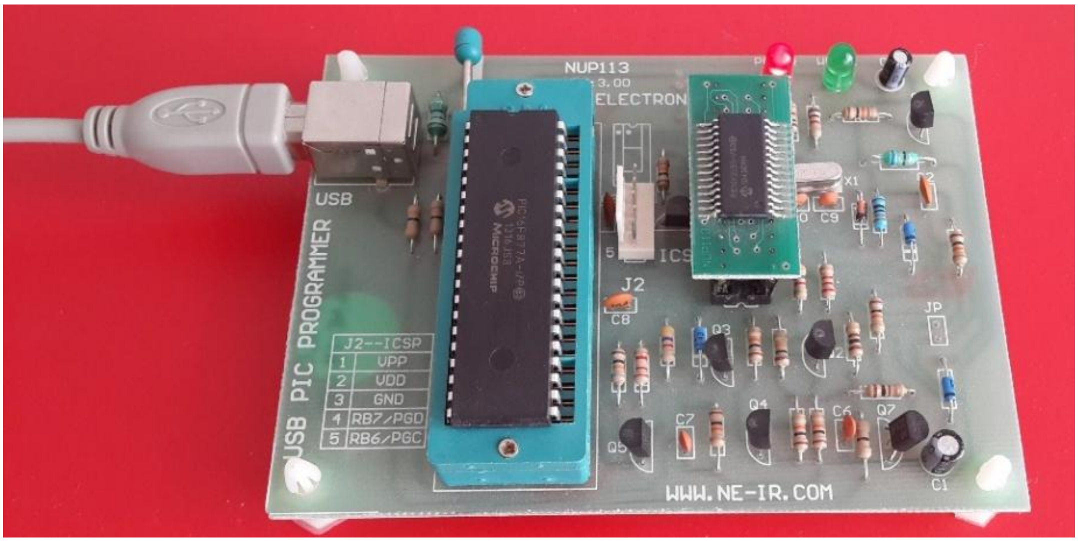


Figure 27. NUP113 PIC programmer loaded with microcontroller

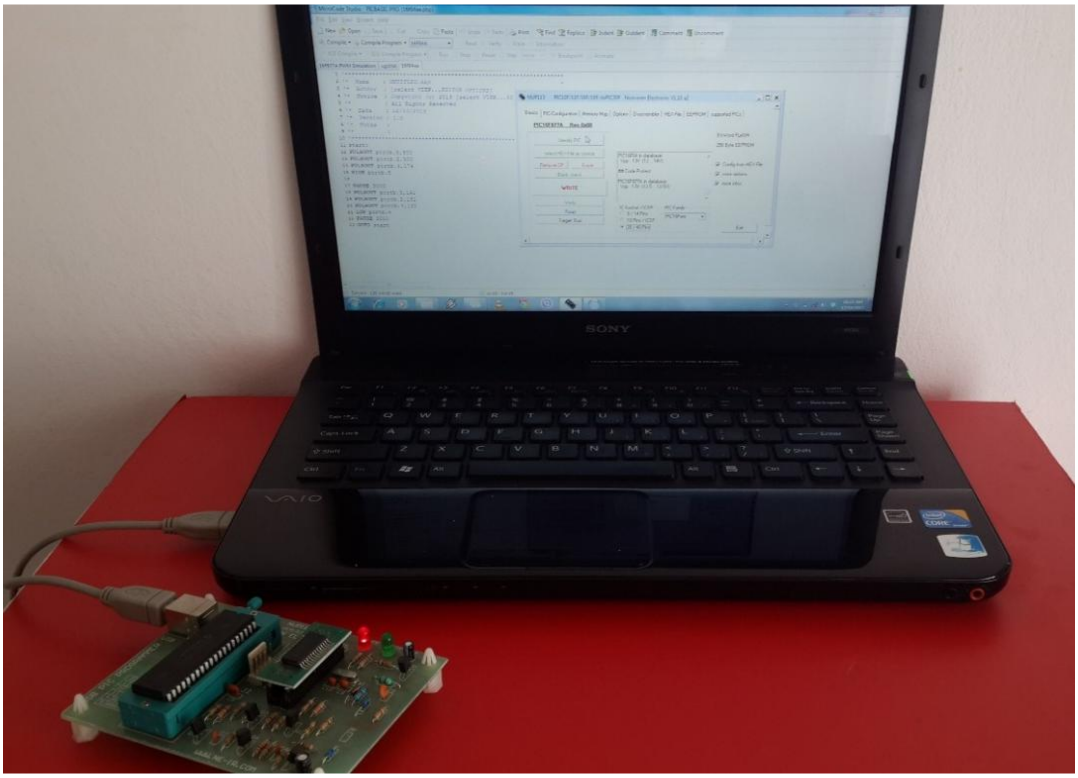


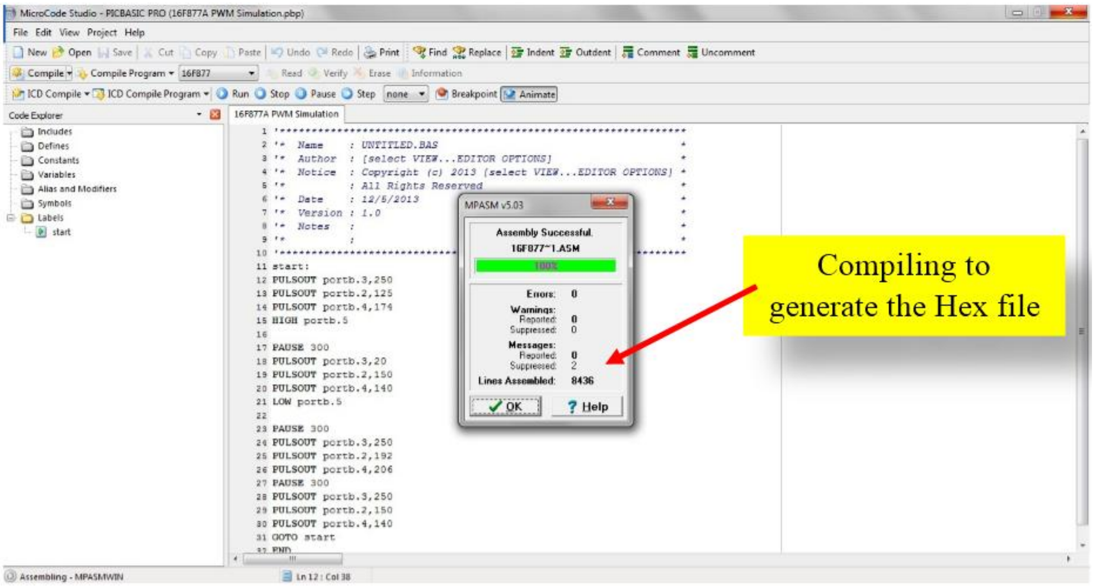
Figure 28. Interfacing with the microcontroller with PC via the PIC programmer

Chapter 4

SIMULATION AND EXPERIMENTAL RESULTS

4.1 Simulation Results and Discussions

A virtual circuit containing PIC 16F877A microcontroller and Tower-pro SG-5010 and Tower-pro SG-90 are designed in Proteus in order to simulate and analyze the results. Figure 29 and 30 show the simulation steps in order to compile the program and write it on the microcontroller.



```
'****************************************************************
'*  Name    : UNTITLED.BAS                                      *
'*  Author  : [select VIEW...EDITOR OPTIONS]                    *
'*  Notice  : Copyright (c) 2013 [select VIEW...EDITOR OPTIONS] *
'*          : All Rights Reserved                               *
'*  Date    : 12/5/2013                                         *
'*  Version : 1.0                                               *
'*  Notes   :                                                   *
'*          :                                                   *
'****************************************************************
start:
PULSOUT portb.3,250
PULSOUT portb.2,125
PULSOUT portb.4,174
HIGH portb.5

PAUSE 300
PULSOUT portb.3,20
PULSOUT portb.2,150
PULSOUT portb.4,140
LOW portb.5

PAUSE 300
PULSOUT portb.3,250
PULSOUT portb.2,192
PULSOUT portb.4,206
PAUSE 300
PULSOUT portb.3,250
PULSOUT portb.2,150
PULSOUT portb.4,140
GOTO start
END
```

Assembly Successful.
16F877~1.ASM
100%

Errors: 0
Warnings:
Reported: 0
Suppressed: 0
Messages:
Reported: 0
Suppressed: 2
Lines Assembled: 8436

Figure 29. Microcode studio compiles the Hex. file

Since there are two types of servos with different specifications in the arm robot, the results obtained from the simulation on Proteus differ in terms of pulse width, angle and servo motor behavior. According to Table-2 and Table-3, the simulation considers the rotation angle with respect to the generated pulses with the pulse width ranges from 600µs to 2400µs for Tower-Pro SG-5010, and from 400 µs to 2400 µs for Tower-Pro SG-90 servo.

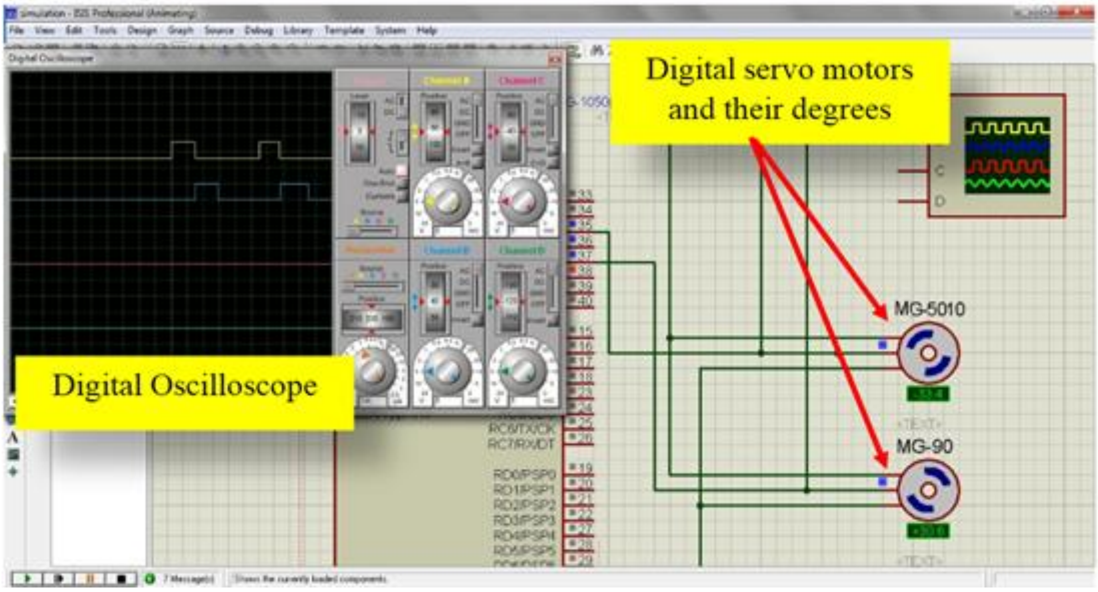


Figure 30. Servo PWM simulation using Proteus

From the experiment and simulation, it is obtained that for SG-5010 which has pulse width range from 0.6ms to 2.4ms, for every 10µs pulse width, the angle of the servo changes with 1° and for SG-90 servo type every 10µs pulse width changes the angle with 0.9°. According to this results, two formulas for each servo are proposed in order to find the necessary PWM signal to rotate the servo with a desired angle as follows.

$$S_1 = \frac{(\theta_1 + 150)}{100} \tag{4-1}$$

$$S_2 = \frac{\left(\theta_2/_{0.9}\right) + 140}{100} \tag{4-2}$$

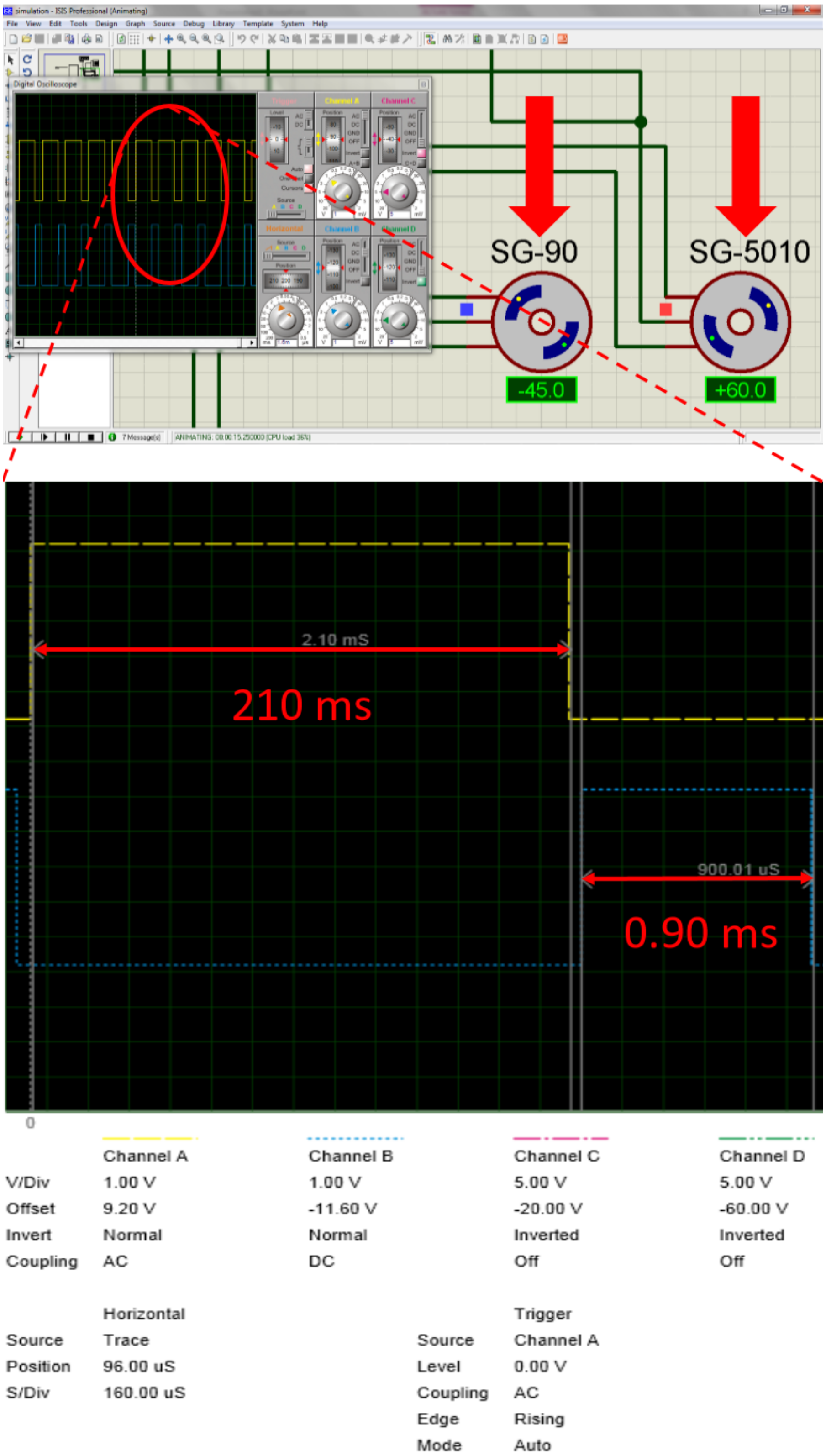
where S_1 indicates pulse width signal of SG-5010 servo, S_2 indicates pulse width signal of SG-90 servo in ms and θ is the desired angle between $-90° \leq \theta \leq 90°$ in which the servo is expected to rotate. Let's consider the following example where it is preferred to rotate SG-5010 to +60º and SG-90 to -45º. Considering (4-1) and (4-2) equations we obtain that 2.10ms pulse is required for S_1 and for S_2 a pulse with 0.90ms width is needed.

In order to verify the answer of (4-1) and (4-2) equations, simulation results can be examined by considering the degree of the servo and measuring the corresponding PWM signal for each servo motor by an oscilloscope. It should be mentioned that the microcontroller in the simulation is operating with 4MHz crystal as the real time experiment.

It was obtained that for the rotation of +60º on the SG-5010 servo motor, a pulse with the width of 2.10ms is required. For -45º rotation on SG-90 servo motor, 0.90ms pulse width is required to meet the specifications.

As shown in Figure 31, both servos have the rotation with the angle which meet our specifications. SG-90 servo motor is showing -45 value which means it has rotated to -45° with the measured PWM signal of 0.90ms as the expectations.

Also SG-5010 is showing the value of +60 which means the servo motor rotation is done with +60° with the measured PWM signal of 2.10ms. As seen in Figure 31, the PWM signals generated by the microcontroller have the same values as obtained by the formula with their expected rotation angle.



	Channel A	Channel B	Channel C	Channel D
V/Div	1.00 V	1.00 V	5.00 V	5.00 V
Offset	9.20 V	-11.60 V	-20.00 V	-60.00 V
Invert	Normal	Normal	Inverted	Inverted
Coupling	AC	DC	Off	Off

	Horizontal
Source	Trace
Position	96.00 uS
S/Div	160.00 uS

	Trigger
Source	Channel A
Level	0.00 V
Coupling	AC
Edge	Rising
Mode	Auto

Figure 31. Verifying the equation result by simulation

By considering the accuracy and comparisons of the results and simulation, a general formula for any servo can be derived to find the required PWM input signal.

$$PWM_n = \frac{\left(\theta/\varepsilon_n\right)+\delta_n}{100} \tag{4-3}$$

For

$$\alpha \leq \theta \leq \beta \tag{4-4}$$

where ε_n is the servo rotation degree with 10µs, δ_n is the center pulse of the servo and $\beta = -\alpha = 180°$

This means that 2.10ms pulse width for SG-5010 and 0.90ms pulse width for SG-90 are required to rotate the servo motors to the desired degrees mentioned above. Figure 32 shows the pulse versus their angles for both of the servo motors.

As seen in the Figure 32, both servos have a stable response to their PWM signals. This stability causes the rotation angle of the servo motors to increase properly with the increase of the PWM signals. In Figure 32, the rotation degree is considered from -90° to +90° as the specifications of the servo motors suggest.

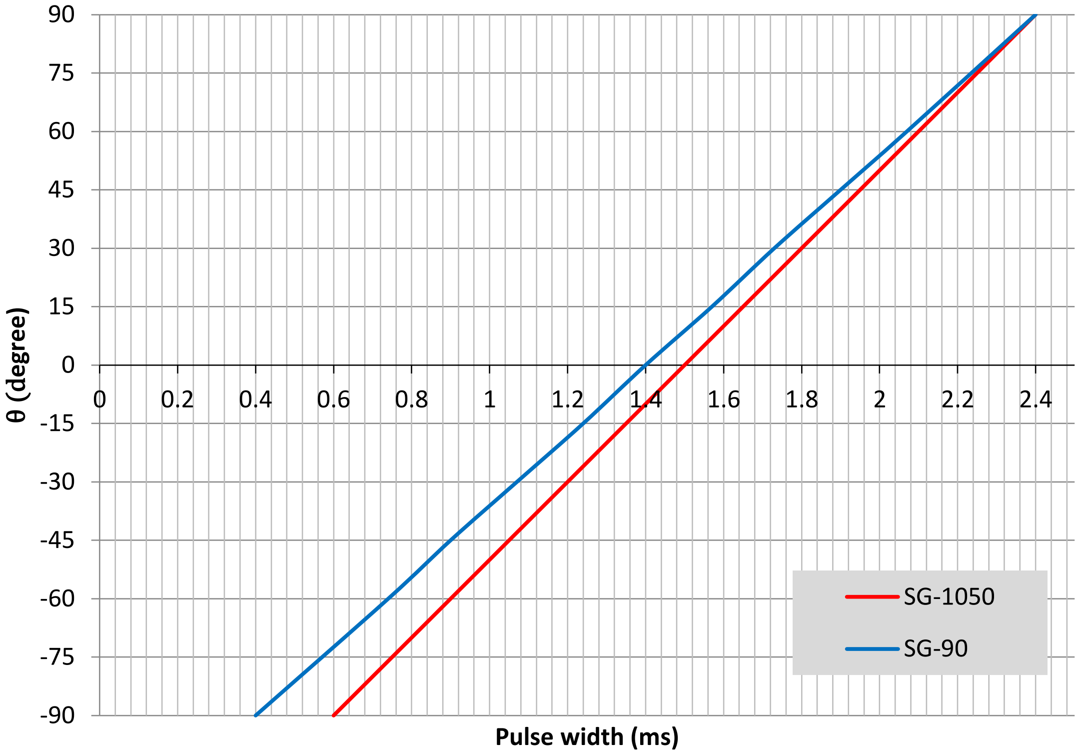
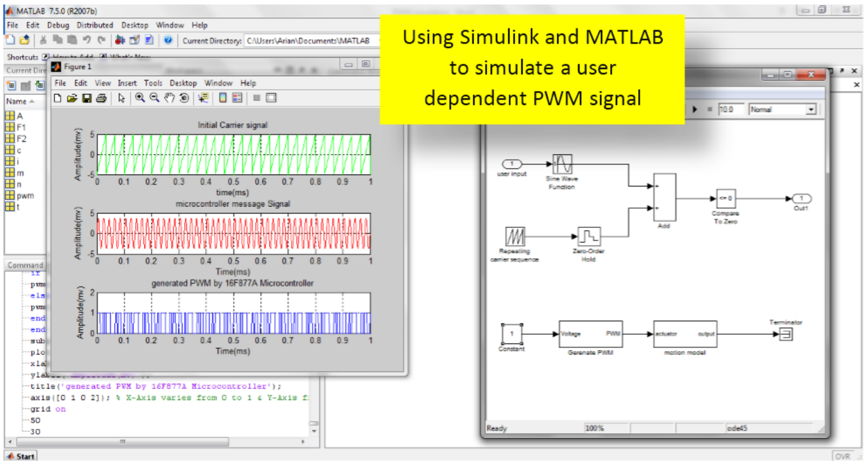


Figure 32. Servo motors pulses vs. their angles

According to the results obtained from Proteus simulation, a simulation of a PWM wave can be illustrated using MATLAB and Simulink as shown in Figure 33 and 34. Here a typical example is shown to illustrate how the microcontroller converts the sinusoidal signals into PWM signals by the carrier signal.

The user is asked to enter the values for the carrier frequency and microcontroller message frequency. The aim of this simulation is to present a better understanding of how the PWM signals are generated.



Using Simulink and MATLAB to simulate a user dependent PWM signal

```
title('generated PWM by 16F877A Microcontroller');
axis([0 1 0 2]); % X-Axis varies from 0 to 1 & Y-Axis f
grid on
50
30
```

Figure 33. Simulating PWM wave with MATLAB and Simulink

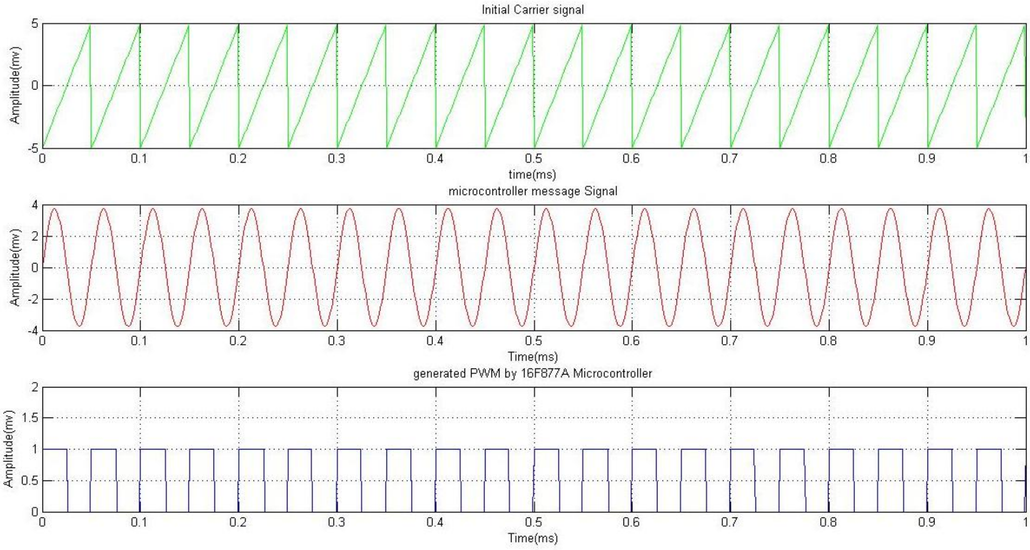


Figure 34. Simulation of PWM generated by microcontroller

The obtained results are compared with the results that Haidar, Benachiaba and Zahir obtained in their paper [10]. Since there are some specific angles mentioned in their job, our simulation is also to determine the PWM in those specific angles. The comparison is done by considering the PWM signals required to rotate the servo motor to the specific angle as shown in Table 4.

Table 4: Comparison of the servo motors in term of PWM

Rotation angle(θ)	SG-5010 PWM(ms)	SG-90 PWM(ms)	Servo Motor in [10] PWM(ms)
90°	2.40	2.40	2.35
80°	2.30	2.29	1.95
24°	1.74	1.67	2.80
17°	1.67	1.59	1.65
-30°	1.20	1.07	1.10
-40°	1.10	0.95	0.75
-60°	0.90	0.74	0.80
-90°	0.60	0.40	0.55

From the comparisons shown in the Table 4, a general comparison in terms of rotation angle and PWM signals can be shown as a graph as illustrated in Figure 35. It is clear from Figure 35 that there exists an instability in the servo motor presented in [10] for -60° to -30°, 150° to 60° and 65° to 90°. On the other hand, there is no such instability problem with the servo motors (SG-5010 and SG-90) used in this study.

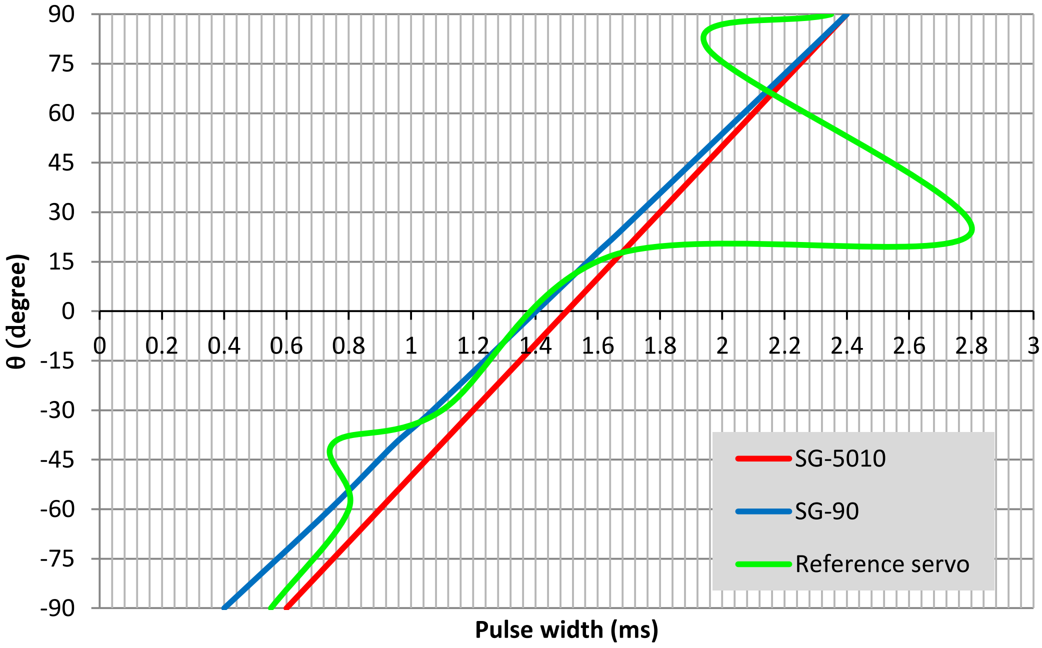


Figure 35. Servo PWM comparison

The robot arm developed by Haidar, Benachiaba and Zahir [10] does not contain any control system such as proximity sensor.

4.2 Experiment Results and Discussions

The general behavior of the system was simulated on Proteus simulation software and the PWM of the servos are shown on a digital oscilloscope. Because of the microcontroller’s large area of RAM and internal EEPROM it makes it suitable for real-time and monitoring applications [38]. Since the output signal is obtained with the help of PWM, the lower order harmonics are minimized [39]. Most microcontrollers have built-in PWM modulators with fixed resolution of 8 or 10 bits [40].

In this section, the values of the simulation is compared with the values which are obtained from the experiments. According to the experiments, the values of the simulation may slightly differ with the values of the real time experiment.

This is because, the values of the simulation are obtained in an ideal platform where, the servo and microcontroller never operate with an error. Figure 36 shows the real-time experiment on SG-90 and SG-5010 where the circuit is implemented on bread board.

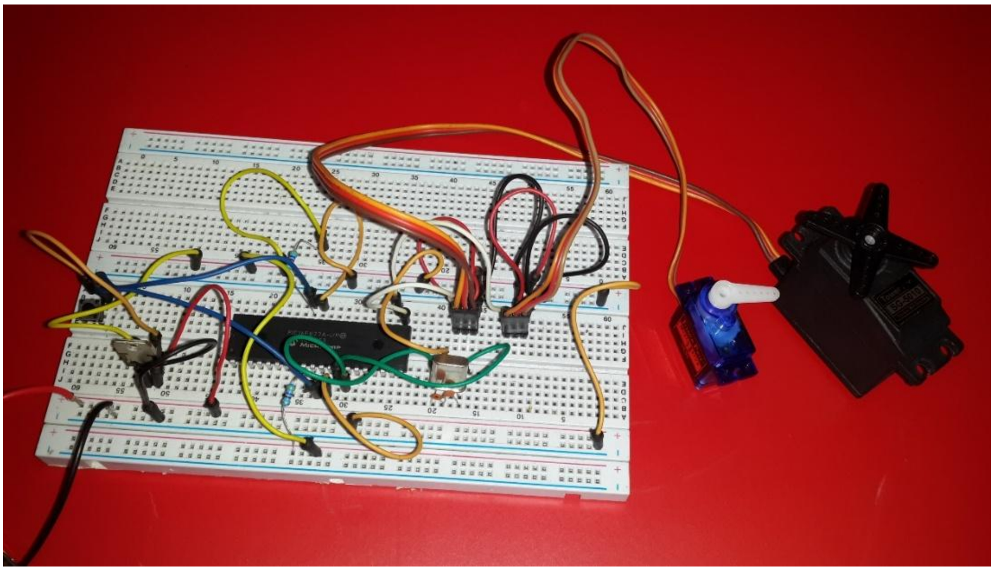


Figure 36. SG-90 and SG-5010 PWM experimentation

In the simulation the servo operates within the exact pulse cycle and pulse width boundaries. However, based on experiments, a manufactured plastic gear servo may not be accurate as determined by the experiments. Also a manufactured PIC 16F877A microcontroller’s built-in oscillator (clock) may not have an exact value of 1 MHz. These facts lead to have a different results than the simulation. In order to overcome these problems, the robotic arm is designed in a way that the rotation of the servos are based on the user command via the controlling switches.

Since we want to have the full rotation at each servo with respect to the activated switches, the generated PWM signals are chosen in a way to give a clockwise and counter-clockwise motions. In this way, the robotic arm can be controlled simultaneously with the switches to actuate the servo motors.

In section 4.1, the generated PWM signals were measured using a simulation and a virtual oscilloscope. In the following, the PWM signals are measured with an oscilloscope in the laboratory as shown in Figure 37 for a full rotation of a servo motor in order to compare and verify the results of the simulation. The microcontroller was programmed to generate 50µs (or 0.05ms) pulse. The value is very smaller than the minimum width control pulse of the servo as shown in Table 2. Therefore, it causes the continuous rotation at full speed.

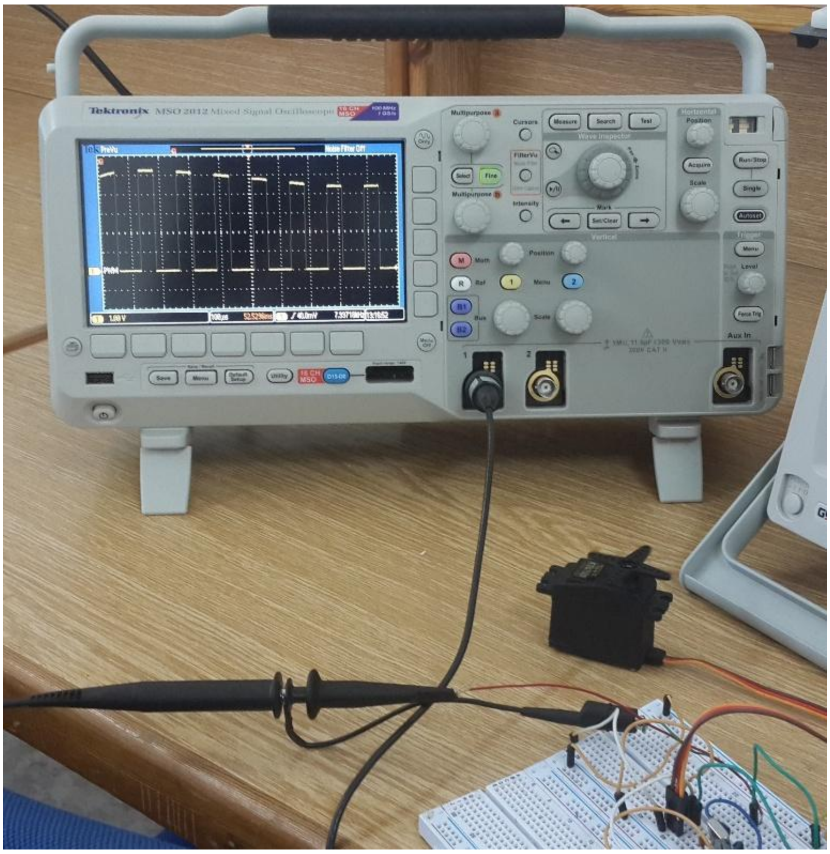
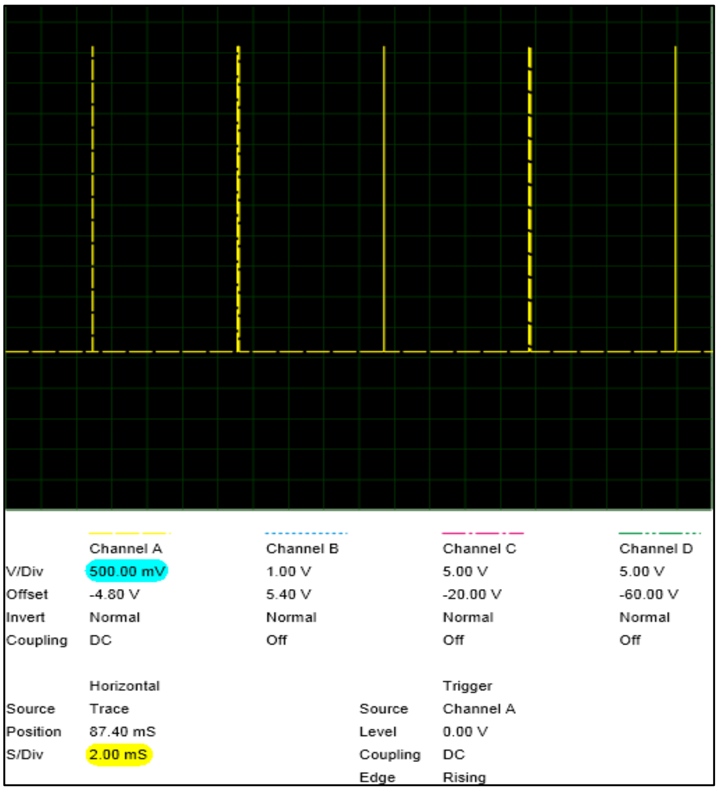


Figure 37. PWM experiment results on oscilloscope

A simulation is carried out in order to obtain the results for the full rotating servo with an applied 50µs PWM signal. After the simulation, it was obtained that the peak to peak value of the PWM signal is 5.00 volts and the width of the generated PWM signals is 49.80µs (or 0.0498ms) as shown in Figure 38 and Figure 39.

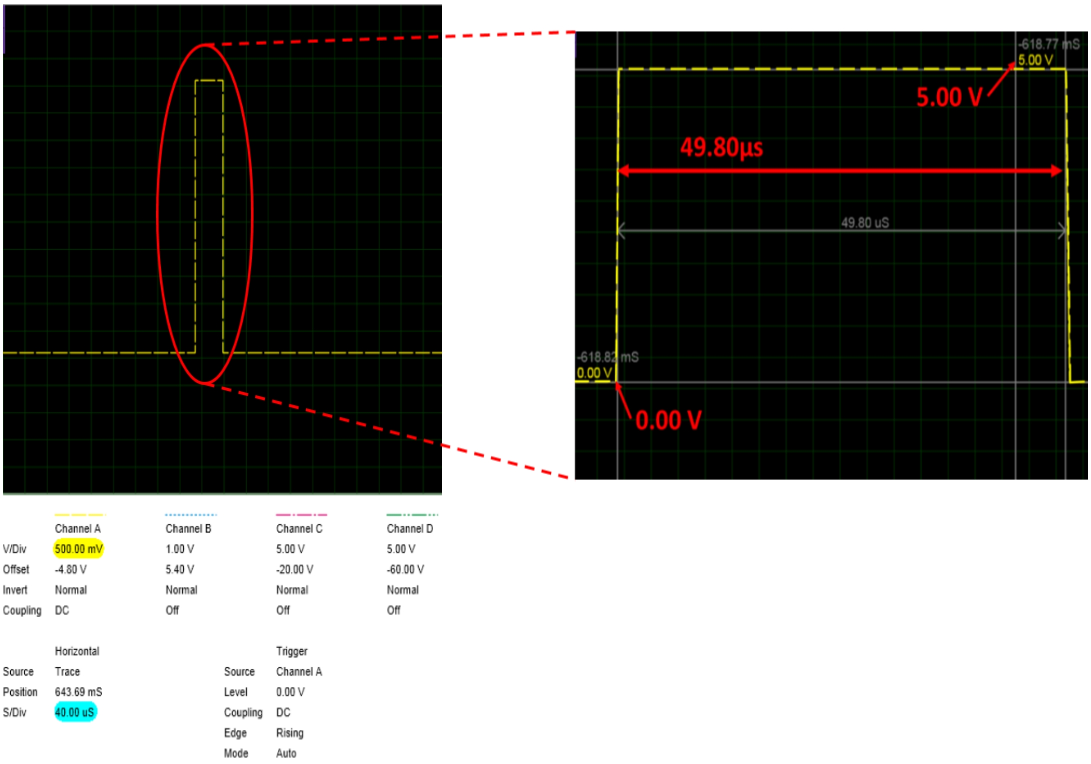


	Channel A	Channel B	Channel C	Channel D
V/Div	500.00 mV	1.00 V	5.00 V	5.00 V
Offset	-4.80 V	5.40 V	-20.00 V	-60.00 V
Invert	Normal	Normal	Normal	Normal
Coupling	DC	Off	Off	Off

	Horizontal
Source	Trace
Position	87.40 mS
S/Div	2.00 mS

	Trigger
Source	Channel A
Level	0.00 V
Coupling	DC
Edge	Rising

Figure 38. Simulation results of a full rotating servo measured by a digital oscilloscope in 200 µs (time/div)



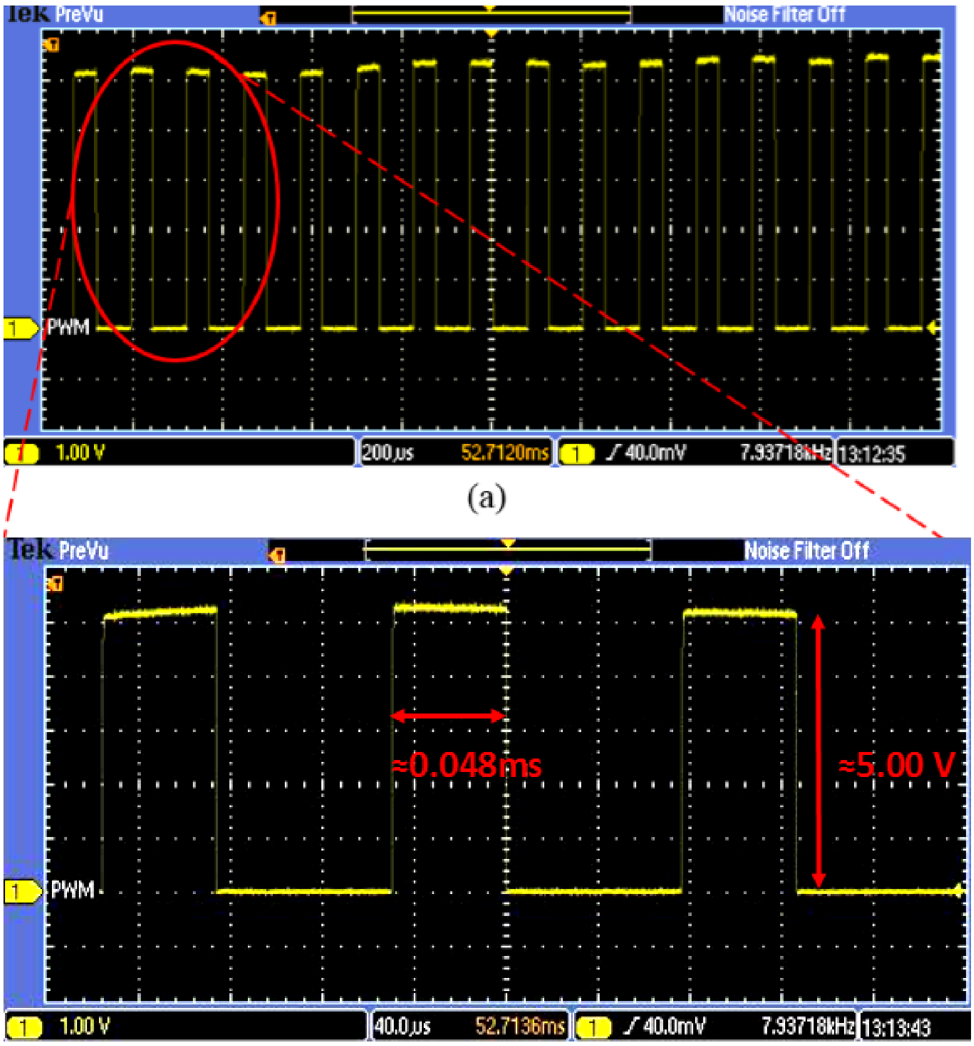
	Channel A	Channel B	Channel C	Channel D
V/Div	500.00 mV	1.00 V	5.00 V	5.00 V
Offset	-4.80 V	5.40 V	-20.00 V	-60.00 V
Invert	Normal	Normal	Normal	Normal
Coupling	DC	Off	Off	Off

	Horizontal
Source	Trace
Position	643.69 mS
S/Div	40.00 uS

	Trigger
Source	Channel A
Level	0.00 V
Coupling	DC
Edge	Rising
Mode	Auto

Figure 39. Simulation results of a full rotating servo measured by a digital oscilloscope in 40 µs (time/div)

On the other hand, the experiment obtained that the peak to peak value of the generated PWM signal is approximately 5.00 volts and the pulse width of the generated PWM signal is close to 48µs (or 0.048ms) as shown in Figure 40.



(a)

(b)

Figure 40. Experiment results of a full rotating servo measured by an oscilloscope
(a) 200 µs (time/div) (b) 40 µs (time/div)

By comparing the experimental results with the simulation results, it is obtained that both have almost the same values with a minor difference. This difference can be used to define the accuracy of equation 4-3 as follows. If we consider the applied PWM signal of 49.80µs by the simulation as the full accuracy percentage and the obtained PWM signal by the experiment (48µs) as the portion of it, the accuracy of the proposed formula can be obtained as almost 97%, this means that the rotation of the servo can possibly have 3% error comparing to the expectations.

Chapter 5

CONCLUSION AND FUTURE WORK

In this thesis, the procedure of building an articulated arm robot using a microcontroller and servo motors with the help of PWM has been discussed. The building procedure consists of building the kinematic structure of robot, hardware design and implementation, software design and implementation and microcontroller programming.

The motion of the robot is controlled via PWM signals which are generated with the microcontroller and discussed the effect of these pulses on the servo motors using both software simulation and real-time experiment. Also, a general formula which facilitates the design and motion control of the robots has been proposed. The values obtained from the simulation and experiments were compared and the comparison results were shown using graphs.

It is important to recall that each servo motor has a different pulse width range where a pulse can have different effects on each servo, therefore it is important to first simulate the system and then implement it accordingly. It is also essential to consider the built-in oscillator of the microcontroller which is usually 1MHz. Different values of the oscillator can cause the system to not operate properly.

As the future work of the developed arm robot, a voice-controlled robot can be considered where the user can simply control the robot by giving voice commands such as giving directions (up, down, left, right) and actions (grasp, drop, rotate left, and rotate right).

REFERENCES

[1] Lin, I. H., Liu, C. Y., Chen, L. C., "Evaluation of Human-Robot Arm Movement Imitation", Proceedings of 8th Asian Control Conference (ASCC), pp.287-292, 2011.

[2] Nikku, S., “Introduction to Robotics” USA: John Wiley & Sons, 2011.

[3] http://www.robotics.org/content-detail.cfm/Industrial-Robotics-Featured-Articles/How-Robots-Will-Affect-Future-Generations/content_id/834

[4] http://machineslikeus.com/news/potential-future-impact-robots-society

[5] http://curiosity.discovery.com/question/how-robots-improve-lives

[6] Nehmzow, U., ”Mobile Robotics: a practical introduction”, UK: Springer, 2003.

[7] Muir, F. P., Neuman, P. C., "Pulsewidth Modulation Control of Brushless DC Motors for Robotic Applications", IEEE Transactions on Industrial Electronics, Issue 3, Vol. 32, pp.222-229, 1985.

[8] Kramer, A. K., Anderson, M., "A PIC18 Robot-Centered Microcontroller Systems Laboratory", IEEE Frontiers in Education Conference (FIE), pp.1-6, 2011.

[9] Hao, G. W., Leck, Y. Y., Hun, C. L., "6-DoF PC-Based Robotic Arm (Pc-Robotarm) With Efficient Trajectory Planning And Speed Control", International Conference on Mechatronics (ICOM), pp.1-7, 2011.

[10] Haidari, A. M. A., Benachiba, C., Zahir, M., “Software Interfacing of Servo Motor with Microcontroller”, Journal of Electrical Systems (JES), pp.84-99, 2013.

[11] International federation of robotics (IFR), “World Robotics”, Switzerland: United Nation, 2005.

[12] Khatib, S., “Springer Handbook of Robotics”, USA: Springer-Verlag Berlin Heidelberg, 2008.

[13] Gieras, F. J., Piech, J. Z., Tomczuk, I. “Linear Synchronous Motors: Transportation and Automation Systems, Second Edition” USA: Taylor and Francis Group,LLC, pp.407-413, 2012.

[14] Kuttan, A., “Robotics”, India: I.K. International publishing house, 2007.

[15] Sanchez, S. P., Cortes, R. F., “Cartesian Control for Robot Manipulators, Robot Manipulators Trends and Development”, 2010.

[16] Saha, S. K., “Introduction to Robotics”, India: Tata McGraw-Hill publishing company Ltd., 2008.

[17] Mathia, K., “Robotics for Electronics Manufacturing: Principles and Applications in Cleanroom Automation”, UK: Cambridge University Press, 2010.

[18] Tsai, L. W., “Robot Analysis: The Mechanics of Serial and Parallel Manipulators” Canada: John Wiley & Sons, 1999.

[19] Merlet, J. P., “Parallel Robots: Second Edition”, Netherlands: Springer, 2006.

[20] Zhang, D., “Parallel Robotic Machine Tools”, Canada: University of Ontario Institute of Technology (UOIT), pp.1-5, 2010

[21] Gogu, G., “Structural Synthesis of Parallel Robots: Part 4: Other Topologies with Two or three degree of freedom”, France: France Institute of advanced mechanism, pp.1-10, 2012.

[22] Nof, Y. S., “Handbook of industrial robotics, volume 1”, Canada: John Wiley & Sons, 1999.

[23] Bagad, V. S., “Mechatronics” India: Technical Publication Pune, 2008.

[24] Roa, P. N., “CAD/CAM Principles and Applications 3rd Edition”, India: Tata McGraw Hill Education private Ltd., 2010.

[25] Kabuka, R. M., Glaskowsky, N. P., Miranda, J., "Microcontroller-Based Architecture for Control of a Six Joints Robot Arm”, IEEE Transactions on Industrial Electronics, Issue 2, Vol. 35, pp.217-221, 1988.

[26] Park, I. D., Park, C., Do, H., Choi, T., Kyung, J., "Design And Analysis of Dual Arm Robot Using Dynamic Simulation" IEEE 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 681-682, 2013

[27] http://jjshortcut.wordpress.com/2012/04/09/cleanup-robotic-arm-parts/

[28] Manjunath, T. C., “Kinematic Modeling and Maneuvering of A 5-Axis Articulate Robot Arm” world academy of science, engineering and technology, pp.363-369, 2007.

[29] Nones, Y. A., “Heterogeneous Modeling & Design of a Robot Arm Control System”, Illinois Tech Robotics (ITR), pp.1-6, 2003.

[30] Vajnberger, V., Terzimehić, T., Silajdžić, S., Osmić, N., "Remote Control of Robot Arm with five DOF", MIPRO Electronics, pp.1707-1711, 2011.

[31] Ibrahim, D., “30 Projects Using PIC Basic and PIC Basic Pro”, UK: MPG books Ltd., 2006.

[32] Seki, K., Yokoi, H., Iwasaki, M., “Experimental Evaluations of Friction Behavior in Micro-Displacement Region Positioning for Servo Motor with Air Bearings” Proceeding of IEEE International Conference on Advanced Intelligent Mechatronics, pp.731-736, 2012.

[33] Bejo, A., Pora, W., Kunieda, H., "Development of a 6-Axis Robotic Arm Controller Implemented on a Low-Cost Microcontroller", " IEEE 10th

International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp.328-331, 2009.

[34] www.servodatabase.com/servo/towerpro/sg-5010

[35] http://www.servodatabase.com/servo/towerpro/sg90

[36] Ahuja, N., Sreedaran, R., "Computer Controlled Robotic Arm", IEEE Computer Society, Symposium on Computer-Based Medical Systems (CBMS), pp.361-366, 2003.

[37] MircoEngineering Labs, “PicBasic Pro Compiler”, microEgnineering Labs, Inc., 2008.

[38] Şahin, S., İşler, Y., "Microcontroller-Based Robotics and SCADA Experiments", IEEE Transactions on Education, Issue 4, Vol. 56, pp.424-429, 2013.

[39] Govil, V. K., Charuasia, Y., “Modeling and Simulation of PWM controlled Cycloconverter FED Split Phase Induction Motor” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE), Issue 3, Vol. 1, pp.126-133, 2012.

[40] Hung, C. P., Liu, W. G., Su, H. J., Chen, J. W., Chiu, B. M., “PIC-Based Muclti-Channel PWM Signal Generation Method and Application to Motion Control of Six Feet Robot Toy” International Journal Of Circuits, Systems And Signal Processing, Issue 2, Vol. 3, pp.73-81, 2009.

APPENDIX

Pic Basic Pro codes for the robotic arm:

```
'****************************************************************
'*  Name    : UNTITLED.BAS                                      *
'*  Author  : [Arian Faravar]                                   *
'*  Notice  : Copyright (c) 2014 [Computer Engineering - E.M.U] *
'*          : All Rights Reserved                               *
'*  Date    : 1/21/2014                                         *
'*  Version : 1.0                                               *
'*  Notes   :                                                   *
'*          :                                                   *
'****************************************************************
s1r var porta.0
s1l var porta.1
s2r var porta.2
s2l var porta.3
s3r var porta.4
s3l var porta.5
s4r var portb.0
s4l var portb.1
s5r var portb.2
s5l var portb.3
s6r var portb.5
s6l var portb.6

a var BYTE
a=150
TRISD=0
PORTB=0
low portd.0
low portd.1
low portd.2
low portd.3
low portd.4
low portd.5

start:
pulsout PORTD.0,a
if s1r=0 then gosub left1
if s1l=0 then gosub right1
return

pulsout PORTD.1,a
if s2r=0 then gosub left2
if s2l=0 then gosub right2
return

pulsout PORTD.2,a
```

```
if s3r=0 then gosub left3
if s3l=0 then gosub right3
return

pulsout PORTD.3,a
if s4r=0 then gosub left4
if s4l=0 then gosub right4
return

pulsout PORTD.4,a
if s5r=0 then gosub left5
if s5l=0 then gosub right5
return

pulsout PORTD.5,a
if s6r=0 then gosub left6
if s6l=0 then gosub right6
goto start
End

right1:
if a=250 then start
a=a+1
pause 50
goto start

left1:
if a=50 then start
a=a-1
pause 50
goto start

right2:
if a=250 then start
a=a+1
pause 50
goto start

left2:
if a=50 then start
a=a-1
pause 50
goto start

right3:
if a=250 then start
a=a+1
pause 50
```

```
goto start

left3:
if a=50 then start
a=a-1
pause 50
goto start

right4:
if a=250 then start
a=a+1
pause 50
goto start

left4:
if a=50 then start
a=a-1
pause 50
goto start

right5:
if a=250 then start
a=a+1
pause 50
goto start

left5:
if a=50 then start
a=a-1
pause 50
goto start

right6:
if a=250 then start
a=a+1
pause 50
goto start

left6:
if a=50 then start
a=a-1
pause 50
goto start
```