Chapter 1

Introduction

1.1 Background

Taking a look back at the history of robot development, a special kind of human-size industrial robotic arm called Programmable Universal Machine for Assembly (PUMA) came into existence. This type of robot is often termed anthropomorphic because of the similarities between its structure and the human arm. The individual joints are named after their human-arm counterparts. In our work the hand is a generalized manipulator. In the proper sense of the word, manipulation is the function of the arm. The function of the arm is to position and orient the hand, act as a mechanical connection and power and sensing transmission link between the hand and the main body of the person. The full functional meaning of the arm rests in the hand (Bejczy & Jau, 1986). Our work provides important elements that are required to build a simple robotic arm of very high quality. As stated earlier we are making use of the 8051-based microcontroller. The 8051's instruction set is optimized for one-bit operations that are often desired in real world, real time operations.

1.2 Objective

The primary objective is to make the Robotic arm, which comprises of three stepper motors, to interface with the Intel 8051-based microcontroller(AT89S52). It provides more interfaces to the outside world and has larger memory to store many programs.

1.2.1 Recycling

Recycling involves processing used materials into new products to prevent waste of potentially useful materials, reduce the consumption of fresh raw materials, reduce energy usage, reduce air pollution (from incineration) and water pollution (from land filling) by reducing the need for "conventional" waste disposal, and lower greenhouse gas emissions as compared to virgin production. Recycling is a key component of modern waste management and is the third component of the "Reduce, Reuse, Recycle" waste hierarchy.

Recyclable materials include many kinds of glass, paper, metal, plastic, textiles, and electronics. Although similar in effect, the composting or other reuse of biodegradable waste – such as food or garden waste – is not typically considered recycling. Materials to be recycled are either brought to a collection center or picked up from the curbside, then sorted, cleaned, and reprocessed into new materials bound for manufacturing.

There is another major impact of air pollution, said Dr Shahnaz Haque Hussain, dean of the earth and environment department of Dhaka University."Increasing emissions and dust are harming the natural growth of trees and plants and destroying the ecological balance," she said.

The another perception of our project is to encourage scientist and others to consider saving the environment and create social awareness to recycle our day to day waste.

1.2 Scope

The scope of this work involves confirming the AT89S52 micro-controller Input/Output (I/O) signals are compatible with that of the robotic arm stepper motors and testing of the robot's motor through programming the 8051 microcontroller. C programming is used to develop the programs on the 8051 micro-controller platform that takes robot's motor signal as I/O and controls the robot operation programmatically. We have assumed that after figuring out the interface issues for the Robot with the AT89S52 microcontroller, the same knowledge can be extended to make very complex robots with enhanced functionality.

1.3 Approach

We were able to perform a detailed study of the robotic arm and the AT89S52 microcontroller. We tested the built robotic arm, and the stepper motors when the robot is loaded. We also learnt and familiarized with the AT89S52 micro-controller using C language, and converting the C language codes to hexadecimal codes using a development board.

1.5 Robotics

The word robotics, meaning the study of robots was coined by Isaac Asimov. Robotics involves elements of both mechanical and electrical engineering, as well as control theory, computing and now artificial intelligence (Selig, 1992). According to the Robot Institute of America, "A robot is a reprogrammable, multifunctional manipulator designed to move materials, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks" (Robotics Research Group, n.d.) .

The fact that a robot can be reprogrammed is important: it is definitely a characteristic of robots. In order to perform any useful task the robot must interface with the environment, which may comprise feeding devices, other robots, and most importantly people.

Chapter 2

Construction of the Robotic Arm

2.1 Methodology

The method employed in designing and constructing the robotic arm are based on the operational characteristics and features of the microcontrollers, stepper motors, the electronic circuit diagram and most importantly the programming of the microcontroller and stepper motors.

2.2 Block Diagram

The block diagram of our work is as shown in Figure 3.

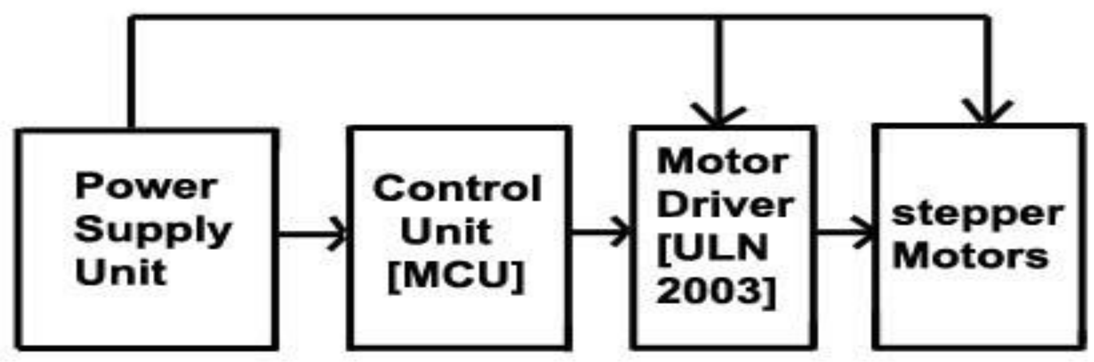
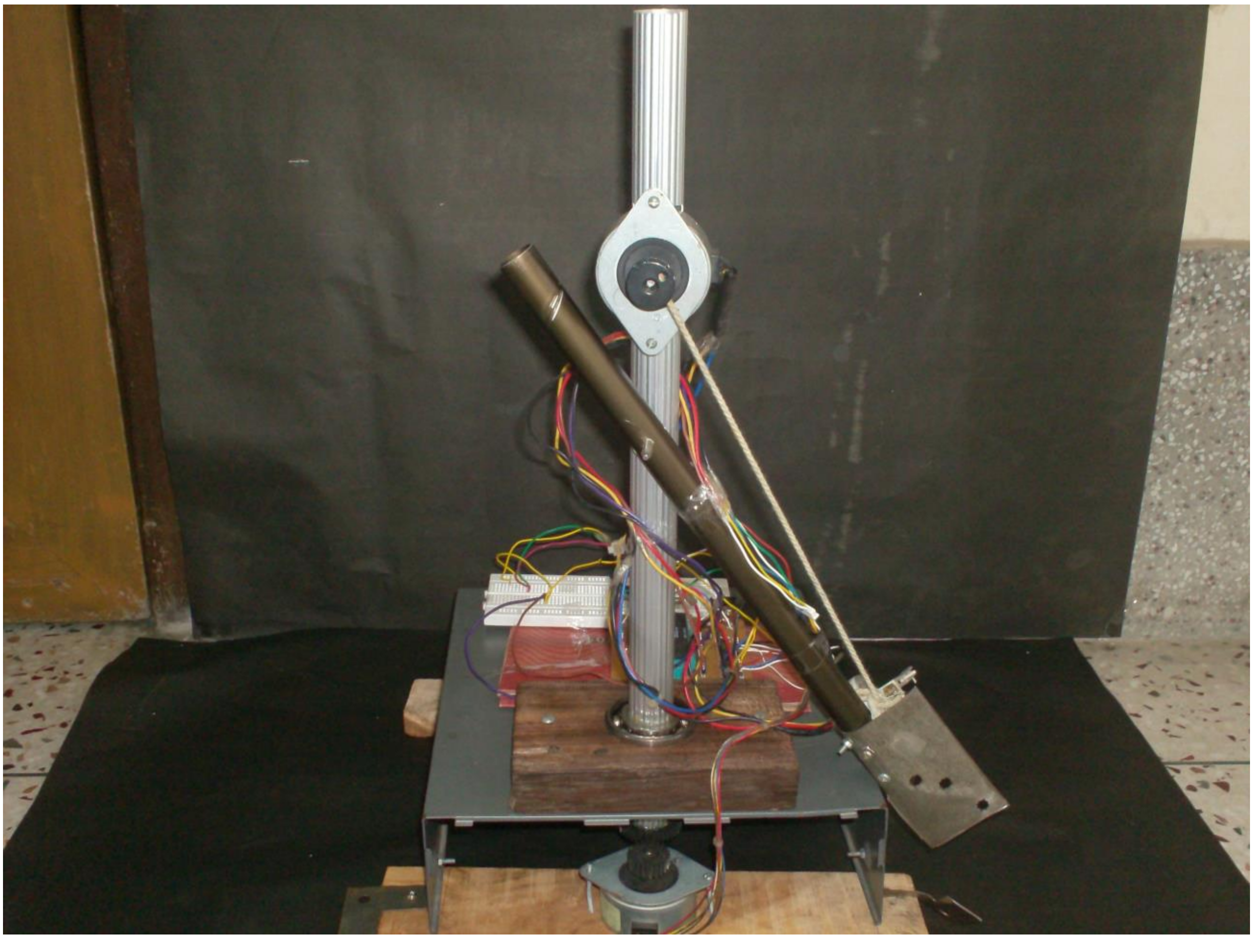


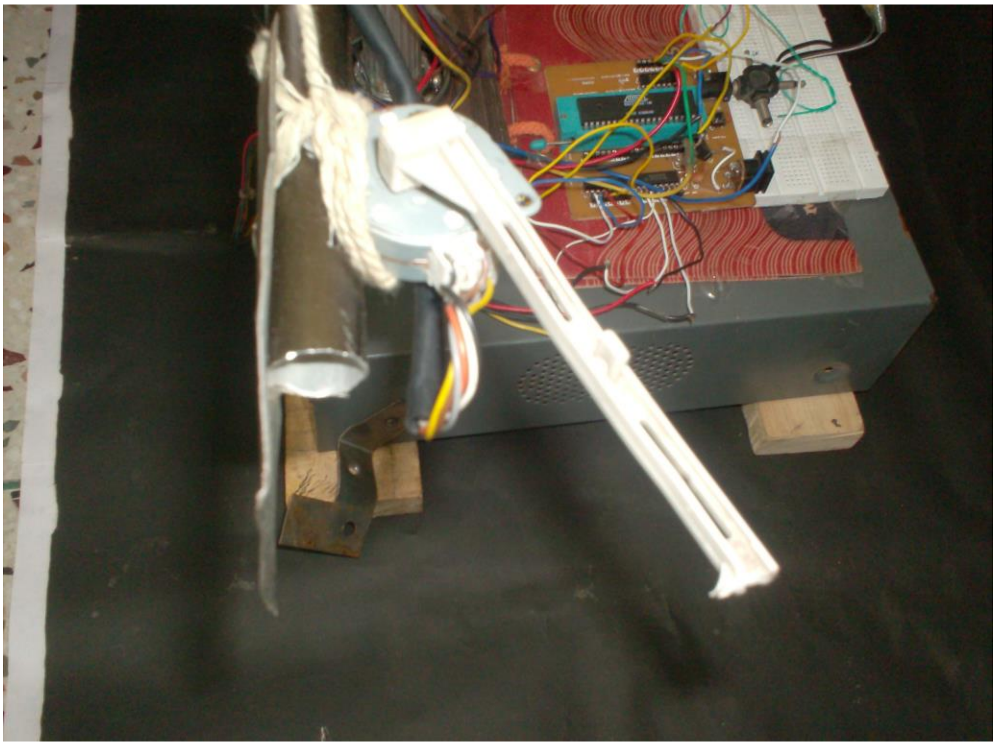
Figure - 2.1 **:** Block diagram of the project

2.3 Mechanical Structure of the Arm

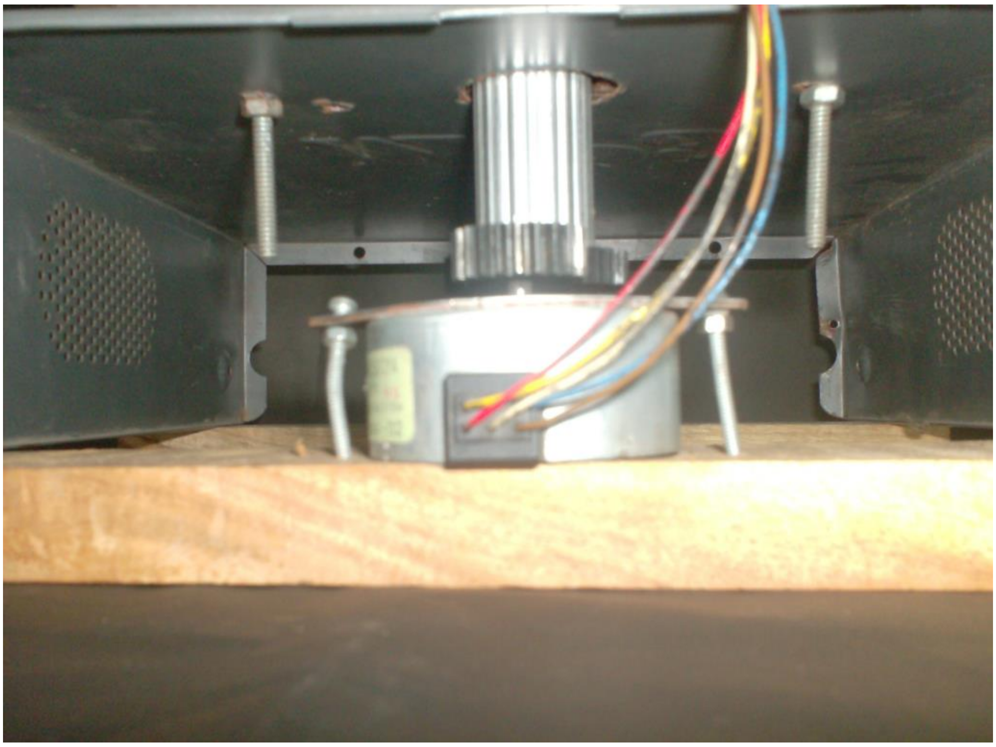
In constructing our arm, we made use of three stepper motors since our structure is a three dimensional structure. The pictures of the robotic arm are shown below. There is a stepper motor (M1) at the base, which allows for circular movement of the whole structure; another at the shoulder which allows for upward and downward movement of the arm; while the last stepper motor at the wrist allows for the picking of objects by the fingers.



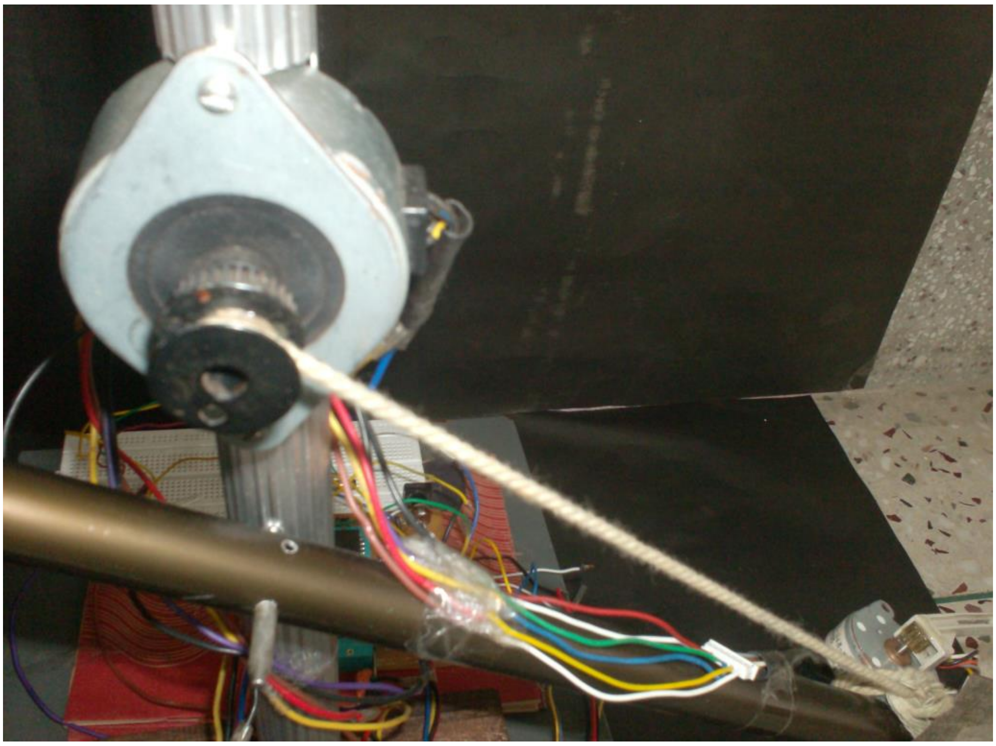
Picture -1 : The total view of the robotic arm



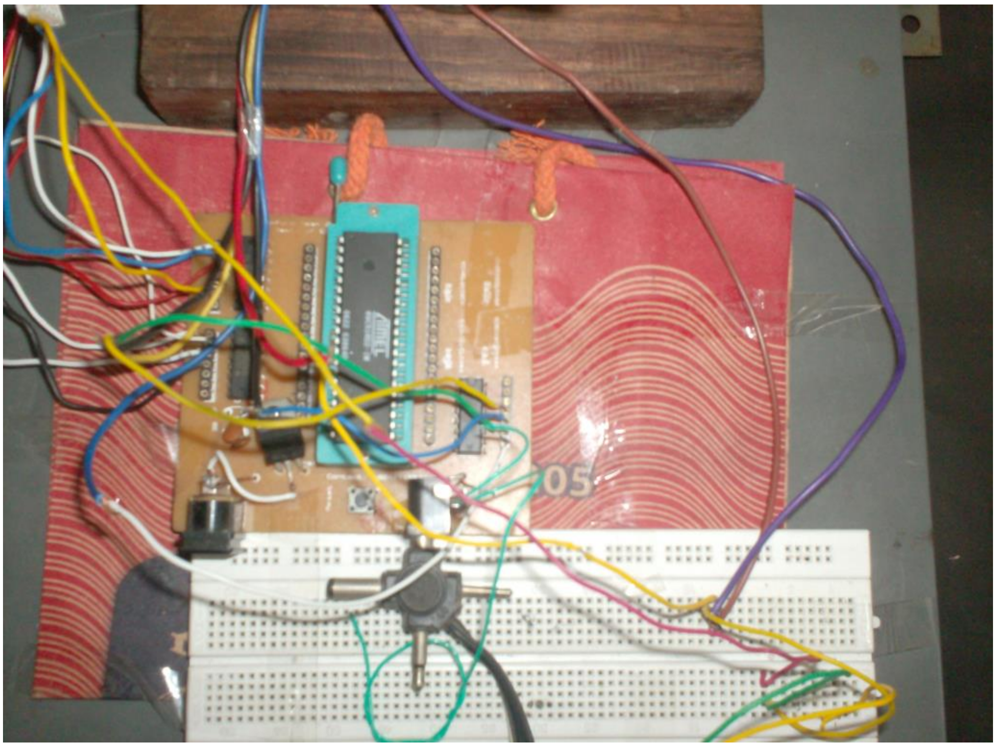
Picture – 2 : The view of the grip of the robotic arm



Picture – 3 : The view of inside the base.



Picture – 4 : The view of the joint of the robotic arm



Picture - 5 : The view of the control circuit of the robotic arm

We used two 1.5′ long metallic pipes (one is of 1″ diameter and the other is of 10mm diameter). Both of the pipes are come from the trash. To make the base of the arm we used a metallic box which is taken from a broken MP3 player and a small piece of wood to make stable the base. A small bearing is inserted into the wood which helps the vertical pipe to move smoothly. A gear is attached at the bottom of the vertical pipe. The vertical pipe is synchronized with the base motor with the help of a gear. When the base motor moves the whole arm rotates with the help of the gear. Two other pieces of woods are used to increase the height of the base. At the joint another stepper motor (M2) is inserted half way of the vertical pipe. A ribbon of sewing machine is glued with the rotor of M2. One end of a thin rope is glued with the ribbon and the other end is tightened with the grip of the arm. When the joint motor (M2) rotates, the rope is twisted in or get released from the ribbon respectively. This helps to lift up or down the upper part of the arm. The metallic pipe which is used as the upper part of the arm can move almost freely (with a little friction) on a thin and small steel rod. The grip is constructed with a small piece of tin sheet and a plastic stick from a broken CD ROM player. The electrical connecting wires used in the arm are taken from a rejected computer power supply.

2.4 Circuit Diagram

The electronic circuit diagram of the development board is as shown in Figure - 2.2. The PCB layouts(top view and bottom view) of the circuit are also shown in fig 2.3 and fig2. 4 respectively. The connection of the identified components and devices are as shown. The components shown are: the *MCU*, *capacitors,ULN2003s* and *Crystal.* This components work together to achieve the set goal of controlling the anthropomorphic-like arrangement of the stepper motor. The microcontroller is the processing device that coordinates all the activities of all the components for proper functioning.

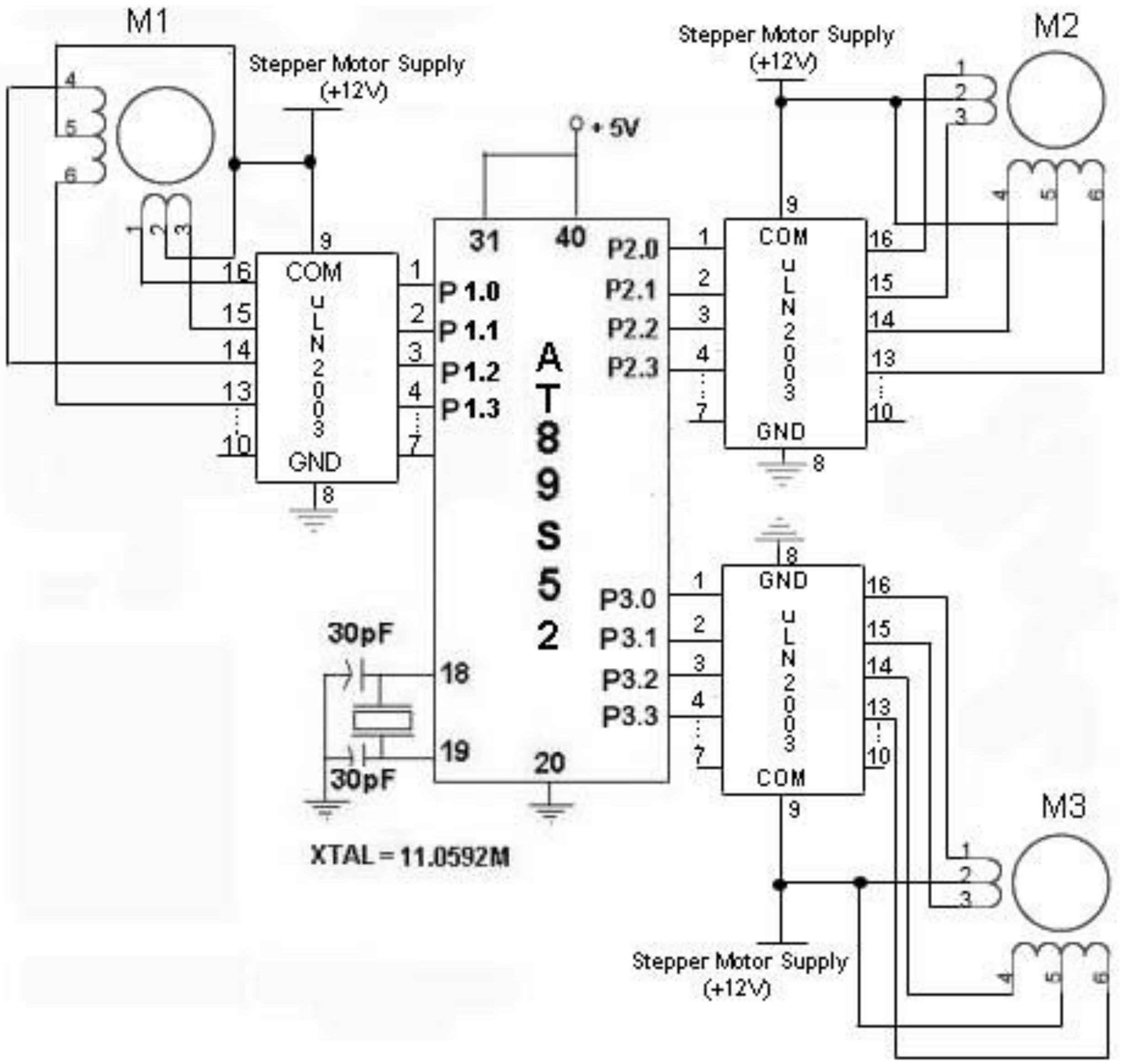


Figure – 2.2 : Circuit diagram of the Control Circuit

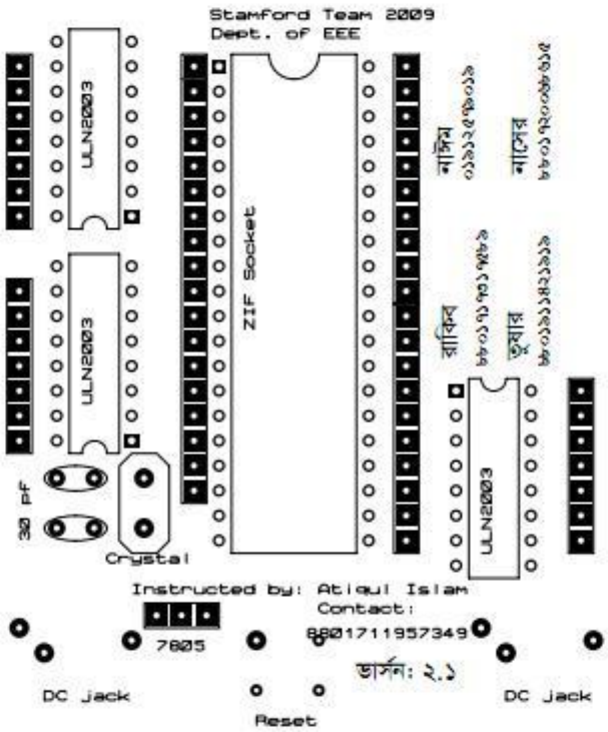


Figure – 2.3 : Top view of the PCB layout

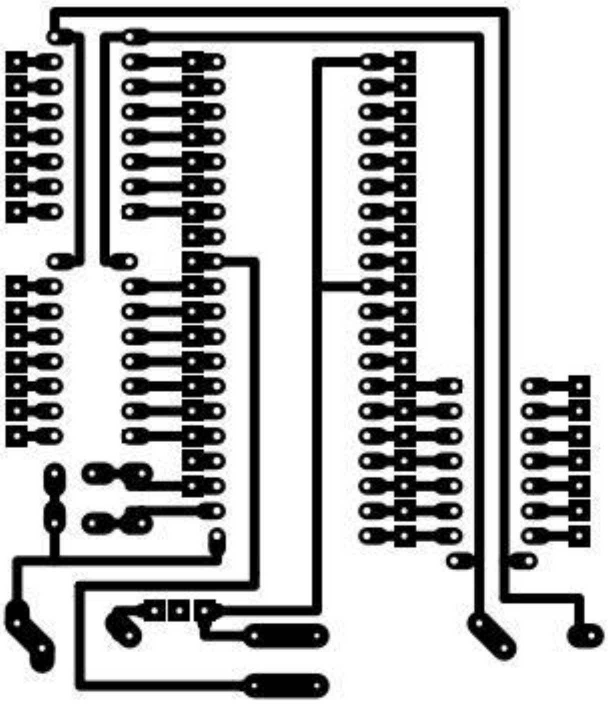


Figure – 2.4 : Bottom View of the PCB layout

2.5 Power Supply

Since our main concentration was on the control of the stepper motors by the microcontroller, we didn't pay much attention to the power supply section. We used a ready made AC power adaptor (output DC 12V, 500mA), the output of which is fed into a voltage regulator LM7805 that has an output of +5V which is required to power the control unit.

To supply +12V to the stepper motors and the ULNs we used a computer power supply which was separated from the main unit of the computer. Go to **Appendix A** to see the specification of the power supplies used in this project.

2.6 Motor Driving Circuit

This is the circuit which drives the coil of stepper motor. Darlington connection-type transistor is used for the drive of the coil. As for the Darlington connection, 2 stages of transistors are connected inside in series. The diode to be putting between the collector and the power is for the protection of the transistor. When the transistor becomes OFF from ON, the coil of the motor tries to continue to pass an electric current and generates high voltage. An electric current by this voltage is applied to the diode and the high voltage which applies over the transistor is prevented.

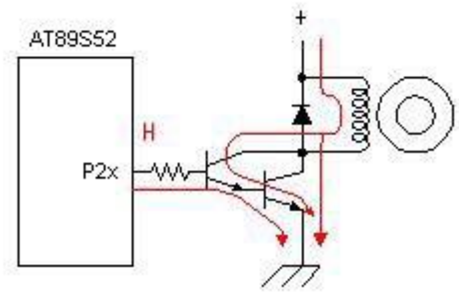


Figure - 2.5 : When the control signal is high the supply current flows through the darlington transistor via the motor coil.

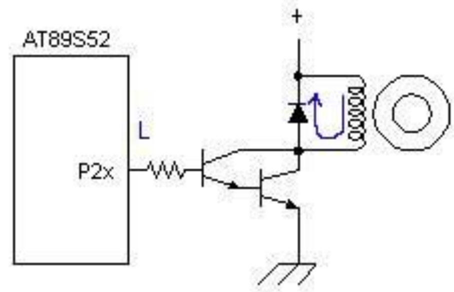


Figure – 2.6 : When the control signal gets low the current stored in the motor coil flows through the free-wheeling diode.

In our project we have used ULN2003AN as the motor driving circuit. It has seven darlington arrays. It has the following freatures :

- Seven darlingtons per package
- Output current 500mA per driver (600mA peak)
- Output voltage 50V
- Integrated suppression diodes for inductive loads
- Outputs can be paralleled for higher current
- TTL/CMOS/PMOS/DTL compatible inputs
- Inputs pinned opposite outputs to simplify layout

Figure – 2.7 : Packaging system of UNL2003AN

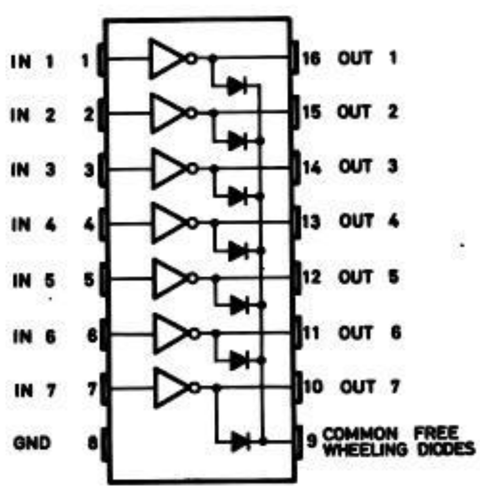


Figure - 2.8 : Pin connection of ULN2003AN

Chapter 3

Stepper Motor

3.1 Definition of Stepper Motor

The stepping motor is a motor that is driven and controlled by an electrical pulse train generated by the MCU (or other digital device). Each pulse drives the stepping motor by a fraction of one revolution, called the *step angle*. Go to **Appendix A** to see the specifications of the three stepper motors used in this project.

3.2 Working Principle of Stepper Motor

Motors convert electrical energy into mechanical energy. A stepper motor converts electrical pulses into specific rotational movements. The movement created by each pulse is precise and repeatable, which is why stepper motors are so effective for positioning applications.

Permanent Magnet stepper motors incorporate a permanent magnet rotor, coil windings and magnetically conductive stators. Energizing a coil winding creates an electromagnetic field with a north and south pole as shown in figure 3.1. The stator carries the magnetic field which causes the rotor to align itself with the magnetic field. The magnetic field can be altered by sequentially energizing or "stepping" the stator coils which generates rotary motion.

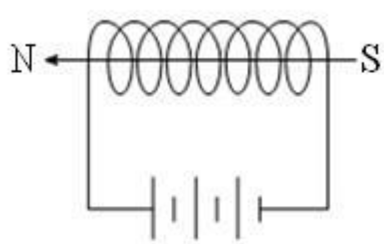


Figure – 3.1: Magnetic field created by energizing a coil winding.

3.2.1 Full Stepping

Figure 3.2 illustrates a typical step sequence for a two phase motor. When the initial is given, the stator is energized. This magnetically locks the rotor, since unlike poles attract. In step 1 when phase A is turned on and other phases are turned off the rotor moves 90° clockwise as shown in the figure 3.2. In step 2 when phase A is turned off and phase B is turned on, the rotor rotates 90° clockwise. In Step 3, phase B is turned off and phase A is turned on but with the polarity reversed from Step 1. This causes another 90° rotation. In Step 4, phase A is turned off and phase B is turned on, with polarity reversed from Step 2. Repeating this sequence causes the rotor to rotate clockwise in 90° steps.

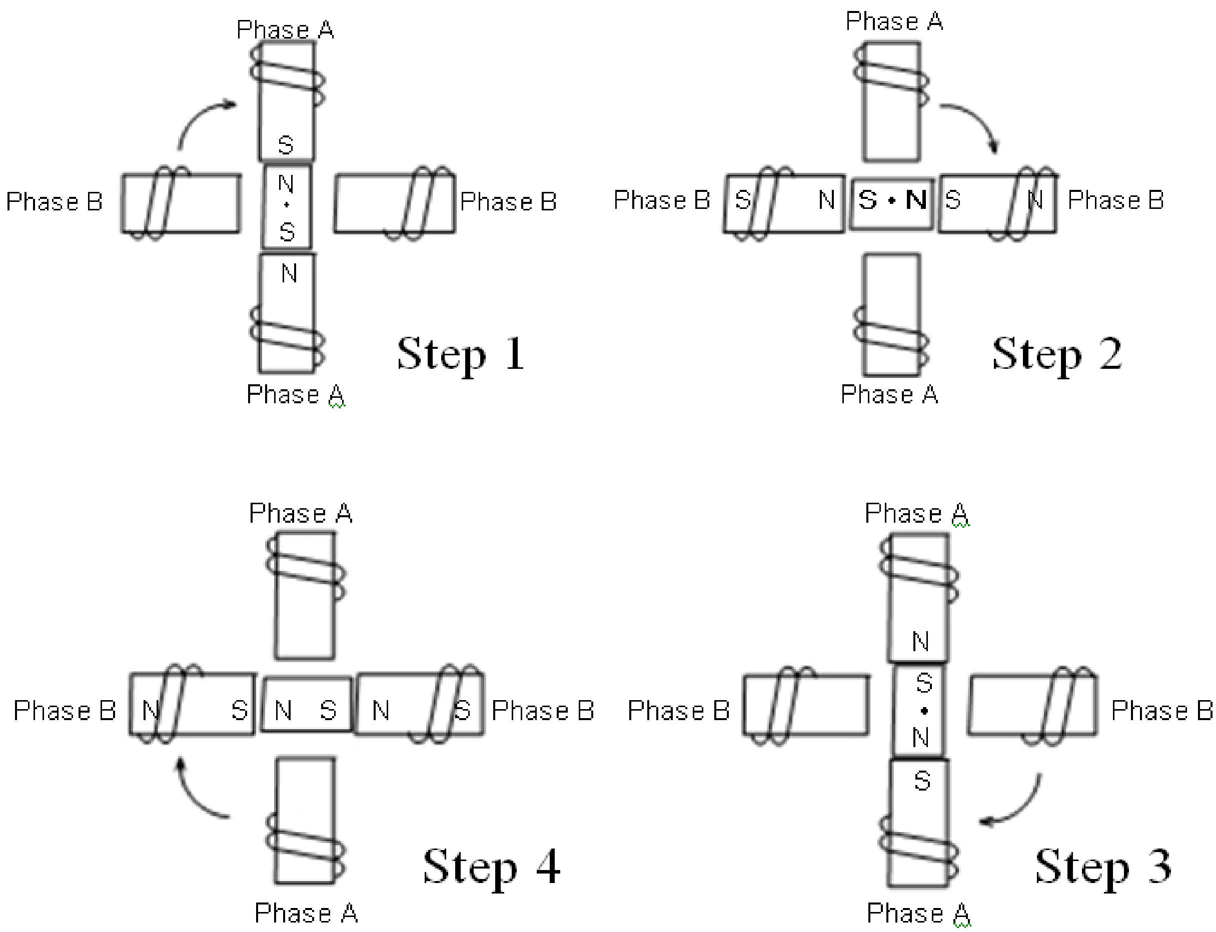


Figure – 3.2: "One phase on" stepping sequence for two phase motor.

The stepping sequence illustrated in figure 2 is called "one phase on" stepping. A more common method of stepping is "two phase on" where both phases of the motor are always energized. However, only the polarity of one phase is switched at a time, as shown in figure 3.3. With two phase on stepping the rotor aligns itself between the "average" north and "average" south magnetic poles. Since both phases are always on, this method gives 41.4% more torque than "one phase on" stepping, but with twice the power input.

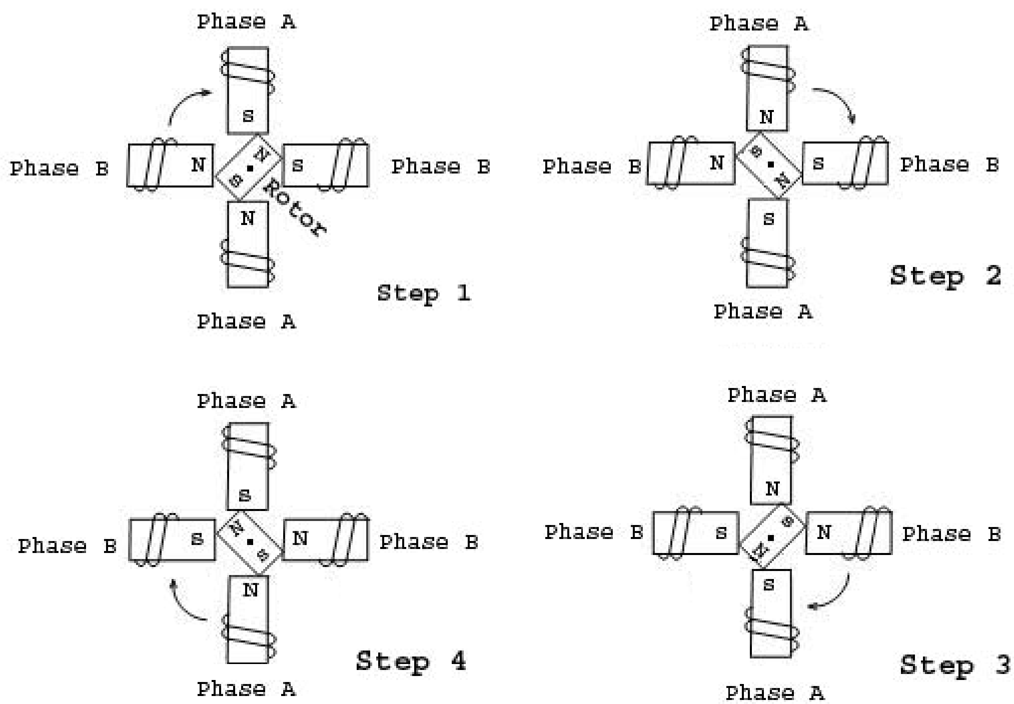


Figure - 3.3: "Two phase on" stepping sequence for two phase motor.

3.2.2 Half Stepping

The motor can also be "half stepped" by inserting an off state between transitioning phases. This cuts a stepper's full step angle in half. For example, a 90° stepping motor would move 45° on each half step, figure 3.4. However, half stepping typically results in a 15% - 30% loss of torque depending on step rate when compared to the two phase on stepping sequence. Since one of the windings is not energized during each alternating half step there is less electromagnetic force exerted on the rotor resulting in a net loss of torque.

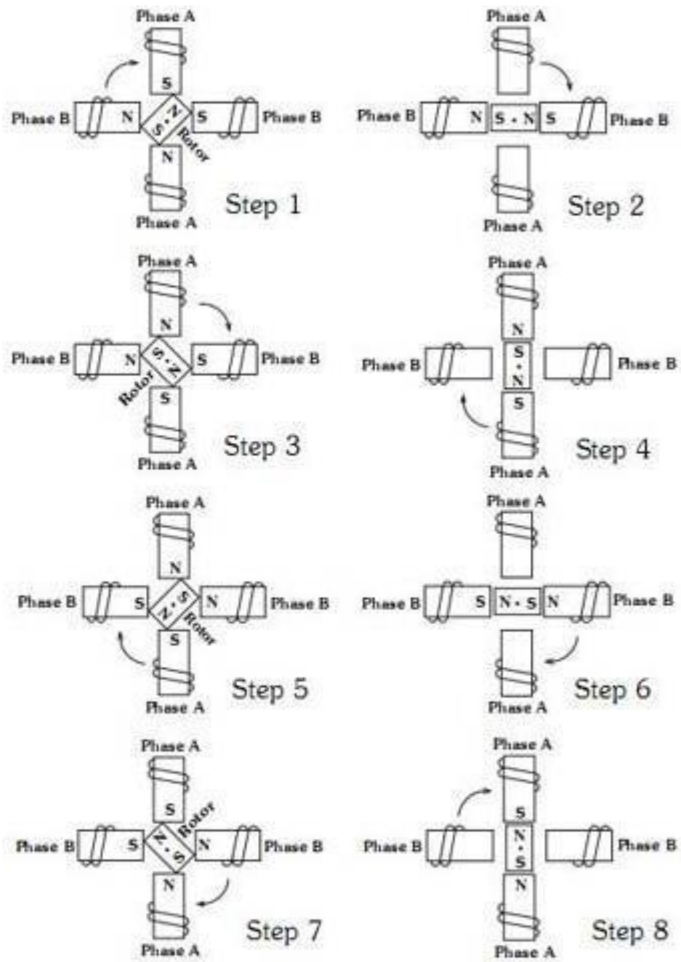


Figure - 3.4: Half-stepping - 90° step angle is reduced to 45° with half-stepping.

3.3 Types of Stepper Motor According To The Winding

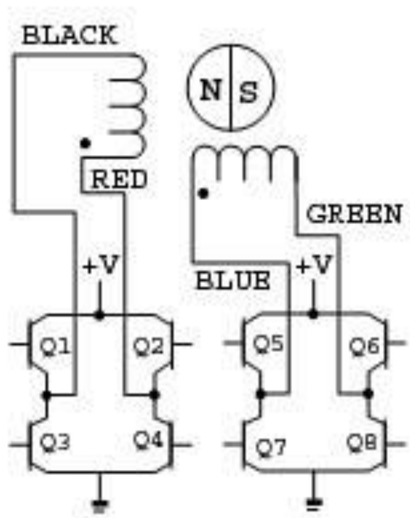
There are two types of stepper motors depending on how the stator coils are constructed. These are :

1. Bipolar Stepper Motor

2. Unipolar Stepper Motor

3.3.1 Bipolar Stpper Motor

The two phase stepping sequence described utilizes a “bipolar coil winding.” Each phase consists of a single winding. By reversing the current in the windings, electromagnetic polarity is reversed. The output stage of a typical two phase bipolar drive is further illustrated in the electrical schematic diagram and stepping sequence in figure 3.5. As illustrated, switching simply reverses the current flow through the winding thereby changing the polarity of that phase.



CW Rotation ↓

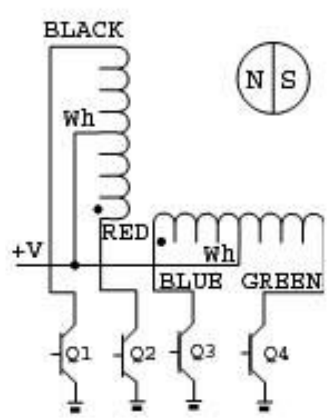
Bipolar Step	Q2-Q3	Q1-Q4	Q6-Q7	Q5-Q8
1	ON	OFF	ON	OFF
2	OFF	ON	ON	OFF
3	OFF	ON	OFF	ON
4	ON	OFF	OFF	ON
1	ON	OFF	ON	OFF

CCW Rotation ↑

Figure - 3.5: Wiring diagram and step sequence for bipolar motor.

3.3.2 Unipolar Stepper Motor

Another common winding is the unipolar winding. This consists of two windings on a pole connected in such a way that when one winding is energized a magnetic north pole is created, when the other winding is energized a south pole is created. This is referred to as a unipolar winding because the electrical polarity, i.e. current flow, from the drive to the coils is never reversed. The stepping sequence is illustrated in figure 3.6. This design allows for a simpler electronic drive. However, there is approximately 30% less torque available compared to a bipolar winding. Torque is lower because the energized coil only utilizes half as much copper as compared to a bipolar coil.



CW Rotation ↓

Unipolar step	Q1	Q2	Q3	Q4
1	ON	OFF	ON	OFF
2	OFF	ON	ON	OFF
3	OFF	ON	OFF	ON
4	ON	OFF	OFF	ON
1	ON	OFF	ON	OFF

CCW Rotation ↑

Figure - 3.6: Wiring diagram and step sequence for unipolar stepper motor.

3.4 Resonance

Stepper motors have a natural resonant frequency as a result of the motor being a spring-mass system. When the step rate equals the motor's natural frequency, there may be an audible change in noise made by the motor, as well as an increase in vibration. The resonant point will vary with the application and load, but typically occurs somewhere between 70 and 120 steps per second. In sev ere cases the motor may lose steps at the resonant frequency. Changing the step rate is the simplest means of avoiding many problems related to resonance in a system. Also, half stepping or micro stepping usually reduces resonance problems. When accelerating to speed, the resonance zone should be passed through as quickly as possible.

3.5 Speed-Torque Characteristic in Terms of Voltage and Current

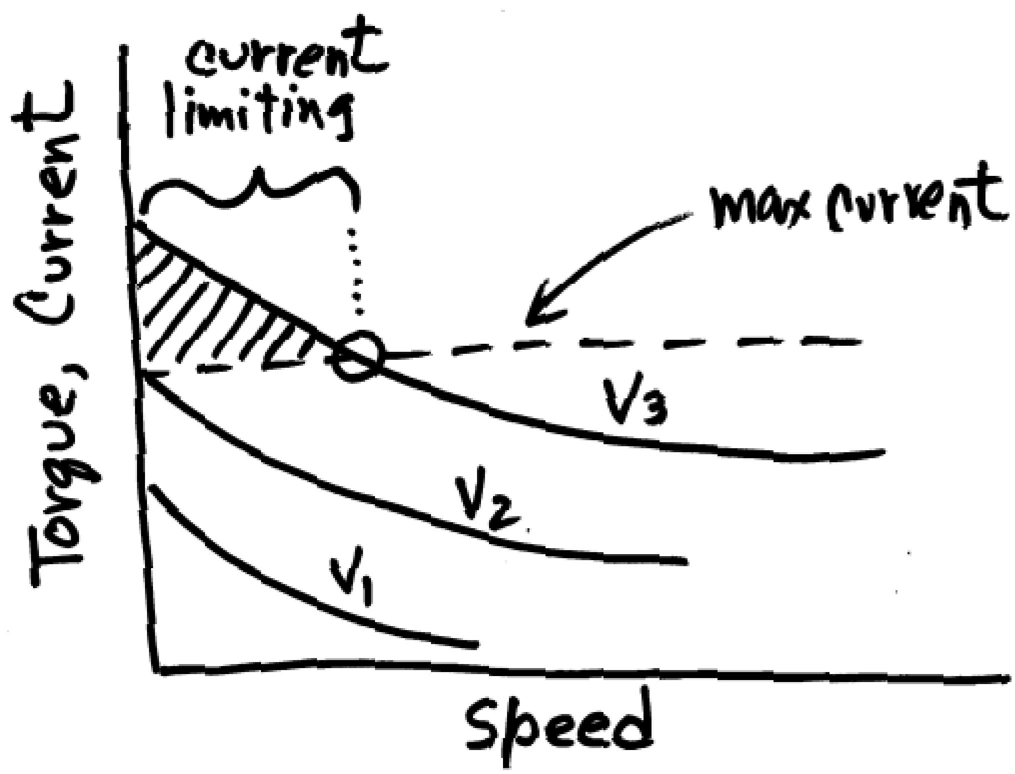


Figure – 3.7: Speed and Torque characteristic of stepper motor

A stepper motor is basically a constant power transducer, where power is the speed-torque product. Therefore, one defining characteristic of step motors is that, at a given voltage, the faster they run the less torque they produce. You trade speed for torque. The way to get more torque is to run them at higher voltage. However, if you run the motor at more than it's rated nemplate voltage, you will draw too much current at low speeds and motor failure from overheating is the likely result. Most commercial step motor drivers include a way to limit the maximum current through the motor windings. This allows high voltages to be used (operating voltages commonly range from 3 to 25 times the nameplate rated voltage) for high torque, high speed applications without burning out the motor.

The graph above shows a stepper being operational at three different voltages, with $V1<V2<V3$. In all cases, increasing the voltage increases the current (and thus the torque) proportionally. At some point however, increasing the voltage will exceed the motor current limit, and some form of current limitation is required at lower speeds**.**

Chapter 4

Microcontroller and Programming

4.1 Micro-Controller

A microcontroller is an entire computer manufactured on a single chip. Microcontrollers are usually dedicated devices embedded within an application e.g. as engine controllers in automobiles and as exposure and focus controllers in cameras. In order to serve these applications, they have a high concentration of such as serial ports, parallel input/output ports, timers, counters, interrupt control, analog-to-digital converters, random access memory,read only memory, etc. The I/O, memory, and on-chip peripherals of a microcontroller are selected depending on the specifics of the target application. Since micro-controllers are powerful digital processors, the degree of control and programmability they provide significantly enhances the effectiveness of the application.

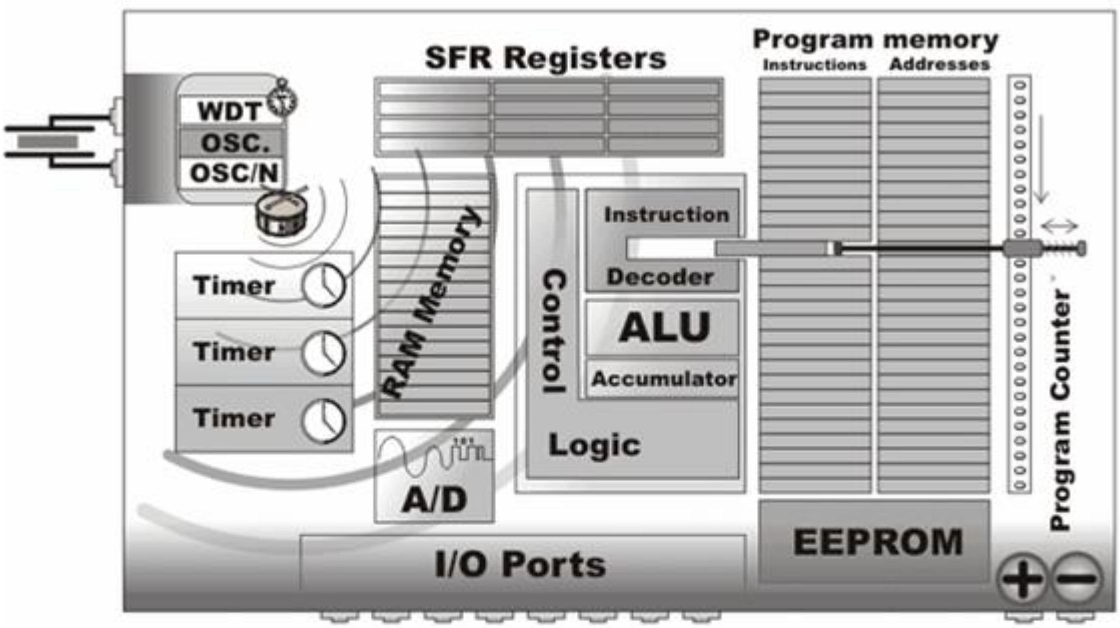


Figure - 4.1: Block diagram of a Microcontroller

Embedded control applications also distinguish the microcontroller from its relative, the general purpose microprocessor. Embedded systems often require real-time operation and multitasking capabilities. Real-time operation refers to the fact that the embedded controller must be able to receive and process the signals from its environment as they are received. Multitasking is the capability to perform many functions in a ***simultaneous or quasi-simultaneous manner***.

The various components of the MCU shown in Figure 4.1 are explained below **:**

Random Access Memory (RAM): RAM is used for temporary storage of data during runtime.

ROM: ROM is the memory which stores the program to be executed.

SFR Registers: Special Function Registers are special elements of RAM.

Program Counter: This is the "*engine*" which starts the program and points to the memory address of the instruction to be executed. Immediately upon its execution, value of counter increments by 1.

Control Logic: As the name implies, it ***supervises*** and ***controls*** every aspect of operations within MCU, and it cannot be manipulated. It comprises several parts, the most important ones including: ***instructions decoder, Arithmetical Logic Unit (ALU) and Accumulator***.

A/D Converter: A/D stands for ***analog to digital***. They convert analog signals to digital signals.

I/O Ports: To be of any practical use, microcontrollers have ports which are connected to the pins on its case. Every pin can be designated as either input or output to suit user's needs.

Oscillator: This is the rhythm section of the MCU. The stable pace provided by this instrument allows harmonious and synchronous functioning of all other parts of MCU.

Timers: Timers can be used for measuring time between two occurrences and can also behave like a counter. The *Watchdog Timer* resets the MCU every time it overflows, and the program execution starts a new (much as if the power had just been turned on).

Power Supply Circuit: this powers the MCU.

4.2 MCU AT89S52

This is the processor. It coordinates the operation of the robotic arm by collecting information from the Flash Memory, interprets and then execute the instructions. It is the heart of the whole system.

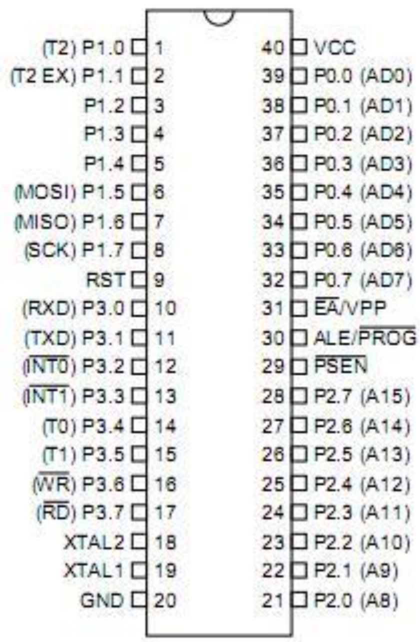


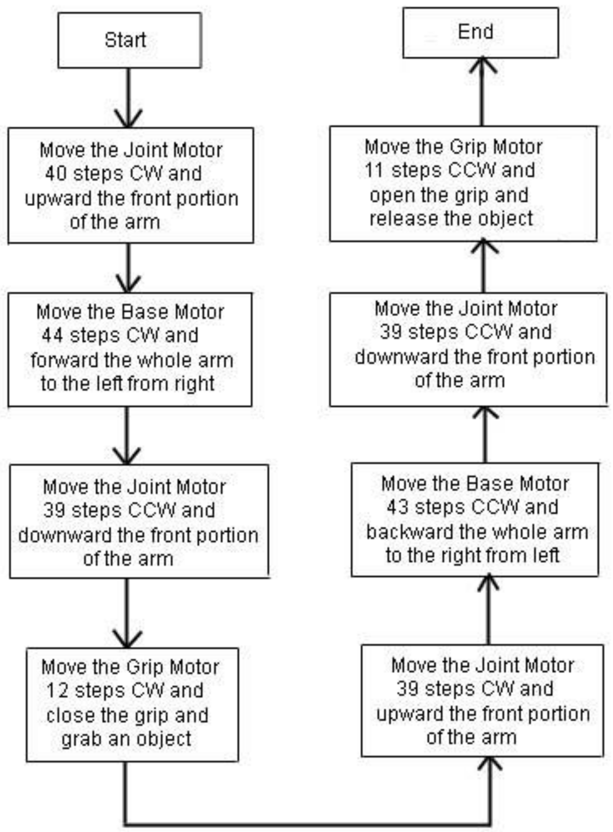
Figure - 4.2 : Pin configuration of AT89S52

4.2.1 Description of AT89S52

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

4.3 Flow Chart



4.4 Source Code

4.4.a Source Code in C Language

```
/* Declaring the header file used for 8051
microcontroller family */
#include<reg51.h>

/* Declaring the data for running the stepper motor CW */
unsigned char data1[] = {0x01,0x02,0x04,0x08};

/* Declaring the data for running the stepper motor CCW */
unsigned char data2[] = {0x08,0x04,0x02,0x01};

/* Declaring the global variables used as counting variable */
unsigned char x,y;

/* Declaring the function prototypes */
void delay();
void b_m_forward();
void b_m_reverse();
void j_m_upward();
void j_m_downward();
void g_m_close();
void g_m_open();

/* Start of main function */
void main()
{
        /* Initializing the microcontroller's ports */
        P1 = 0x00;
        P2 = 0x00;
        P3 = 0x00;

        /* Calling the functions sequencially for moving the
         robotic arm right to left */
        j_m_upward();
        b_m_forward();
        j_m_downward();
        g_m_close();

        /* Calling the functions sequentially for moving the
         robotic arm from left to right */
        j_m_upward();
        b_m_reverse();
        j_m_downward();
        g_m_open();
}
/* End of main function */
```

```
/* Function used for delay */
void delay()
{
	unsigned char i,j,k;
	for(i=0;i<2;i++)
		for(j=0;j<255;j++)
	for(k=0;k<255;k++);
}

/* Funtion used for moving the whole arm 44 steps
from right to left */
void b_m_forward()
{
	for(y=0;y<=10;y++)
	{
		for(x=0;x<=3;x++)
		{
			P1 = data1[x];
			/* Sending data to port 1 */
			delay();
		}
	}
}

/* Function used for moving the whole arm 43 steps
from left to right */
void b_m_reverse()
{
	for(y=0;y<=10;y++)
	{
		for(x=0;x<=3;x++)
		{
			P1 = data2[x];
			/* Sending data to port 1 */
			delay();
		}
	}
}

/* Function used to move 40 steps upward the part of
the arm from shoulder joint to grip */
void j_m_upward()
{
	for(y=0;y<=9;y++)
	{
```

```
                for(x=0;x<=3;x++)
                {
                        P2 = data1[x];
                        /* Sending data to port 2 */
                        delay();
                }
        }
}

/* Function used to move 39 steps downward the part
of the arm from shoulder joint to grip */
void j_m_downward()
{
        for(y=0;y<=9;y++)
        {
                for(x=0;x<=3;x++)
                {
                        P2 = data2[x];
                        /* Sending data to port 2 */
                        delay();
                }
        }
}

/* Function used to close the grip and
grab an object */
void g_m_close()
{
        for(y=0;y<=2;y++)
        {
                for(x=0;x<=3;x++)
                {
                        P3 = data1[x];
                        /* Sending data to port 3 */
                        delay();
                }
        }
}

/* Function used to open the grip and
release the object */
void g_m_open()
{
        for(y=0;y<=2;y++)
        {
                for(x=0;x<=3;x++)
```

```
        {
                P3 = data2[x];
                /* Sending data to port 3 */
                delay();
        }
    }
}
```

4.4. b Source Code in Assembly Language

```
        ORG 00H

MAIN:   MOV     P1,00H
        MOV     P2,00H
        MOV     P3,00H
        MOV     A,#88H
        MOV     R0,#41
JNTM:   MOV  P2,A                   ; moves joint motor  upward
        RR   A
        ACALL   DELAY
        DJNZ R0,JNTM

        MOV     A,#88H
        MOV     R0,#45
BSM:    MOV  P1,A                   ; moves base motor counter clock wise
        RR   A
        ACALL   DELAY
        DJNZ R0,BSM

        MOV     A,#88H
        MOV     R0,#49

JNTM1:  MOV  P2,A                   ; moves joint motor  downward
        RL   A
        ACALL   DELAY
        DJNZ R0,JNTM1

        MOV     A,#88H
```

```
          MOV       R0,#48

          MOV       A,#88H
          MOV       R0,#13
GRPM:     MOV  P3,A                  ; moves grip motor clockwise
          RR    A
          ACALL     DELAY
          DJNZ R0,GRPM

          MOV       A,#88H
          MOV       R0,#41

JNTM3:    MOV  P2,A                  ; moves joint motor  upward
          RR    A
          ACALL     DELAY
          DJNZ R0,JNTM

          MOV       A,#88H
          MOV       R0,#48

BSM1:     MOV  P1,A                  ; moves base motor clockwise
          RL    A
          ACALL     DELAY
          DJNZ R0,BSM1

          MOV       A,#88H
          MOV       R0,#49

JNTM4:    MOV  P2,A                  ; moves joint motor  upward
          RL    A
          ACALL     DELAY
          DJNZ R0,JNTM1

          MOV       A,#88H
          MOV       R0,#4
GRPM1:    MOV  P3,A                  ; moves grip motor counter clockwise
```

```
        RL    A
        ACALL       DELAY
        DJNZ R0,GRPM1

DELAY:                              ; delay subroutine
        MOV     R7,#8
WAIT2:  MOV     R6,#0FFH
WAIT1:  MOV     R5,#0FFH
WAIT:   DJNZ R5,WAIT
        DJNZ R6,WAIT1
        DJNZ R7,WAIT2
        RET
        END
```

4.5 Control Circuit Operation

This is the control panel of the system as it oversees the operations of the mechanical arm. The MCU AT89S52 of the control unit acts as the brain of the control panel as it coordinates all the activities of the other devices. When power (+5V) was supplied to the control unit, the MCU started off by loading the program from the FLASH ROM, *interpreted* and *executed* the instruction codes through the various operational principles of the 8051 based(AT89S52) microcontroller.

The AT89S52 then sends signal to the stepper motors to move according to the sequence of the movements of the robotic hand . The stepper motor (M2) at the shoulder first moves fourty times twisting the rope in the ribon to cause an upward movement of the hand . The stepper motor at the base (M1) moves next stepping fourty four times and makes the connected gear to cause the movement of the arm 180° forward causing the whole structure to turn from left to right. Again the motor M2 moves to cause the arm go downward by releasing the rope from the ribbon. Then the stepper motor at the wrist(M3) moves twelve times CW and the fingers grab the object to be picked . Then M2 moves the arm up while M1 moves (rotates the structure) from right to left . After that M2 moves the arm down and then M3 moves backward(CCW) to release the object . Measured capacity of our robotic arm is shown in **Appendix B**.

Discussions and Conclusion

1. To Make Sure That the Motor Is OK

Before making the project we had to make sure that the stepper motors are ok. Because in our country, brand new stepper motors are not available. So we had to buy old, low level efficient stepper motors from second hand market named “Dholaikhal”. But the shop owner didn’t give us any guarantee on the motors.

To make sure that the motors are ok, we first test the continuity of the coils of the motor, then we check the resistance of each coil and resistance from the center tapped wire to the both ends of each coil.

To be a good stepper motor

- The continuity of each coil has to be present.
- The resistance of each coil has to be same.
- The resistances from the center tap to both ends of a coil has to be same.

2. To Find Out The Proper Sequence of The Wires of The Motor

To run the stepper motor properly the wires of the motor have to be connected sequentially with the microcontroller via the interfacing IC(ULN 2003).

To find out the sequence of the wires of the motor we employed a popular hand rule as follows:

- first we connected the positive end of the power supply to the common end(s) of the stepper motor.

- Then we connect the ground wire of the power supply for an instant with the other four wires of the two coils by considering a sequence in the wires one by one and observe whether the motor steps in only one direction or not. And repeat the process by changing the sequence of the wire until the motor turns in only one direction. If the motor turns in only one direction then the sequence of the wires is ok.

3. Conclusion

In this paper we have interfaced the robot with an I/O device and our method allows for storing more programs to enhance more functionality. From our work, we deduced that in comparison to humans, robots can be much stronger and are therefore able to lift heavier weights and exert larger forces. They can be very precise in their movements, reduce labor costs, improve working conditions, reduce material wastage and improve product quality. This is why they're very important in industries because the overall objective of industrial engineering is productivity .

Meanwhile, intelligent Control is the discipline that implements Intelligent Machines (IMs) to perform anthropomorphic tasks with minimum supervision and interaction with a human operator. This project can be further enhanced to as a multi-disciplinary project involving electrical and mechanical engineers to work together to create more complex, intelligent robots controlled by the 8051 based micro-controller.

APPENDIX A

Specifications of Power Supplies and Stepper Motors

Name	**Specifications**
Computer Power Supply	12V dc, 10A
Adaptor	12V dc, 500mA
Base motor	12V dc, 0.6A. 7.5° per step
Joint motor	12V dc, 0.6A. 7.5° per step
Grip motor	12V dc, 0.6A. 7.5° per step

APPENDIX B

To measure the maximum capacity of the robotic arm we carry different objects of different weights and observe the results from the experiments.

Weights Carried	**Result Observed**
75 gm	Enable to carry
80 gm	Enable to carry
100 gm	Enable to carry
140 gm	Enable to carry
200 gm	Enable to carry
320 gm	Enable to carry
340 gm	Unable to carry
380 gm	Unable to carry
400 gm	Unable to carry

So the maximum capacity of the robotic arm to carry an object is not more than 320 gm.

Bibliography

[1] Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay "The 8051 Microcontroller and Embedded Systems Using Assembly and C(Second Edition)" Publication of Prentice Hall India.

[2] Satya Ranjan "Dev Robotics Technology and Flexible Automation" Publication of Tata McGraw-Hill.

[3] Engelbarger, J.F., "Robotics in practice", American Management Association,1980.

[4] Albus, J.S. Brains, Behaviour Robotics, Byte Books, Subsidiary of McGraw-Hill Peterborough, New Hampshire, 1981.

[5] Saeed B. Niku "Introduction to Robotics Analysis, Systems, Applications" Publication of Prentice Hall India.

[6] Flora, P.C., ed., "Robotics Industry Directory, 4^{th} Edition", Publication of Technical Database Corporation, Conroe, Texas, 1984.

[7] Lundstrom**,** G.,B. Glemme and B. W. Rooks, " Industrial Robots Gripper Review", IFS Publications, Bedford, UK, 1977.

[8] Schreiber, R.R., "Waukesha Welcomes Robots to Its Foundry", Robotics Today, Vol. 6, No. 2, April, 1984.

[9] Nakagawa, Y., "Automatic Visual Inspection Solder Joints on Printed Circuit Boards", Proceedings of SPIE, Vol.336, Robot Vision, May,1982.

[10] Tesar, D., and H. Lipkin, “Assessment for the Man-Machine Interface Between the Human Operator and the Robotic Manipulator ”, NSF Grant/DOE Contact, CIMAR, Gainesville, FL, 1983.