Sudan University of Science and Technology

College of Engineering

Department of Biomedical Engineering

A project submitted in partial fulfillment for the degree of B.Sc. in biomedical engineering

Design and Simulation of Mobile Robotic Arm Controller Using Microcontroller and Cell Phone

تصميم ومحاكاة حاكمة الذراع الالي المتنقل باستخدام المتحكم الدقيق والهاتف السيار

By :

Sami Siddig Nour Aldien

Osama khalid Fadl Elmwla

Abd Algani Mohammed Alhassen

Supervisor:

Dr. Eltahir Mohammed Hussein

Oct 2015

بسم الله الرحمن الرحيم

قال تعالى:

(وَلَقَدْ خَلَقْنَا الإنسان مِن سلالة مِّن طِينٍ* ثُمَّ جَعَلْنَاهُ نُطْفَةً في قَرَارٍ مَّكِينٍ* ثُمَّ خَلَقْنَا النُّطْفَةَ عَلَقَةً فَخَلَقْنَا الْعَلَقَةَ مُضْغَةً فَخَلَقْنَا الْمُضْغَةَ عِظَاماً فَكَسَوْنَا الْعِظَامَ لَحْماً ثُمَّ أَنشَأْنَاهُ خَلْقاً آخَرَ فَتَبَارَكَ اللَّهُ أَحْسَنُ الخالقين* ثُمَّ إِنَّكُمْ بَعْدَ ذلِكَ لَمَيِّتُونَ* ثُمَّ إِنَّكُمْ يَوْمَ الْقِيَامَةِ تُبْعَثُونَ* وَلَقَدْ خَلَقْنَا فَوْقَكُمْ سَبْعَ طرائق وَمَا كُنَّا عَنِ الْخَلْقِ غافلين)

صدق الله العظيم

سورة المؤمنون الاية (12-17)

Dedication

With all love, we dedicate this work to our precious mothers and fathers, may Allah give them peace and health, our grateful thanks goes to Mr. Obada abass

Acknowledgment

Praise and thanks to Allah almighty, who gave us the ability to complete this research. We would like to acknowledge with much appreciation the crucial role of the staff of Sudan University of Science and Technology, who gave the full support to complete this research.

A special gratitude we give to our supervisor**, Dr. Eltahir Mohamed Hussein**, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this research.

Table of Contents

List of Figures

List of Tables

List of abbreviations

Abbreviations	Meaning
DTMF	Dual Tone Multi-Frequency
PIC	Peripheral Interface Controller
GSM	Global System for Mobile
FET	Field Effect Transistor
MCU	Micro Controller Unit
DC	Direct Current
COM	COMmon
EMF	Electro-Magnetic Field
CPU	Central Processing Unit
RF	Radio Frequency
GND	Ground
DOF	Degree Of Freedom
ESt	Early Steering flag
IC	Integrated Circuit
LCD	Liquid Crystal Display

Abstract

The objectives of design this robot is simply to help the humans in the future for security purpose. In the present scenario, there are many recent developments of robotics and communication on a large scale. The methodology used in this research is Dual Tone Multi-Frequency (**DTMF**) and PIC microcontroller. Robotic arm was controlled by a cell phone and it can be communicate on a large scale over a large distance even from different cities.

Furthermore there are number of advantages as well as important features such as accuracy when receiving transmitted signals from cell phone and movement in all directions. This robot can also be used to reach the places where humans cannot reach such as small tunnels, dangerous places and also can be used to transport of hazardous material combustible etc. This is the way in which the robot can be used in different fields as well as for research purpose by further manipulation in programming and it can be modified accordingly. Finally the simulation results obtained were acceptable using **Mikroc** compiler and **Proteus** simulation program.

المستخلص

تناولت هذه الدراسة تصميم وبرمجة نظام تحكم في حركة الذراع الالي باستخدام تقنية المتحكم الدقيق وجهاز الهاتف السيار والذي يستخدم لإرسال إشارات التحكم في الحركه حيث يتم إستقبالها عن طريق هاتف آخر مثبت على هيكل يمثل المستقبل ، تم استخدام دائرة **"DTMF"** لتحليل الإشارات التماثلية المرسلة وتحويلها إلى إشارات رقمية تتم معالجتها عن طريق المتحكم الدقيق.

استخدم المتحكم الدقيق كوحدة معالجة مركزية تحتوي على برنامج التحكم في المحركات المختلفة المثبتة على هيكل الذراع ، يمكن استخدام النظام في الوصول الى المناطق المحفوفة بالمخاطر والتى لا يستطيع الانسان الوصول اليها كالانفاق والممرات الضيقه ومناطق الاشعاع ، كما يمكن استخدامه في نقل المواد الكيميائية القابلة للإشتعال بصورة آمنة بعيدا عن المستخدم.

استخدم برنامج **"MIKROC compiler"** لكتابة برنامج التحكم كما استخدم برنامج المحاكاة " **Proteus** " لتوضيح نماذج الحركة المختلفة وكانت النتائج المتحصل عليها مقبولة.

Chapter One

Introduction

1-1 General View

A robotic arm is a type of mechanical arm, usually programmable, with similar function to a human arm; the arm may be the sum total of the mechanism or may be part of a more complex robot.

The end effector, or robotic hand, can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application. For example robot arms in automotive assembly line perform a variety of tasks such as welding and parts rotation and placement during assembly. In some circumstances, close emulation of the human hand is desired, as in robots designed to conduct bomb disarmament and disposal.

Robotics requires a working knowledge of electronics, machines and software, and is usually accompanied by a large knowledge of many subjects.

A robot is a computer controlled machine that is programmed to move, manipulate objects and accomplish work while interacting with its environment.

Mobile robots have the capability to move around in their environment and are not fixed to one physical location. In contrast, Industrial robots usually consist of a jointed arm (multi-linked manipulator) and gripper assembly (or end effectror) that is attacked to a fixed surface.

Robots have replaced humans in assistance of performing those repetitive and dangerous tasks with humans prefer not to do, or are unable to do due to size limitation, or even those such as in outer space or at the bottom of the sea where humans could not survive extreme environments.

There are concern about the increasing use of robots and their role in society.

Robots are blamed for rising unemployment as they replace workers in some functions; the use of robots in military combat raises ethical concerns. The possibility of robot autonomy and potential repercussions has been addressed in fiction and may be a realistic concern in the future.

1-2 Problem Statement

Conventionally, wireless controlled robots uses circuits, which have a drawback of limited working range, limited frequency range and limited control. Use of mobile phones for robotic control can overcome these limitations. It provides the advantages of robust control, working range as large as the coverage area of the service provider, no interference with other controllers and up to twelve controls.

1-3 Objectives

The main aim of this thesis is to:

- Design and programming control system based on PIC microcontroller and dual tone multi frequency (DTMF) technique to control mobile robotic arm.

- Transportation of dangerous material and flammable items in safety way.
- Placing or picking the objects that far away from the user.

1-4 Methodology:

A mobile robotic arm that functional to do pick and place operation is controlled by using DTMF technique and PIC microcontroller. It can move forward, reverse, turn right and left for specific distance according to the

Controller program and specification, the development of this robot is based on PIC microcontroller platform that will be interfaced with the mobile phone module to control the mobile robotic arm movement. This prototype of the robot is expected to overcome the problem such as placing or picking object that far away from the user, pick and place hazardous objects in the fastest and easiest way.

In this project, the hardware and software functions ware combined to make the system reliable. Mobile phone was used to send a suitable data and singles to the PIC microcontroller with help of DTMF MT8870 decoder to control movement of the robot. The project is depends on PIC microcontroller, GSM technique and MikroC compiler that used to execute the program code.

1-5 Thesis Layouts

This thesis contains six chapters. The first chapter is general introduction to the project and its goals. Chapter two contains previous studies (Reviews). Chapter three discusses the PIC microcontroller interfacing circuits and explains sensors, stepper motor and dc motor interface. Mechanical model and components of robotic arm are discussed in chapter four.

Chapter five discusses the Design and Programming of Mobile Robotic Arm. Chapter six is the last chapter of this research gives a summary of the project and some of recommendations.

Chapter Two

Reviews

1. In 2013 Puran Singh, Anil Kumar,[1]. A robot is a system combining many subsystems that interact among themselves as well as with the environment in which the robot works. In robotics, end effectors are a device at the end effectors or tool to grasp any physical thing that may be a human hand or any instrument. To achieve this goal we intend to incorporate a simple linkage actuation mechanism. An AC motor is used along with spur gears and a threaded shaft arrangement. The gripper can perform the basic function of picking, holding and grasping of object by means of a DC motor and forms the mechanism for the spot welding. The human design forms the basic of this project of developing a robotic gripper and is the source of inspiration to achieve the sufficient level of dexterity in the domain of grasping and manipulation if coupled with wrist and arm.

2. In 2013 D.A.N. Lakmal. Weerakkody, P. Samsoodeen[2] the project involves the construction of a mechanical arm which will be operated by stepper motor.

The force exerted by the hand on the object can be measured by a force sensor which will be attached to the end of an end effecter of the arm. This force sensor will provide the necessary feedback to the controller which will adjust the stepper motor to reduce or increase the gripping force exerted. The controller will be a micro-controller which will control the stepper motor and collect data from the force sensor. The requirement was to create a mechanical arm which can safely lift a landmine if necessary. The landmine has to be lifted in such away that there's no risk of it exploding. Therefore the force sensor will make sure that the force exerted is not too much.

The project essentially consists of three main parts, the mechanical construction of the arm, and construction of the drive circuit for the stepper motor and the obtaining of force profiles for the force sensor.

3. In 2013 Ashutosh pattnaik,rajiv ranjan [3]. In today's world there is an increasing need to create artificial arms for different inhuman situations where human interaction is difficult or impossible. They may involve taking readings from an active volcano to diffusing a bomb. Here we propose to build a robotic arm controlled by natural human arm movements whose data is acquired through the use of accelerometers. For proper control mechanism and to reduce the amount of noise coming in from the sensors, proper averaging algorithm is used for amoothening the output of the accelerometer. The development of this arm is based on ATmega32 and ATmega640 platform along with a personal computer for signal processing, which will all be interfaced with each other using serial communication. Finally, this prototype of the arm may be expected to overcome the problem such as placing or picking hazardous objects or non-hazardous objects that are far away from the user.

Chapter Three

PIC Microcontroller Interfacing Circuit

3-1 power outputs

The microcontroller or microprocessor port only provides a limited amount of current: about 20mA in the case of PIC, and even less for standard microprocessor ports. Therefore, if we want to drive an output device that needs more current than this, some kind of current amplifier or switch is needed.

3-1-1 current Drivers

All solid-state current driver and switches are derived from the semiconductor technology which is the basis of the transistor. The bipolar transistor was the first to be developed, and is still extensively used as it is robust and easy to design around. The FET(field effect transistor) was later developed alongside integrated circuit, because it generally has a higher input impedance and consumes less power in high-density circuit . In addition, the power FET has some distinct advantages over its bipolar equivalent, and is used extensively in motor control and similar application.

3-2 Power out puts interface

The PIC has a simple program attached, which simply switches on each output in turn when the button is pressed.

3-2-1 Motor interface

As discussed above, the basic function of a motor is to convert electrical input current into output mechanical power (torque). All use electromagnetic coils to provide this conversion, and need current switches or amplifiers to operate them from an MCU.

A simple method of controlling AC motors is to use a relay as switch. Another is to use a triac to control the current, as outlined above, but in practice there are some tricky issues associated with controlling inductive loads with thyristors and triacs which require reference to specialist texts. Three phase motors require, in simple terms, each phase to be controller by a separate device, but simultaneously, that is, three relays or triacs operated by the some controller.

Three typical small motor interfaces, a DC motor, a DC servo and a stepper motor. The motors can be operated in turn by pressing the select button. Operating parameters (speed, position, and direction) can then be changed via the additional push buttons.

3-2-2 Stepper Motor interfacing

A stepper motor is a widely used device that translates electrical pulses into Mechanical movement. In application such as disk drivers, dot matrix printers, and robotics, the stepper motor is used for position control. Stepper motors commonly have a permanent magnet rotor (also called the shaft) surrounded by a stator (see Figure 3-1). There are also steppers called variable reluctance stepper motors that do not have a permanent magnet rotor. The most common stepper motors have four stator windings that are paired with a centre-tapped common as shown in Figure 3-2. This type of stepper motor is commonly referred to as a four-phase or unipolar stepper motor. The centre tap allows a

Change of current direction in each of two coils when a winding is grounded thereby resulting in a polarity change of the stator. Notice that while a conventional motor shaft runs freely, the stepper motor shaft moves in a fixed repeatable increment, which allows one to move it to a precise position. This repeatable fixed movement is possible as a result of basic magnetic theory where poles of the same polarity repel and opposite poles attract. The direction of the rotation is dictated by the stator poles.

The stator poles are determined by the current sent through the wire coils. As the direction of the current is changed, the polarity is also changed causing the reverse motion of the rotor. The stepper motor discussed here has a total of six leads: four leads representing the four stator windings and two commons for the centre-tapped leads. As the sequence of power is applied to each stator winding, the rotor will rotate. There are several widely used sequences, each of which has a different degree of precision. Table 3-1 shows a two-phase, four-step stepping sequence.

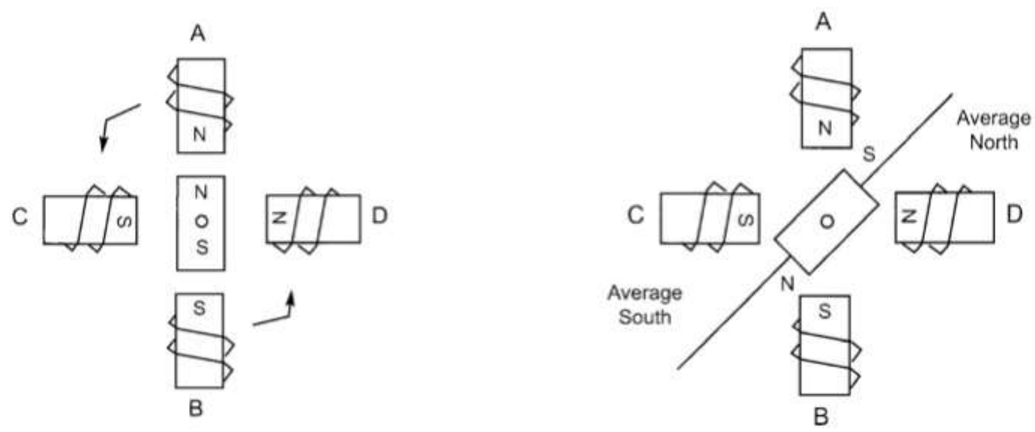

Figure 3.1 Rotor Alignment

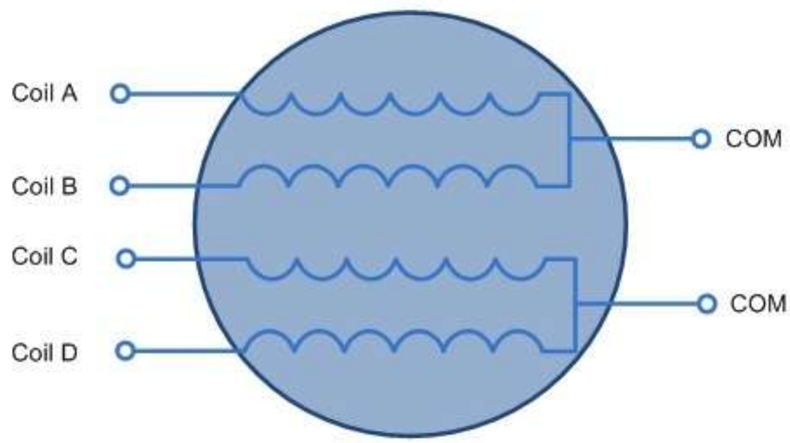

Figure 3.2 Stator winding configuration

Note that although we can start with any of the sequences in Table3-1, once we start we must continue in the proper order. For example, if we start with step 3 (0110), we must continue in the sequence of steps 4, 1, 2, etc.

Table 3.1 Normal four- step sequence[4]

step	Winding A	Winding B	Winding C	Winding D
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

Step Angle

How much movement is associated with a single step? This depends on the internal construction of the motor, in particular the number of teeth on the stator and the rotor. The step angle is the minimum degree of rotation associated with a single step. Various motors have different step angles. Table 3.2 shows some step angles for various motors. In Table 3.2, notice the term steps per revolution. This is the total number of steps needed to rotate one complete rotation or 360 degree (e.g., 180 steps x 2 degree =360). It must be noted that perhaps contrary to one's initial impression, a stepper motor does not need more terminal leads for the stator to achieve smaller steps. All the stepper motors discussed in this section have four leads for the stator winding and two COM wires for the centre tap. Although some manufacturers set aside only one lead for the common signal instead of two, they always have four leads for the stators.

Table 3.2 stepper motor step angles

Step angle	Steps per revolution
0.72	500
1.8	200
2	180
2.5	144
5	72
7.5	48
15	24

Figure3.3 shows the interface circuit between PIC microcontroller and stepper motor.

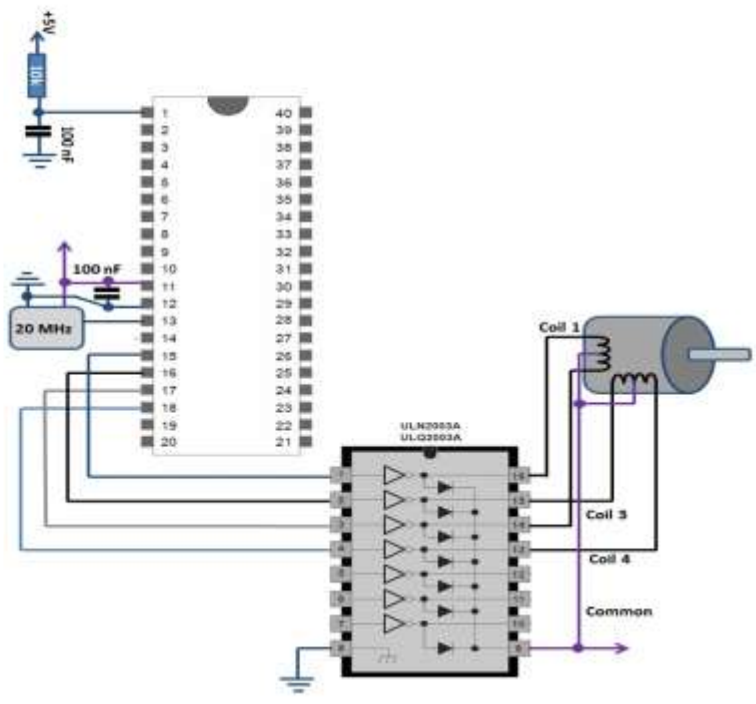

Figure 3.3 PIC connection to stepper motor

Motor speed

The motor speed, measured in steps per second (steps/s), is a function of the switching rate. By changing the length of the time delay loop, we can achieve various rotation speeds.

Holding torque

The following is a definition of holding torque:'' With the motor shaft at standstill or zero rpm condition, the amount of torque, from an external source, required to break away the shaft from its holding position. This is measured with rated voltage and current applied to the motor'' The unit of torque is (kg-cm).

The following table shows the characteristics of some selected stepper motor.

Table3.3 selected stepper motor characteristics [4]

Part No	**Step angle**	**Drive system**	**Volts**	**Phase resistance**	**Current**
151861CP	7.5	Unipolar	5 V	9 ohms	550 mA
171601CP	3.6	Unipolar	7 V	20 ohms	350 mA
164056CP	7.5	Bipolar	5 V	6 ohms	800 mA

Unipolar versus bipolar stepper motor interlace

There are three common types of stepper motor interfacing: universal, unipolar, and bipolar. They can be identified by the number of connections to the motor. A universal stepper motor has four. The universal stepper motor can be configured for all three modes, while the unipolar can be either unipolar or bipolar. Obviously the bipolar cannot be configured for universal nor unipolar mode. Table 3.3 shows selected stepper motor characteristics. Figure 3.4 shows the basic internal connections of all three types of configurations.

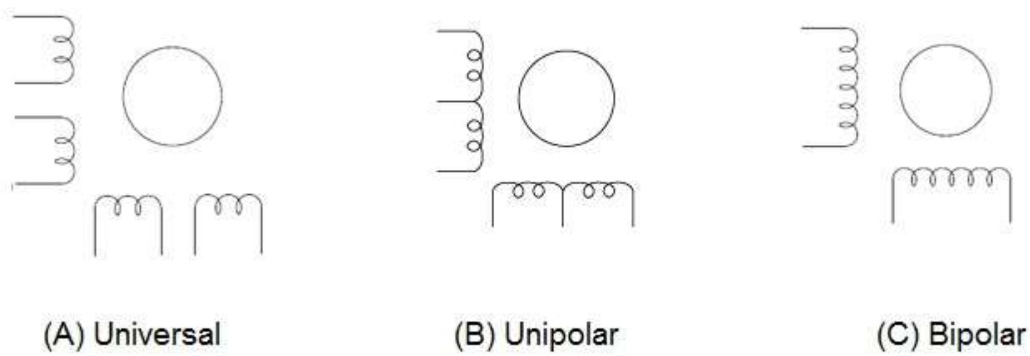

Figure3.4 common stepper motor types

Controlling stepper motor via opt isolator

Optoisolators are widely used to isolate the stepper motor's EMF voltage and keep it from damaging the digital/microcontroller system. This is shown in Figure 3.5.

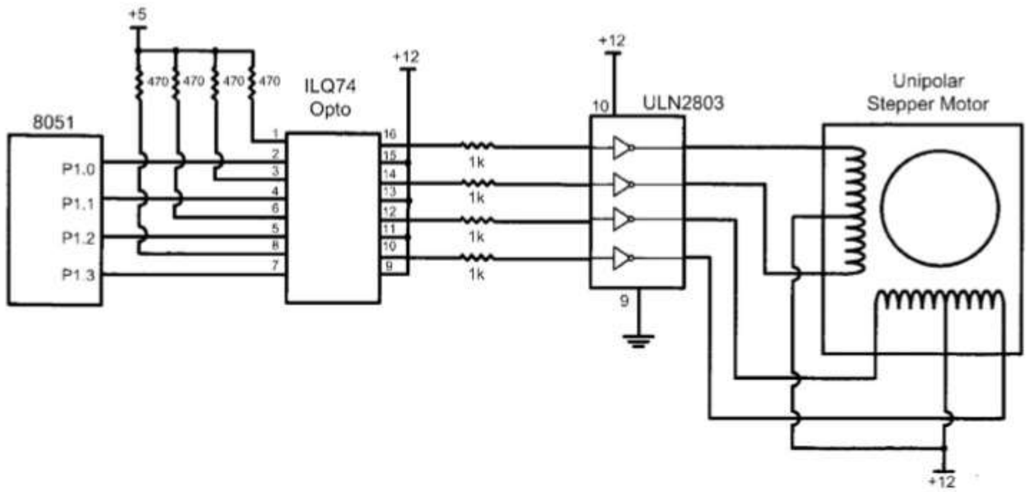

Figure3.5 controlling stepper motor via optoisolator

3-3 DC Motor interfacing

A direct current (DC) motor is another widely used device that translates electrical pulses into mechanical movement. In the DC motor we have only (+) and (-) leads. Connecting them to a DC voltage source moves the motor in one direction. By reversing the polarity, the DC motor will move in the opposite direction. One can easily experiment with the DC motor.

For example, small fans used in many motherboards to cool the CPU are run by DC motors. By connecting their leads to the +and - voltage source, the DC motor moves. While a stepper motor moves in steps of 1 to 15 degrees, the DC motor moves continuously. In a stepper motor, if we know the starting position we can easily count the number of steps the motor has moved and calculate the final position of the motor. This is not possible in a DC motor. The maximum speed of a DC motor is indicated in rpm and is given in the data sheet. The DC motor has two rpms: no-load and loaded. The manufacturer’s data sheet gives the no-load rpm. The no-load rpm can be from a few thousand to tens of thousands. The rpm is reduced when moving a load and it decreases as the load is increased. For example, a drill turning a screw has a much lower rpm speed than when it is in the no-load situation.

DC motors also have voltage and current ratings. The nominal voltage is the voltage for that motor under normal conditions, and can be from 1 to 150 V, depending on the motor. As we increase the voltage, the rpm goes up. The current rating is the current consumption when the nominal voltage is applied with no load, and can be from 25mA to a few amps. As the load increases, the rpm is decreased, unless the current or voltage provided to the motor is increased, which in turn increases the torque [5].

With a fixed voltage, as the load increases, the current (power) consumption of a DC motor is increased. If we overload the motor it will stall, and that can damage the motor due to the heat generated by high current consumption.

Chapter Four

Architectures of Robotic Arm

4-1 Introduction

Robotic arm are commonly used in industries. In many field applications where technical support is required, man-handling is either dangerous or is not possible. In such situations three or more arm manipulators are commonly used. These arms are used to weld, package, paint, position and assemble a host of products that we use daily.

Basically a robot arm is a series of linkages that are connected in such a way that a servo motor can be used to control each joint. The controlling computer, the brain of the robot, is programmed to control the various motors on the robot in a way that allows it to perform specific tasks.

The robot arm can be designed in a number of different ways, the size and shape of this arm is critical to the robotic architecture of the robot. The arm is the part of the robot that positions the final grabber arm or spray head to do their pre-programmed business. If the design of the arm is too large or small this positioning may not be possible. Many arms resemble the human arm, containing shoulders, elbow wrists and hands. The design of the human arm is exceptional and allows for precise and complicated movement.

As a rule we need one motor for each degree of freedom that you want to achieve. A degree of freedom is typically one joint movement. So a simple robot with 3 degree of freedom can move three ways: up and down, left and right, forward and back. Figure 4.1 is a simple pan and title robot has 3 degrees of freedom, powered by servo motors. It can rotate left to right and lift up and down [6].

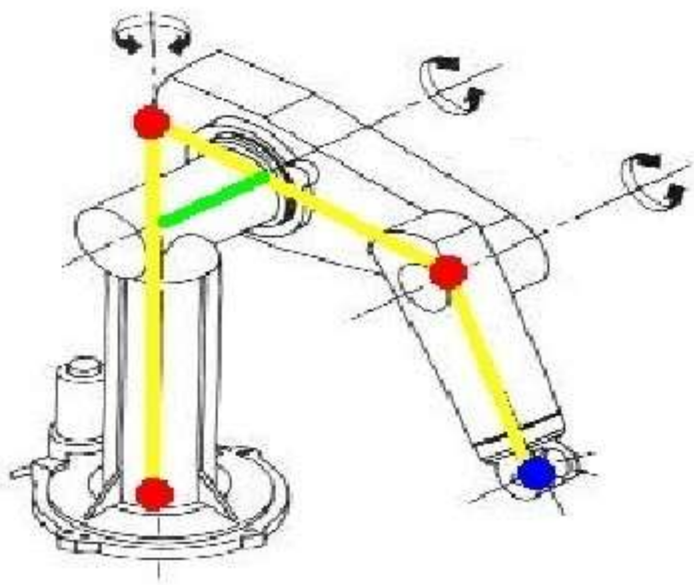

Figure4.1 simple robot arm with three degree of freedom

Many robots today can be designed to move with 7 degree of freedom.

4-2 Degree of Freedom

The following will demonstrate the degree of freedom using the human arm.

First Degree: shoulder Pitch

If you Point your entire arm straight out in front of you. Moving your shoulder up and down. The up and down movement of the shoulder is called the shoulder pitch.

Second Degree: Arm Yaw

If you Point your entire arm straight out in front of you. Move your entire arm from side. This side to side movement is called the arm yaw.

Third Degree: Shoulder roll

If you Point your entire arm straight out in front of you. Now, roll your entire arm from the shoulder, as if were screwing in a light bulb. This rotating movement is called a shoulder roll.

Fourth Degree: Elbow Pitch

If you Point your entire arm straight out in front of you. Hold your arm still, then bend only your elbow. Tour elbow can move up and down. This up and down movement of the shoulder is called the elbow pitch.

Fifth Degree: Wrist Pitch

If you Point your entire arm straight out in front of you. Without moving your shoulder or elbow, flex your wrist up and down. This up and down movement of the wrist is called the wrist pitch.

Sixth Degree: Wrist Yaw

If you Point your entire arm straight out in front of you. Without moving your shoulder or elbow, flex your wrist from side to side. The side to side movement is called the wrist yaw.

Seventh Degree: Wrist Roll

If you Point your entire arm straight out in front of you. Without moving your shoulder or elbow, rotate your wrist, as if you were turning a doorknob. The rotation of the wrist is called the wrist roll.[6]

4-3 Robotic joints

It is obvious that in order to achieve different degree of freedom, different robotic joint are needed. Unlike human joint where we saw3 degree of freedom in the shoulder, the joints robot are normally restricted to 1 degree of freedom, to simplify the mechanics and control of the manipulator. There are two types of commonly found in robots: rotary joints and prismatic joints.

4-4 End Effectors

In the robotic world it is generally understood that the end of the wrist is the end of the robot. The robot has the capability of moving to various positions within the limits of its work envelope. The robot is not yet prepared for the operation that it has to carry out; it does not have the correct ’’Hand’’. The end effector is the correct name for the attachments can be for grasping, lifting, welding, painting and many more [6].

This means that the standard robot can be carry out a vast range of different application depending on the end effectors that is fitted to it. There are two types of end effectors**- grippers** and **tools**. Tools are used where an operations such as welding, painting or drilling need to be performed. Their shapes and types are numerous and varied.

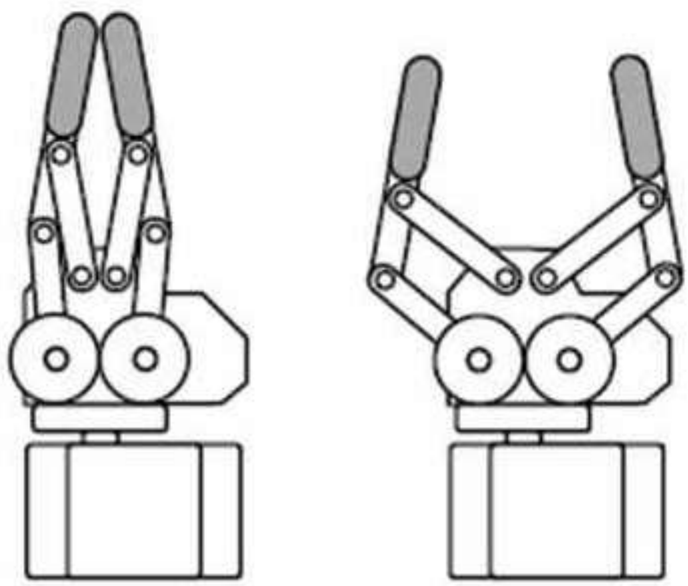

Figure4.2mechanical Gripper

Chapter Five

The proposed system

5-1 Design and programming of Mobile Robotic Arm

Using of mobile phone for robotic control can overcome the limitation of control range when we use cable interface or RF signal. It provides the advantages of robust control; working rang as large as the coverage area of the service provider, no interference with other controllers and up to twelve controls.

Although, the appearance and capabilities of robot very vastly, all robots share the feature of a mechanical, movables structure under some form of control. The control of robot involves three distant phase: perception, processing, action. Generally, the preceptors are sensors mounted on the robot, processing is done by the board microcontroller and the task is performed using motors or with some other actuators.

The important components of this robot are a DTMF decoder, microcontroller and motor drivers. A CM8870 series DTMF decoder is used here. All types of the CM8870 series use digital counting techniques to detect and decode all the 16 DTMF tone pairs into a 4-bit code output. The built-in dial tone rejection circuit eliminates the need of pre-filtering.

Hardware Component of the Project

The project consists of the following components:

- PIC 16F877A [7]
- MT8870 DTMF decoder [8]
- GSM Remote Security Camera

- DC motors
- Stepper motors
- Cell phone
- Head phone
- ULN2003 Current driver [9]
- L293 D Driver [10]
- Robotic ARM Model

Basic parts of a mobile Robotic Arm

- A mechanical device, such as a wheeled platform, arm, or other construction, capable of interacting with its environment.
- Cell phone and DTMF decoder on the device those are able to receive the transmitted signals and decode them to give useful feedback to the device.
- System that process sensory input in the context of the device's current situation and instruct the device to perform actions in response to the situation.
- GSM Remote Security Camera that used to track the movement of robotic arm with high accuracy.

5-2 Circuit Description and simulation Results

The important simulation tools used in this project are:

- Proteus simulation program
- MiKroC compiler for PIC
- PIC simulator

In circuit diagram (Figure 5.1) PIC16F877A microcontroller pin 1 is connected to push button for reset. Pins 11,32 is connected to vcc+5v and pins no: 12,31 is connected to GND. Form DTMF decoder IC m8870 the output pins is connected directly to the PIC16F877A PORTA pins RA0,RA1,RA2,RA3.

In signal are given at PORTA pins RA0,RA1,RA2,RA3, differential input configuration is recognized to be effective, the correct 4-bit decode signal of the DTMF tone is transferred to Microcontroller. The pin11 to pin14 of DTMF decoder are connected to the PORTA pins of microcontroller. Figure below (5.1) illustrates the circuit simulation scheme for the project and how to design it. The project consists of mechanical model of robotic Arm which contain three degree of freedom (3DOF) depending on stepper motors and two DC motor to make it move forward, reverse, turn right and turn left. PIC16F877A microcontroller is the brain of the system which containing appropriate program code with helping of mobile phone that receive the signals comes from anther cell phone which represent the transmitter. The DTMF decoder MT8870 is connected directly to the cell phone attached to the robotic arm to receive the frequency combination and decode them to generate digital outputs.

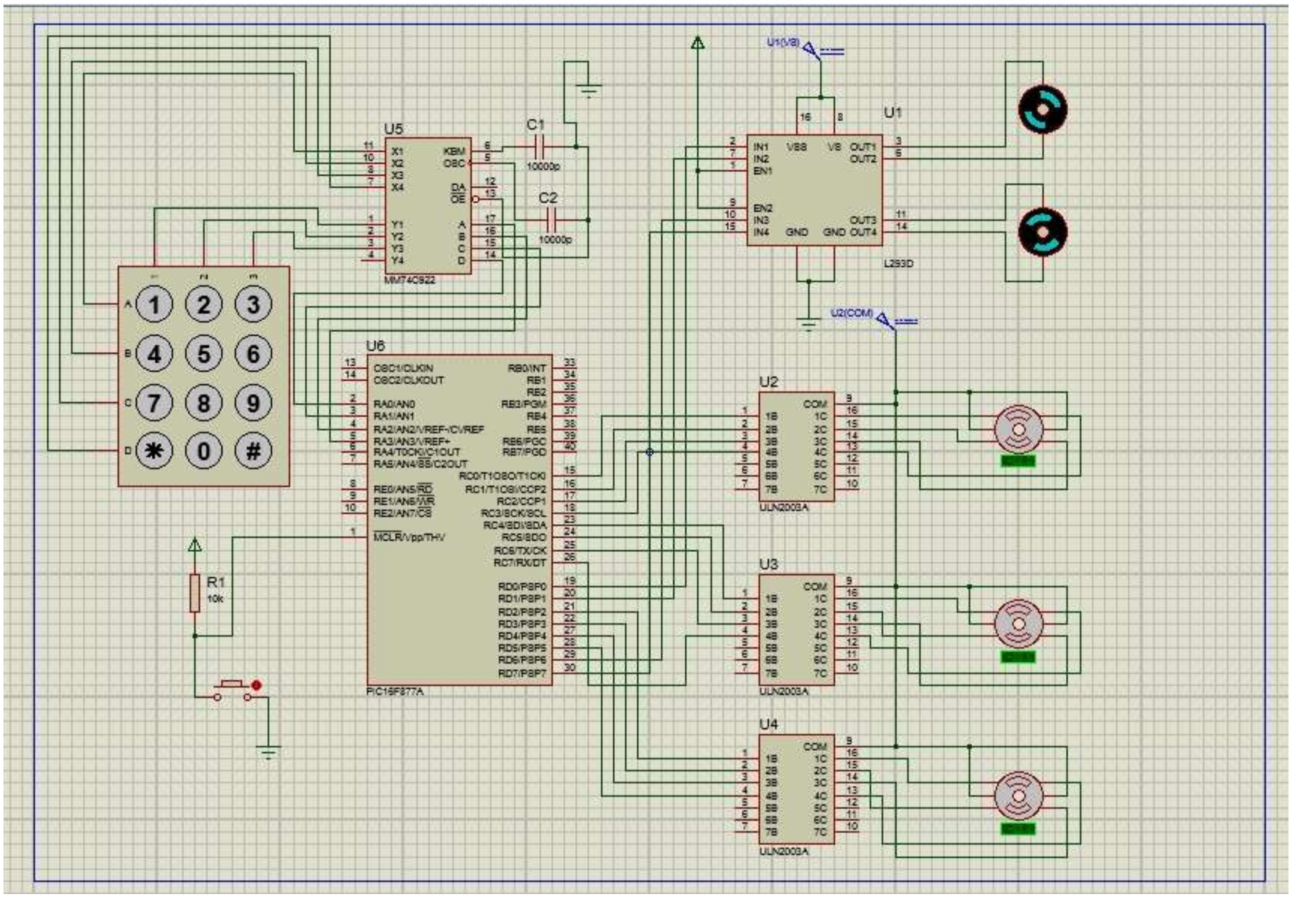

Figure5.1 Simulation circuit of Mobile Robotic Arm Using PIC microcontroller

The flow chart in figure 5.2 explain how the system works

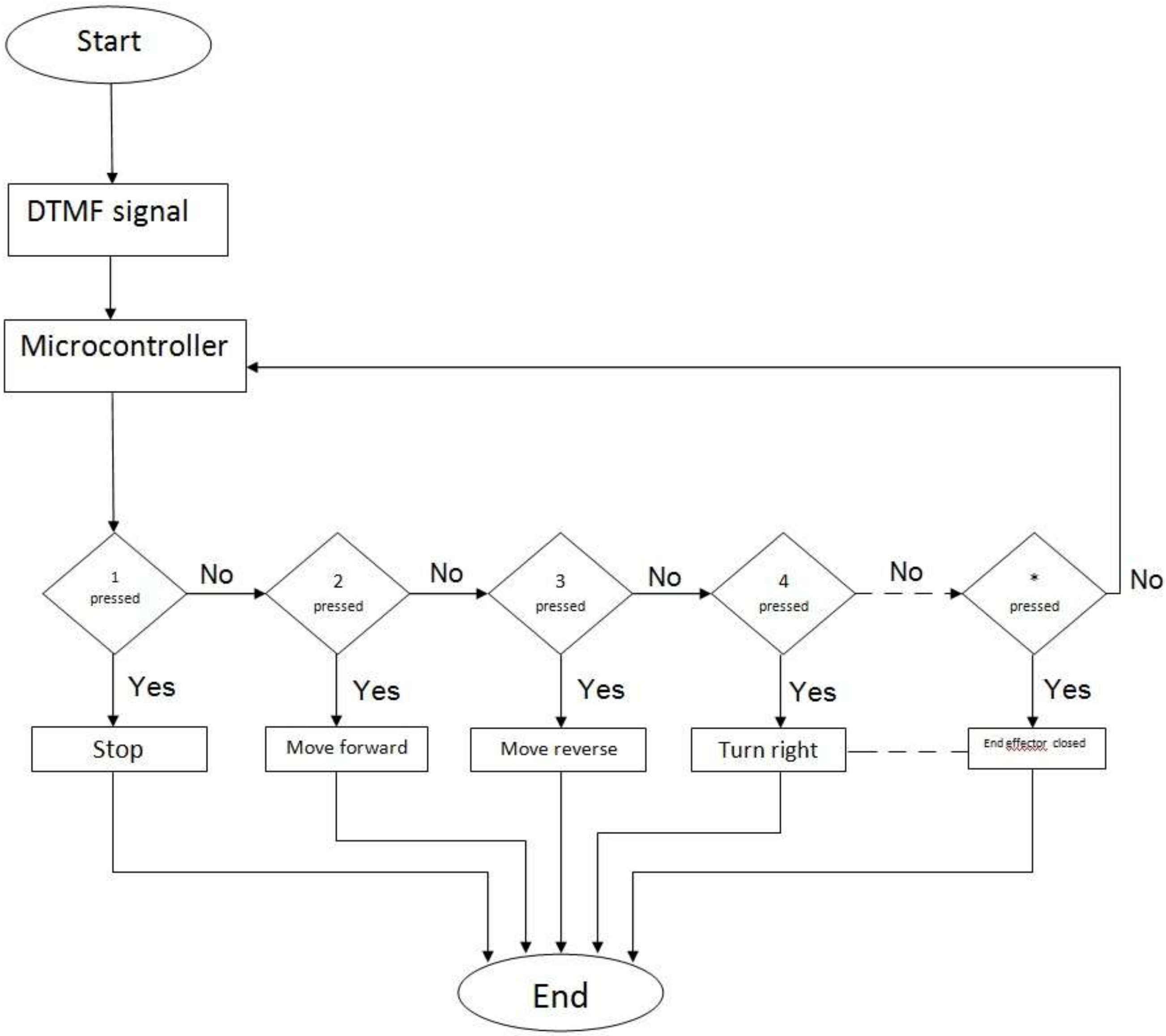

Figure5.2 flowchart of the system

5-3-1 DTMF 8870 Circuit:

It is used for decoding the mobile signals. It gets DTMF tone from the mobile headset's speaker pins (here we used pin 1) at its Pin 2&3 and decodes the DTMF tone into 4-bit digital signal which is fed via Pins 11,12,12,14 to the PORTA of microcontroller.

The DTMF Decoder is operated with a 3.579MHz crystal (Xtal2) applied between pin7 and 8. Table 5.1 shows DTMF data outputs.

Table 5.1: DTMF 8870 Data outputs

Low group (Hz)	High group (Hz)	Digit	OE	D3	D2	D1	D0
697	1209	1	H	L	L	L	H
697	1336	2	H	L	L	H	L
697	1477	3	H	L	L	H	H
770	1209	4	H	L	H	L	L
770	1336	5	H	L	H	L	H
770	1477	6	H	L	H	H	L
852	1209	7	H	L	H	H	H
852	1336	8	H	H	L	L	L
852	1477	9	H	H	L	L	H
941	1336	0	H	H	L	H	L
941	1209	*	H	H	L	H	H
941	1477	#	H	H	H	L	L
697	1633	A	H	H	H	L	H
770	1633	B	H	H	H	H	L
851	1633	C	H	H	H	H	H
941	1633	D	H	L	L	L	L

The M-8870 decoder uses a digital counting technique to determine the frequencies of the limited tones and to verify that they correspond to standard DTMF frequencies. A complex averaging algorithm is used to protect against tone simulation by extraneous signals (such as voice) while tolerating small frequency variations. The algorithm ensures an optimum combination of immunity to talk- off and tolerance to interfering signals (third tones) and noise. When the detector recognizes the simultaneous presence of two valid tones. (Known as signal condition), it raises the Early Steering flag (ESt). Any subsequent loss of signal condition will cause Est.

5-3-2 Results and Discussion

The following block diagram explains method of operation:

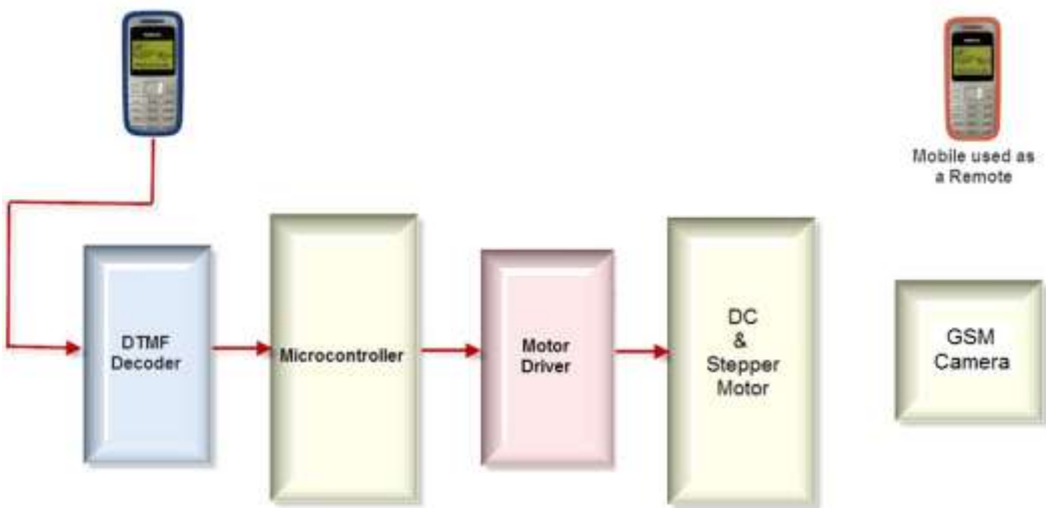

Figure 5.3: Block diagram of the proposed system

In this project the pick and place robot is controlled by a mobile phone that makes a cell to the mobile phone attached to the robot. In the course of a cell, if any button is pressed a tone corresponding to the button pressed is heard at the other end called ‘Dual Tone Multi-Frequency’ (DTMF) tone. The robot receives these tones with help of phone stacked in the robot. The received tone is processed by the microcontroller with helping of DTMF decoder IC MT8870. This IC sends a signals to the PIC microcontroller which interface with motor drivers that drivers the motor in directions forward, reverse, left, right, pick, release, up, down, rotate right. MiKroC compiler for PIC was used to write and compile appropriate programme code. GSM remote security camera that used to track the movement of robotic arm with high accuracy. Table 5.2 shows the simulation results.

Figure below show the GSM camera with motion detection unit that used to track the movement of robotic arm and send image to cell phone.

Figure5.4 GSM camera

Table 5.2 Simulation results of the proposed system

Low group (Hz)	**High group (Hz)**	**Digit**	**D3**	**D2**	**D1**	**D0**	**Robotic arm movement**
697	1209	1	0	0	0	1	Stop (object detect)
697	1336	2	0	0	1	0	Forward
697	1477	3	0	0	1	1	Reverse
770	1209	4	0	1	0	0	Turn right
770	1336	5	0	1	0	1	Turn left
770	1477	6	0	1	1	0	Shoulder (m3) clockwise
852	1209	7	0	1	1	1	Shoulder (m3) anticlockwise
852	1336	8	1	0	0	0	Elbow down
852	1477	9	1	0	0	1	Elbow up
941	1336	0	1	0	1	0	End effector open
941	1209	*	1	0	1	1	End effector closed

Chapter Six

Conclusion and Recommendation

6-1 Conclusion

The objective of designing this robot is simply to help the humans in the future for security purposes. It can move forward, reverse, turn right and left for a specific distance according to the controller program and specifications .The development of this robot is based on PIC microcontroller platform which interfaced with mobile phone module to control the mobile robotic arm movement. This prototype of the robot is expected to overcome the problem such as placing or picking object the far away from the user, pick and place hazardous objects in the fastest and easiest way.

The methodology used in this project is Dual Tone Multi-frequency (DTMF). The robot is controlled by a cell phone, through this we can make our robot communicate on a large scale over a large distance even from different cities. The movement of this robot is displayed on a LCD display. This robot can also be used to reach the places where humans cannot reach such as small tunnels, etc. This is how we can use this robot in different fields as well as for research purpose by further manipulation in programming it can be modified accordingly.

6-2 Recommendation

It’s recommended that:

1. Future robots can be able to move under their own power and navigation systems and image features capturing can be added to the system by using suitable type of cameras.
2. Robot gripper design can be more sophisticated, and universal hands capable of multiple tasks will be available.
3. Use biopotential signal as the source of the signal to control the robotic arm.

References

[1] Puran Singh, Anil kumar, Mahesh Vashisth, ’’Design of a Robotic Arm with Gripper& End Effector for Spot Welding ’’, Amity School of Engineering and Technology Delhi, India, Horizon Research Publishing ,2013.

[2] N.A.PS.K .Narangoda, D.A.N. Lakmal, T.M.Weerakkody, P.Samsoodeen ’’implementation of mobile robotic arm’’ ,india ,2013.

[3] Ashutosh pattnaik, Rajiv ranjan, ’’ robotic arm control through human arm movement using accelerometers’’ ,national institute of technology , Rourkela, 109EI0297 ,2013.

[4] Mohammed Ali Mazidi , Rolin.D Mckinlay danny causey,’’PIC microcontroller and embedded systems using assembly and C for PIC18, courier Kendallivlle ,Inc, coral graphic services , new jersey , USA , 2008.

[5] Bert Van Dam, ‘’PIC Microcontroller’’, printed in Netherlands by wilco, Amersfoort,2007.

[6] T4 Galway Education Centre,’’ applied control technology-introduction to robotics’’,2007.

[7] http://akizukidenshi.com/download/PIC16F87XA.pdf

[8]http://www.microsemi.com/document-portal/doc_view/127041-mt8870d-datasheet-oct2006

[9]http://toshiba.semiconstorage.com/info/docget.jsp?pid=ULN2004APG&lang=en&type=datasheet

[10] http://users.ece.utexas.edu/~valvano/Datasheets/L293d.pdf

Appendix

Microcontroller code

```
void main()

{

 trisb=0b1100000;

 trisc=0x00;

 trisd=0x00;

 ADCON1=0b1000111;

 trisa=0b00111111;

 trise=0b00000111;

 portb=0;

 portc=0;

 loop:

 if(porta.b0==0&&porta.b1==0&&porta.b2==0&&porta.b3==0)

 {portd.b0=0;portd.b1=0;portd.b6=0;portd.b7=0;}

 if(porta.b0==0&&porta.b1==0&&porta.b2==0&&porta.b3==1)

 {portd.b0=0;portd.b1=0;portd.b6=0;portd.b7=0;}
```

```
if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==0)

{portd.b0=1;portd.b1=0;portd.b6=1;portd.b7=0;}

if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==1)

{portd.b0=0;portd.b1=1;portd.b6=0;portd.b7=1;}

if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==0)

{portd.b0=0;portd.b1=1;portd.b6=1;portd.b7=0;}

if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==1)

{portd.b0=1;portd.b1=0;portd.b6=0;portd.b7=1;}

if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)

{goto stepper;}

if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)

{goto anticlock;}

if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==0)

{goto elbowdown;}

if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==1)

{goto elbowup;}

if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==0)

{goto openendeffector;}
```

```
 if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==1)

 {goto closeendeffector;}

stepper:

 if(porta.b0==0&&porta.b1==0&&porta.b2==0&&porta.b3==1)

 {goto loop}

 if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==0)

 {goto loop;}

if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==1)

 {goto loop;}

if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==0)

 {goto loop;}

if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==1)

 {goto loop;}

   if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)

 {goto anticlock;}

if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==0)

 {goto elbowdown;}
```

```
if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==1)
{goto elbowup;}
if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==1)
{goto closeendeffector;}
if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==0)
{goto openendeffector;}
 if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
{portd.b2=0;portd.b3=0;portd.b4=0;portd.b5=1;//45
delay_ms(1000);
 a:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
{portd.b2=0;portd.b3=0;portd.b4=1;portd.b5=1;//90
delay_ms(1000);
 b:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
{portd.b2=0;portd.b3=0;portd.b4=1;portd.b5=0;//135
delay_ms(1000);
```

```
c:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
{portd.b2=0;portd.b3=1;portd.b4=1;portd.b5=0;//180
delay_ms(1000);
d:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
{portd.b2=0;portd.b3=1;portd.b4=0;portd.b5=0;//225
delay_ms(1000);
e:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
{portd.b2=1;portd.b3=1;portd.b4=0;portd.b5=0;//270
delay_ms(1000);
f:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
{portd.b2=1;portd.b3=0;portd.b4=0;portd.b5=0;//315
delay_ms(1000);
g:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
```

```
{portd.b2=1;portd.b3=0;portd.b4=0;portd.b5=1;//0-180
delay_ms(1000);
 goto g;
goto f;
goto e;
goto d;
goto c;
goto b;
goto a;}}}}}}}}
 goto stepper;
 anticlock:
 if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
{goto stepper;}
if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==0)
{goto elbowdown;}
if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==1)
{goto elbowup;}
if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==1)
```

```
 {goto closeendeffector;}

if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==0)

 {goto openendeffector;}

 if(porta.b0==0&&porta.b1==0&&porta.b2==0&&porta.b3==1)

 {goto loop}

if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==0)

 {goto loop;}

if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==1)

 {goto loop;}

if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==0)

 {goto loop;}

if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==1)

 {goto loop;}

  if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)

 {portd.b2=1;portd.b3=0;portd.b4=0;portd.b5=0;//-45

 delay_ms(1000);

h:

 if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)
```

```
{portd.b2=1;portd.b3=1;portd.b4=0;portd.b5=0;//-90
delay_ms(1000);
i:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)
{portd.b2=0;portd.b3=1;portd.b4=0;portd.b5=0;//-135
delay_ms(1000);
j:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)
{portd.b2=0;portd.b3=1;portd.b4=1;portd.b5=0;//-180
delay_ms(1000);
k:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)
{portd.b2=0;portd.b3=0;portd.b4=1;portd.b5=0;//-225
delay_ms(1000);
l:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)
{portd.b2=0;portd.b3=0;portd.b4=1;portd.b5=1;//-270
delay_ms(1000);
```

```
m:
 if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)
 {portd.b2=0;portd.b3=0;portd.b4=0;portd.b5=1;//-315
 delay_ms(1000);
n:
 if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)
 {portd.b2=1;portd.b3=0;portd.b4=0;portd.b5=1;//0-180
 delay_ms(1000);
goto n;
 goto m;
 goto l;
 goto k;
 goto j;
 goto i;
 goto h;}}}}}}}}
  goto anticlock;
  elbowdown:
```

```
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)

{goto stepper;}

if(porta.b0==0&&porta.b1==0&&porta.b2==0&&porta.b3==1)

{goto loop}

if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==0)

{goto loop;}

if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==1)

{goto loop;}

if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==0)

{goto loop;}

if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==1)

{goto loop;}

if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)

{goto anticlock;}

if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==1)

{goto elbowup;}

if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==1)

{goto closeendeffector;}
```

```
if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==0)
 {goto openendeffector;}
if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==0)
 {portc.b4=0;portc.b5=0;portc.b6=0;portc.b7=1;//45
 delay_ms(1000);
  o:
 if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==0)
 {portc.b4=0;portc.b5=0;portc.b6=1;portc.b7=1;//90
 delay_ms(1000);
  p:
 if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==0)
 {portc.b4=0;portc.b5=0;portc.b6=1;portc.b7=0;//135
 delay_ms(1000);
  goto p;
 goto o;}}}
  goto elbowdown;
  elbowup:
  if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
```

```
 {goto stepper;}
if(porta.b0==0&&porta.b1==0&&porta.b2==0&&porta.b3==1)
 {goto loop}
if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==0)
 {goto loop;}
if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==1)
 {goto loop;}
if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==0)
 {goto loop;}
if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==1)
 {goto loop;}
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)
 {goto anticlock;}
if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==0)
 {goto elbowdown;}
if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==1)
 {goto closeendeffector;}
if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==0)
```

```
 {goto openendeffector;}
if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==1)
 {portc.b4=0;portc.b5=0;portc.b6=1;portc.b7=0;//-45
 delay_ms(1000);
q:
 if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==1)
 {portc.b4=0;portc.b5=0;portc.b6=1;portc.b7=1;//-90
 delay_ms(1000);
r:
 if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==1)
 {portc.b4=0;portc.b5=0;portc.b6=0;portc.b7=1;//-135
 delay_ms(1000);
  s:
 if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==1)
 {portc.b4=1;portc.b5=0;portc.b6=0;portc.b7=1;//-180
 delay_ms(1000);
goto s;
 goto r;
```

```
goto q}}}}
 goto elbowup;
openendeffector:
 if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
{goto stepper;}
if(porta.b0==0&&porta.b1==0&&porta.b2==0&&porta.b3==1)
{goto loop}
if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==0)
{goto loop;}
if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==1)
{goto loop;}
if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==0)
{goto loop;}
if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==1)
{goto loop;}
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)
{goto anticlock;}
if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==0)
```

```
 {goto elbowdown;}
if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==1)
 {goto elbowup;}
if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==1)
 {goto closeendeffector;}
  if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==0)
 {portc.b0=0;portc.b1=0;portc.b2=0;portc.b3=1;//-45
 delay_ms(1000);
  dd:
 if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==0)
 {portc.b0=0;portc.b1=0;portc.b2=1;portc.b3=1;//-90
 delay_ms(1000);
  ee:
 if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==0)
 {portc.b0=0;portc.b1=0;portc.b2=1;portc.b3=0;//-135
 delay_ms(1000);
  ff:
 if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==0)
```

```
{portc.b0=0;portc.b1=1;portc.b2=1;portc.b3=0;//-180
delay_ms(1000);
 goto ff;
goto ee;
goto dd;}}}}
 goto openendeffector;
 closeendeffector:
if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==0)
{goto stepper;}
if(porta.b0==0&&porta.b1==0&&porta.b2==0&&porta.b3==1)
{goto loop}
if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==0)
{goto loop;}
if(porta.b0==0&&porta.b1==0&&porta.b2==1&&porta.b3==1)
{goto loop;}
if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==0)
{goto loop;}
if(porta.b0==0&&porta.b1==1&&porta.b2==0&&porta.b3==1)
```

```
 {goto loop;}

if(porta.b0==0&&porta.b1==1&&porta.b2==1&&porta.b3==1)

 {goto anticlock;}

if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==0)

 {goto elbowdown;}

if(porta.b0==1&&porta.b1==0&&porta.b2==0&&porta.b3==1)

 {goto elbowup;}

 if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==0)

 {goto openendeffector;}

if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==1)

 {portc.b0=0;portc.b1=0;portc.b2=1;portc.b3=0;//45

 delay_ms(1000);

aa:

 if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==1)

 {portc.b0=0;portc.b1=0;portc.b2=1;portc.b3=1;//90

 delay_ms(1000);

bb:

 if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==1)
```

```
 {portc.b0=0;portc.b1=0;portc.b2=0;portc.b3=1;//135
 delay_ms(1000);
cc:
 if(porta.b0==1&&porta.b1==0&&porta.b2==1&&porta.b3==1)
 {portc.b0=1;portc.b1=0;portc.b2=0;portc.b3=1;//180
 delay_ms(1000);
goto cc;
 goto bb;
 goto aa;}}}}
goto closeendeffector;
   }
```