TABLE OF CONTENTS

CERTIFICATION OF APPROVAL

Robot manipulator and controller design

By:
Soheil Bonakdar Hashemi

A project dissertation submitted to
The Electrical & electronics engineering program
UNIVERSITI TEKNOLOGI PETRONAS
In partial fulfillment of requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL& ELECTRONIC ENGINEERING)

Approved:

Ap. Dr. MOHD NOH KARSITI
Project supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
DEC 2007

CERTIFICATE OF ORIGINALITY

This is to certify that I am responsible for work submitted in this project, that the original work is my own except as specified in the references and acknowledgments, and that the original work contained herein have not been undertaken or done by unspecified source or person.

SOHEIL BONAKDAR HASHEMI

I would like to dedicate my research, my work, my final year project to my respective and dear mother. She is a mother any lucky person in the world would wish for, caring and kind which sacrificed a lot of her youth age to raise me. I could have never been where I am today if it wasn't for her unlimited support. To reply her in the smallest of ways which I am sure is nothing compared to her effort, I dedicate my 5 years of studying, my FYP. I love you mum.

ABSTRACT

Objectives of this project are to design and manufacture a robot manipulator arm and the controller. This robot is a high precision robot rather than a heavy duty. As it is supposed to reach points in constant speed and with an approximately zero offset error. The robot to be designed in this project will have three degree of freedom. This robot will be controlled manually and also it will have the ability to be controlled in an automated way by using the aim of computer. In this project microcontrollers will be performing as Communication Bridge between the computer and controller of the robot as well as the controller itself.

The robot which is to be manufactured in this project is designed to perform pick and place jobs. The load which is to be treated by this robot is having a weight of 1 kilogram in the worst case. It means the load could either 1 kilogram or less in term of the weight.

The problem to be considered in this project is to fined the source point either manually or automatically exactly as it had been addressed in term of the dimensions and then pick the target piece and deliver it to the destination exactly as it had been addressed by the user in a constant speed and acceptable time.

Scope of study in this project is to design a feedback control loop which is controlling the robot movement in the way that it moves smoothly and it reaches every point in the work space exactly as it had been addressed. this aim will be achieved by studying the position of the robot in term of the pulses which will be sent to the microcontroller from the encoders as feedback information, And then affecting the magnitude of the current which is flowing trough the DC motor armature by aid of H-Bridges and microcontroller. This affect includes maintaining the magnitude of the current, reducing the magnitude of the current, and finally increasing the magnitude of the current.

The position of the robot will be approximated based on the number of pulses which is being detected by microcontroller.

ACKNOWLEDGMENT

First of all I would like to thank god, which gave me the will and power to overcome this magnificent task, to be successful in finishing my FYP project proudly.

Secondly I would like to take the opportunity to thank associated professor Dr. MOHD. NOH KARSITI which helped me all the way, it was with his respective advises and helps which I over came the problems and issues regarding to this project. He was always there for me no matter what and helped me out in the darkest of all times.

Thirdly I would like to thank my dearest of all, my mum (Mrs. MINOO ORDOOKHANI). She helped me all the way both spiritually and financially in order for me to complete the task which was wanted from me. I thank god I have a mother so dedicated so caring which was there for me every time I needed her, thank you mother.

SOHEIL BONAKDAR HASHEMI (6156)
Electrical and Electronics Engineering

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DC: Direct Current

DOF: Degree Of Freedom

PID: Proportional Integral Derivative

PWM: Pulse width modulation

CHAPTER 1

INTRODUCTION

1.1 Introduction to robot:

Articulated robot: An articulated robot is a robot with rotary joints. Articulated robots can be used to lift small parts with great accuracy. They are also known as a jointed-arm. The arm has a trunk, shoulder, upper arm, forearm, and wrist. With the ability to rotate all the joints, these robots generally have six degrees of freedom. These axes are called: X, Y, Z, pitch, yaw, and roll. Pitch is when you move the wrist up and down, yaw is when the hand moves left and right, and roll is the motion made when you rotate your entire forearm. Articulated robots are often used for tasks such as welding, painting, and assembly. The components of an articulated robot could be simply explained as below:

Continuous path: A control scheme whereby the inputs or commands specify every point along a desired path of motion. The path is controlled by the coordinated motion of the manipulator points.

The robot which is to be designed in this project as an articulated one and it has three degree of freedom (DOF).The work space of this robot is designed in a way that it would be able to reach all of the point which are located in front of it and all the points which are under the horizon, as well as all the point which are included in the +90 degree above the horizon.

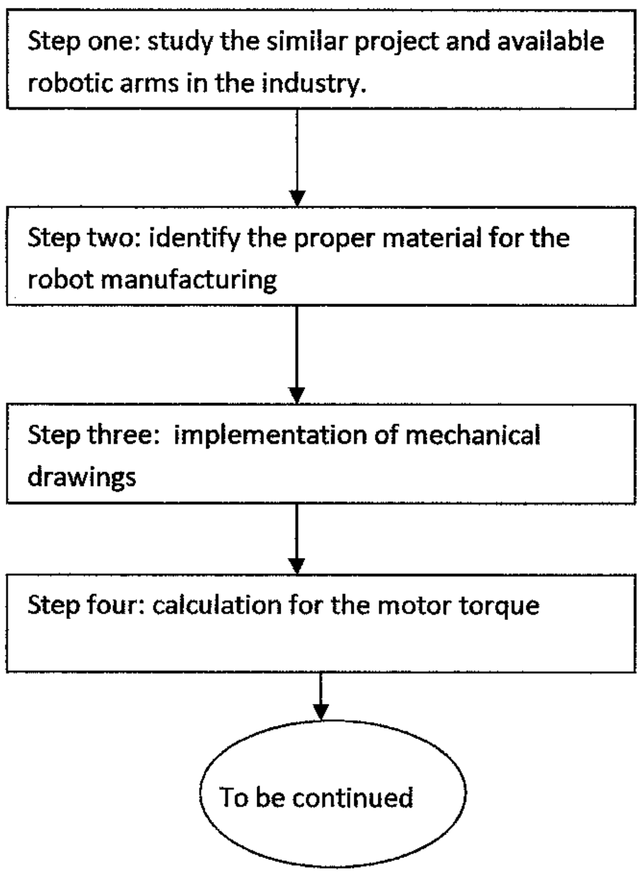
The way that this robot is being controlled is both manual and semiautomatic. Those tasks which don't need to be repeated for a considerable number of times could be done manually or in the other words, the robot could be controlled via the human commands and the operator will have the freedom of moving any of the joints of the robot to perform the task. This job will be performed by using of a joystick which is read into the computer then the movement of the joystick will be interpreted into the electrical signals and these signals will be sent to the micro controllers which are controlling the robot.

The robot is designed to move in several directions. The causing force for this movement will be provided by the DC motors which are located in a not far distance from each joint of the robot. These DC motors will be controlled via the H-Bridge and microcontroller as it will be explained in the following chapters.

In order to manufacture this robot different kind of materials had been studied and considered. Among all of those it had been decided to make a use of aluminum.

The movement of this robot is divided to three separate parts which are namely accelerating portion constant speed portion and finally decelerating portion.

Manufacturing and development of this project within the given time interval was a though job as there have been too many points to be considered and too many circuit design. The following block diagram shows the flow of robot implementation in order to achieve the fully complete robot at the end of FYPII.



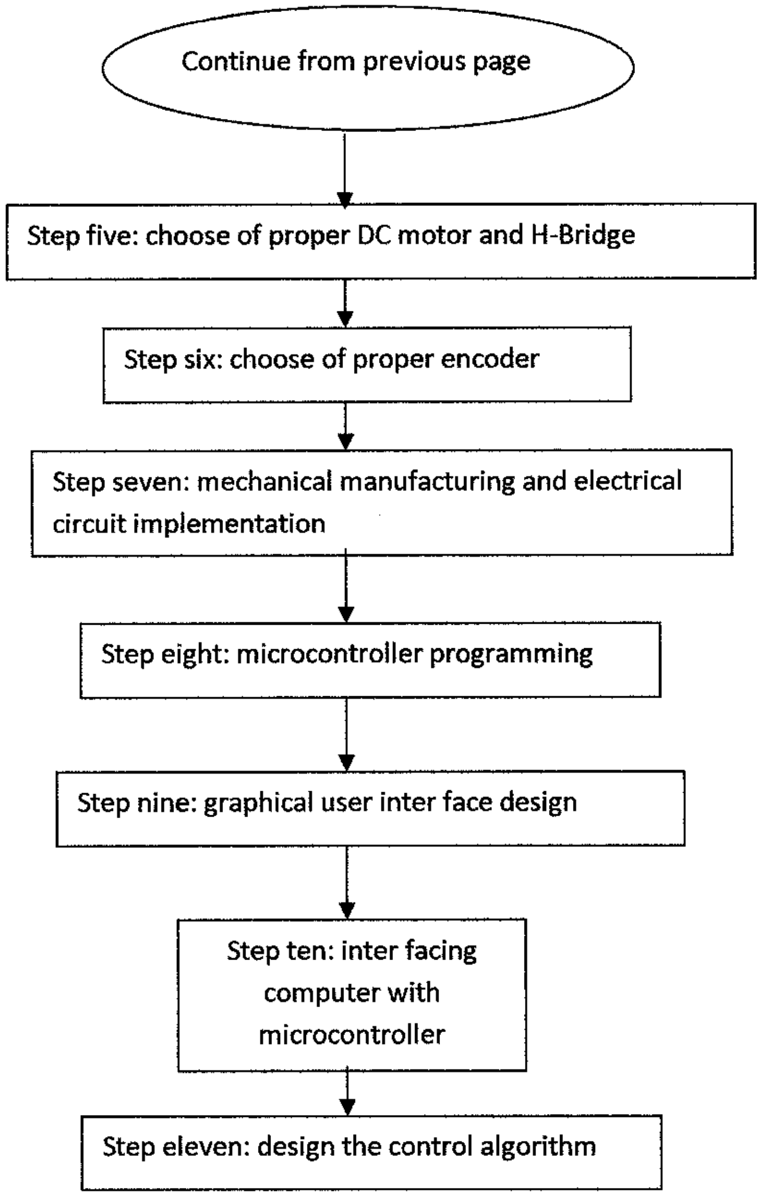


Figure 1 (flow of work)

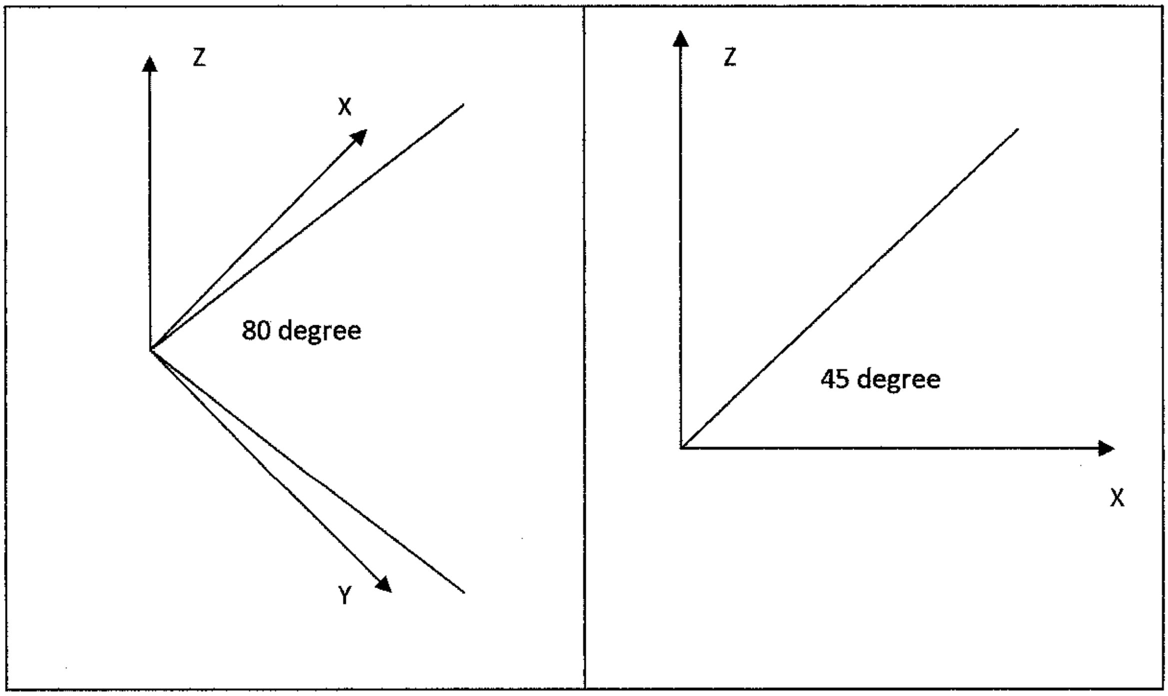
Finally at last but not at least we have to mention that this project have too many points that could be developed in the further steps.

CHAPTER 2

LITERATURE REVIWE AND THEORY

2.1 Robot design:

The robot which is to be designed in this project as an articulated one and it has five degree of freedom (DOF). This means that five different parts of this robot will be able to move in dependently to reach all of the pointes which are located inside the workspace. The number of the joints in this robot had been designed to be 5 to make it clearly different with the other similar projects which had been accomplished in the university technology Petronas. Among the five joints two of them are designed to perform cylindrical movement and the rest of them are designed in a form that they can move vertically in the Z direction. The work space of this robot is designed in a way that it would be able to reach all of the point which are located in front of it and all the points which are under the horizon, as well as all the point which are included in the +45 degree above the horizon. The work space of the robot is illustrated in the figure number one.



Top view horizontal freedom

Side view vertical freedom

Figure 2 (degree of freedom)

2.2Robot specification:

The way that this robot is being controlled is both manual and automatic. For the tasks which are not repeatable the robot could be controlled via the human commands and the operator will have the freedom of moving any of the joints of the robot to perform the pick and place task. This job will be performed by using of a joystick which is read into the computer then the movement of the joystick will be interpreted into the electrical signals and these signals will be sent to the micro controllers which are controlling the robot.

All of the movement which has been performed in the manual mode could be recorded and stored then they could be repeated for a number of times automatically. There is some other ways to give the instructions to the robot to perform some tasks automatically. As the graphical user interface of the computer program is designed the user will be able to give the source point and destination point in the term's of the numerical addresses in the space and then the requested job will be selected from the menu inside the graphical user interface, number of the times that the robot is supposed to repeat an specific task, and then by clicking on the run the robot will perform its job. It could be mentioned that the robot will have the ability of doing a series of tasks in cascaded form and then they will be repeated in the form which they had been asked to repeat.

The robot is designed to move in several directions. The causing force for this movement will be provided by the DC motors which are located in a not far distance from each joint of the robot. These DC motors which are from the heavy duty type of the DC motors are coupled with a gear set which is providing a speed reduction equal to the 516:1. by using of this ratio the required torque for moving different parts of the robot will be provided. Although this reduction will not be ended here, as we know the speed of the dc motor is still extremely high for the requirements of this project, therefore another set of gears which had been designed and manufactured in this purpose will be added into the set to provide enough accuracy for the robot movement.

Besides all the points which had been mentioned in this part there are some other features which make this robot to differ from the similar versions which are available in the market and industry. As a high light it could be mentioned that this robot will be able to estimate the weight of the load which is to be treated in its program. This will make the robot able to estimate the approximate magnitude of the current which is to be used for running the certain motors which will result in the power saving and speed regulation as well as providing a smooth movement for the robot arms.

In order to manufacture this robot different kind of materials had been studied and considered. Among all of those it had been decided to make a use of aluminum as it has an acceptable weight so it require less torque in order to move the arms by using DC motor. Although the other materials such as stainless steel and nylon rode had been used in order to manipulate the mechanical of this robot.

The torque of the DC motors which had been used in this project was one of the main concerns of the designer. The torque of the DC motors had to be enough satisfactory to move all of the aluminum arms which are loaded by the load.

The robot which had been designed in this project have overall number of 3 joints which make this robot to move in all the direction in order to reach all the points inside its working area. By making the robot able to move vertically and horizontally this aim had been achieved.

Another concern of this project was the gear ratio. Gear sets had been used to transfer the power from the DC motors into the moving parts as well as regulating the speed and providing the required torque for the different parts of the robot body.

Beside the gear sets it could be mentioned that the speed of the dc motors will indicate the speed of robot movement. There is always a trade off between the speed and accuracy therefore we tried to provide the option of having both option for the user in the graphical user interface, therefore the user can decide whether he/she prefer a high speed or high accuracy.

After all it could be mentioned that the robot which is to be implemented in this project is counted as a medium robot in term of the size. The following table (table number one) will show the size of all the robot parts.

Table 1 (size specification)

	height	**Width**	**Length**
Base	25 Cm	15 Cm	30 Cm
Shoulder	15 Cm	30 Cm	20 Cm
Arm	------------	15 Cm	20 Cm
Forearm	------------	15 Cm	15 Cm
Wrist	------------	10 Cm	10 Cm
gripper	-----------	-----------	7 Cm

The way that the motors are working is indicated by the microcontroller in term of the pulse width. For sake of reaching certain place the micro controller will decide how long is the path in term of the pulses which are coming from encoder and then this path will be divided into three separate parts one is the accelerating period and the second one is constant speed period and the third and the last part which is decelerating period.

As the robot is assigned to reach some specific place the micro controller will set the magnitude of the current flowing toward the motor by aim of PWM and also H-Bridge.

The following figures illustrate the working schemes of the robot while it is trying to reach the destination

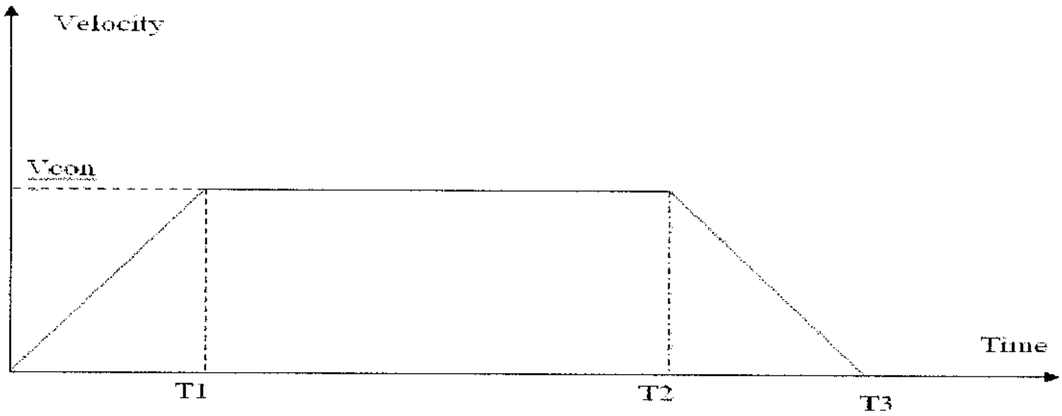


Figure 3 (velocity specification)

As it can be seen from the figure above the robot will start to accelerate for source to time T1 and then when it reaches an appropriate speed it will keep the speed constant till it reaches the time T2. At that time it will start to slow down the speed and finally at the time T3 it will stop completely which is the destination as well.

The scheme of the acceleration is illustrated in the figure below:

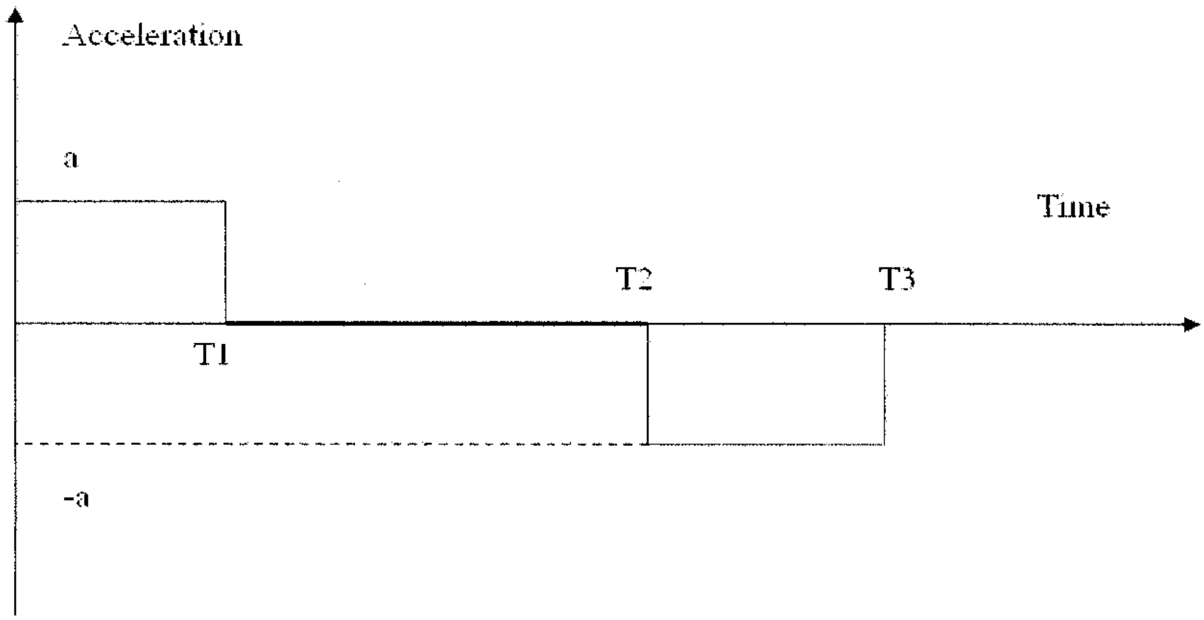


Figure 4 (acceleration analysis)

As it can be seen from the graph above the acceleration during the first period is positive till it reaches the time T1 then the magnitude of acceleration will be zero in order to keep the speed constant then from time T2 to time T3 the acceleration is negative for breaking purpose and for making the motor to stop in exact position.

After all it could be mentioned that for implementing of this robot there had been some limitation in term of the weight and power which made us to make this robot small enough to meet the criteria which had been imposed to this project. Criteria's such as the weight, the power of the DC motors and the amount of the current which could be handled by the circuitry safely are important.

CHAPTER 3

METHODOLOGY

3.1 Electrical characteristic of the robot:

For the electrical part it can be divided into few separate parts for the circuitries.

Micro controllers:

In this part we will construct the circuitry of our micro controller. In this project each DC motor is controlled via an independent micro controller and then all of these microcontrollers will be communicating with each other via the serial communication protocol. For achieving this purpose we will employ one master micro controller which is communicating with the pc and it will take the proper command like destination and source of movement. Then accordingly it will divide the tasks between other micro controllers. Therefore each one of them will be controlling their respective DC motors to reach the point or the angel that they are supposed to reach the figure below will show the connection between the microcontrollers (master and slaves)

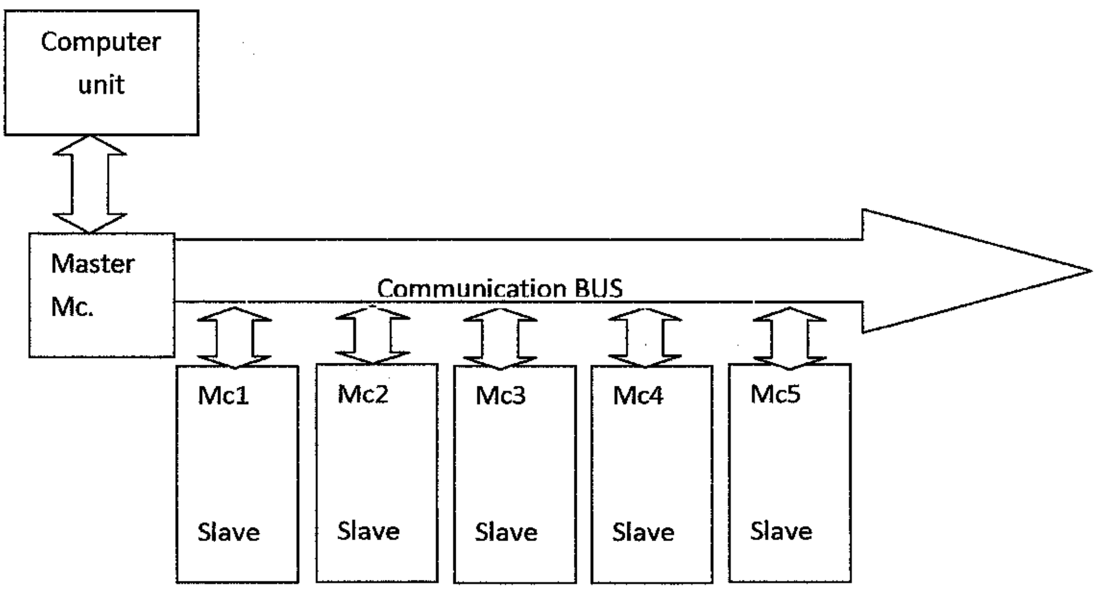


Figure 5 (micro controller communicating with each other)

Each simple microcontroller will be attached to the encoder which is indicating the exact possessions of the DC. Motor and by that information it will detect the magnitude of the voltage which is to be applied to the armature of the DC motor to make it either run or stop.

The information which is provided by the optical encoder is all in terms of number of pulses and therefore the TIMER 1 will be employed to detect the exact possession of the final element and then this information will be interpreted by using the PID algorithm to provide logical output for the micro controller to be represented at the input of the DC motor driver.

The figure below shows the connection between the micro controller and DC motor driver and encoder

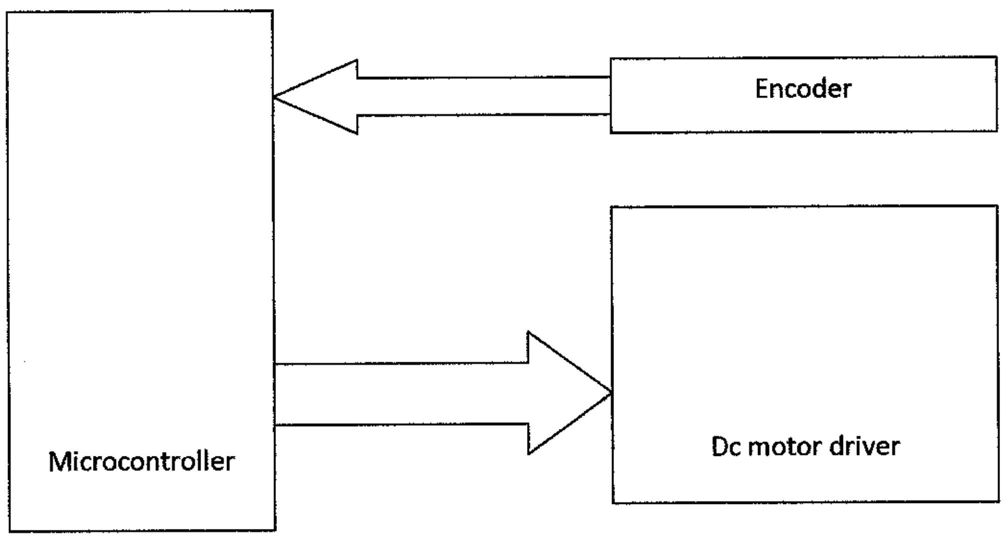


Figure 6 (Micro controller communicating with encoder and H-Bridge)

Timer 2 will be used along with the rest of the programs inside each micro controller to give the proper commands to the H-Bridges. For this purpose the PWM method had been chosen and developed. As the speed of motors and also the distance to the destination is estimated the proper pulse width will be chosen and then by calling the specific functions inside the main program timer 2 will be set to provide the proper pulse width.

The clock which had been used to run the micro controller is equal to 4MHz. although this clock is not very high but it is still satisfactory and will meet all of the needs of our project.

Dc motor driver:

The dc motor driver is playing a very central and important role in this project as we see the torque of motor will be adjusted to the need be changing the voltage at the input of this armature. This purpose could be achieved by different methods one of these methods is by representing the wave forms with different duty cycles at the input of the DC motors but another way is providing a digital to analogue converter which is providing different voltage levels at the input of the DC motor. The amount of the current that will be drawn from the DC motor will not be limited as we are provided with AC power supply.

The H-Bridge which had been used in this project is shown in the following figure.

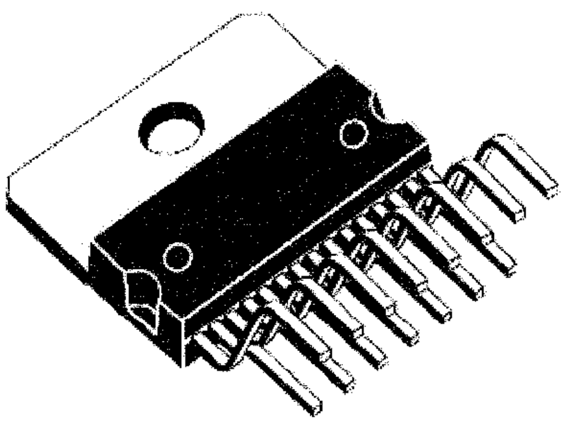


Figure 7 (L298N)

This H-Bridge is able to handle up to 2ampear of current and also each IC could be used to run two different DC motors independently. This property will make this IC more cost effective and also more effective in term of the space which is occupied by the circuitry as well as reducing the heat which will be produced by the IC.

There are two input pin specified for each H-Bridge which trough them the micro controller will decide whether the motor move forward or reverse. This configuration is illustrated in the following table.

Table 2 (specific condition for the H-Bridge input)

Inputs		Function
V_{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don't care

As it can be understood from the table for some specific inputs the motor will move forward and for some other specific inputs the motor will be run by reverse. Also the fast motor stop is supported by this kind of H-Bridge.

The PWM signal from micro controller will be fed into the enabling pin of the H-Bridge so for the time that the motor is supposed to be fed by the H-Bridge the high signal will be sent into the enable pin and for the time that the motor is supposed to be running out of current the H-Bridge will be disabled. Intact the PWM will be applied to the enable pin of the H-Bridge and then the pulse width of the PWM will indicate the duration in which the H-Bridge is supplying current into the DC motor. The internal circuit diagram of the H-Bridge is shown in the following figure:

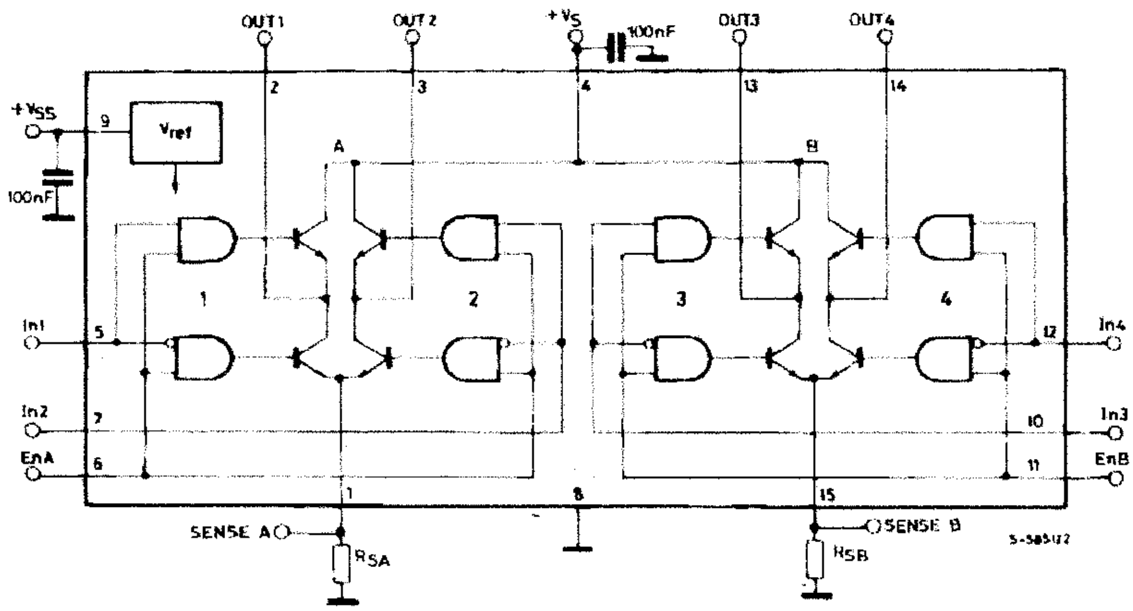


Figure 8 (internal circuit diagram of the H-Bridge)

Optical Encoder:

In order to keep the track of DC motors optical encoders have been employed for this project which will provide acceptable number of pulses per rotation and therefore few pulses per degree of movement. This will provide the enough accuracy for this project to be accepted by the need of industry.

The optical encoder will be powered up by 5 volt and it will provide pulse square at its out put as the shaft is rotating. The pulse square will be shown in two channels which are namely called channel A and channel B. channel A will provide the output square wave as the shaft is rotating in the clock wise direction and channel B will provide the output square wave as the shaft is turning counter clock wise. These pulses will be captured and counted by the TIMER1 inside the microcontroller. By indicating the number of counts the approximate angle will be realized and by multiplying the angle by the length of the arm approximate displacement will be achieved. The encoder which had been used in implementation of this robot is providing a number of 1000 pulses per rotation which will result in 0.36 degree per pulse; this will provide enough accuracy for this project to be accepted by the need of the industry. The following figure is showing the encoder itself.

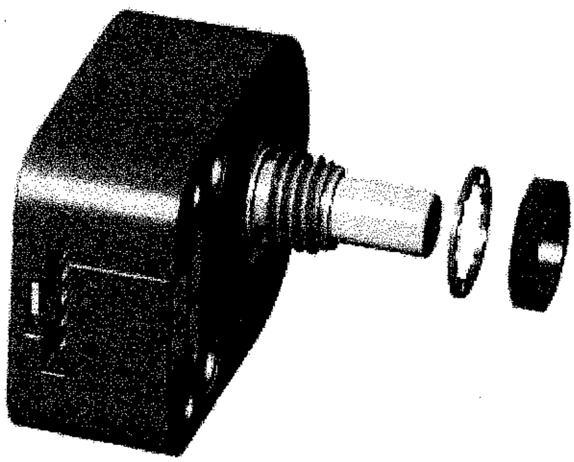


Figure 9 (optical encoder)

DC motors:

As we discussed in previous chapter DC motors have been used to move different parts of the robot body. There have been a wide variety of the DC motors available but a heavy duty DC motor has been used to make sure that it will provide enough torque to move different parts of the robot body.

The type of the dc motor which had been used is a series DC motor in which the field is in series with the armature and it will have the ability to start the movement while it is heavily loaded. The following figure is showing the DC motor which has been used in the implementation of this project.

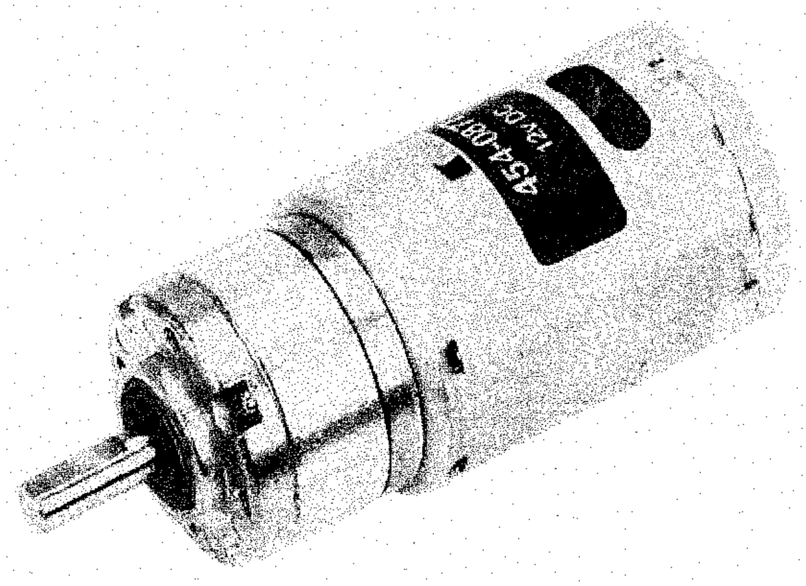


Figure 10 (DC motor)

The physical specification of this DC motor is shown in the following table:

Table 3 (physical specification of DC motor)

Overall body length	87·1
Body diameter	32
Shaft length	17
Shaft diameter	6
Flat length	12
Flat depth	0·5

The electrical specification of this DC motor is shown in the following table:

Table 4 (electrical specification of DC motor)

Voltage (volt)	Angular velocity (round per minute)
4.5V	9.5
6.0V	13
9.0V	20
12.0V	27
15.0V	32

Power supply:

This project is using AC power supply and for this purpose a transformer has been employed to step down the AC power from 240v to 24v which is enough for the need of this project. Also a single phase rectifier will be employed along with this transformer to convert the AC power into the DC power. This power supply is able to handle up to 2ampear of current which later will flow trough the DC motors in order to establish the required torque.

In order to prevent the feed back into the microcontroller we have provided different power supplies for the driver and controller, but at the end of the day these two are common grounded to prevent the problems of having open circuit.

The transformer which had been used in implementation of this project is a multi tap transformer which is able to step down the voltage from 240 to few values in between of 6-24 volts. For the efficient use of the power we have chosen the maximum voltage which is 24 volt this voltage will be rectified and then according to the need of robot will be stepped down to 12 and 5 volt by using DC-DC converters in different parts of the circuitry.

3.2 Mechanical characteristic of the robot:

Calculation:

The torque of the dc motors must be sufficient to hold the arms when all of the arms are extended in zero degree compared to vertical line.

Therefore we will calculate for the moment when highest torque is required and then the rest of time the torque of DC motors are satisfactory.

Below are the calculations:

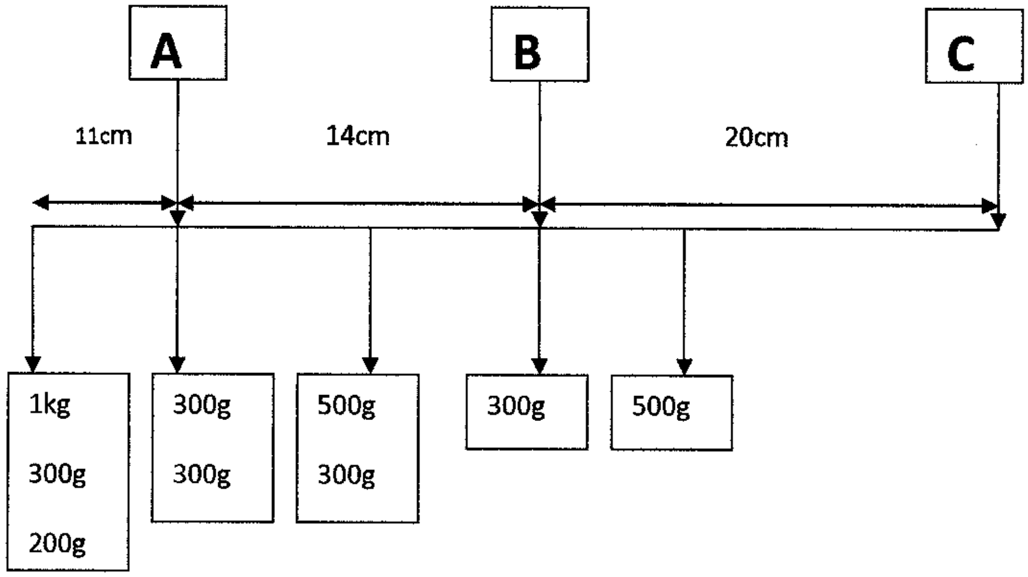


Figure 11 (torque analysis)

PointA➔(1000g+300g+200g+2500g)x10x0.11=1.925Nm

PointB➔ (1000g+300g+200g+2500g)x10x0.25+0.8x10x0.14+10x0.7=

12.495Nm

PointC➔(1000g+300g+200g+2500g)x10x0.45+0.8x10x0.34+0.3x10x0.2+0.5x10x0.1+1 x10x0.27=14.395 Nm

Most of the parts which are to be implemented in this project have been sketched by using the auto cad program. All of these files could be found in the appendix A.

Choice of material:

The materials which are to be used in construction of this project are as follow:

There were different types of materials available for constructing this project which is listed as below:

1- aluminum
2- stainless still
3- plastic

Each of these materials has some positive and negative pointes to be considered in the design. For better production we have listed all of the positive and negative aspects of each material in the tables below to make a better and more accurate decision in choosing the proper material for manufacturing the robots body.

Table 5 (aluminum specifications)

Positive points	**Negative points**
Light	Expensive
Strong	Welding cannot be used
Could be shaped by milling	
Plates could be formed by available facilities inside the lab	
Could be connected by using screws	
In case that one part is malfunctioning it could be replaced in dependent of the other parts	

Table 6 (Stainless steel: specifications)

Positive points	**Negative points**
Proper price	heavy
Plates could be formed by available facilities in side the lab	In case that one part is malfunctioning it couldn't be replaced in dependent of the other parts
Could be shaped by milling	
Strong	
Could be connected by using screws	
Welding could be used	

Table 7 (Stainless steel: specifications)

Positive points	**Negative points**
Proper price	In case that one part is malfunctioning it couldn't be replaced in dependent of the other parts
Could be shaped by milling	Not strong enough
Could be connected by using screws	Plates could be formed by available facilities inside the lab
	Welding cannot be used

By considering all of the positive and negative points of these materials we have chosen the aluminum to be used for building and manufacturing the body of this robot.

But the other two materials have been used as well to make the robot more flexible and also more effective.

For making the body of the robot we have chosen the aluminum but for the joints which is to be used as shoulder and wrist of the robot the nylon bar had been used which according to its properties it has got satisfactory strength against vertical forces and also twisting as long as its length is not more than 15 centimeters.

For the base most of the parts will be made by using the aluminum and also the main shaft will be made of aluminum, gear sets and gripper will be made of aluminum.

For connecting the plate rollers to the shaft and the base and body stainless still will be used. As we know rollers are made of stainless still so for connecting the rollers to the central shaft we made a use of stainless still plates to weld them to the roller and then using the screws to connect them to the central shaft.

Next factor to be considered in this report is the tools which had been used for manufacturing different parts of this project. For the sake of simplicity we start introducing each equipment followed by a short explanation.

Wire cutting machine: Is a machine provided in mechanical department which is cutting the metal pieces by using a wire. This wire will produce electric spark along the path which is to be cut and then these electric sparks will cause the cutting action to occur. The diameter of the wire is nearly 2 micrometers and it is able to cut only metallic pieces as it need the work piece to be conductive.

CNC Milling machine: is an automated high accuracy milling machine which will use computer and program to cut the materials with a high accuracy and by using drills.

Conventional milling machine: is a milling machine provided by university which is used for milling purposes. Trough this project it was used to manufacture some parts which were not available in the market but they were also needed in aim of manufacturing the mechanical components of this robot.

Conventional sawing machine: a sawing machine which had been used to cut the work pieces into approximate sizes before feeding the work piece to the milling machine. This was performed to accelerate the procedure of manufacturing. The sample of the CNC codes which had been used to manufacture different parts of the robot's body is attached in the appendixes.

Filler rod aluminum: as we had to connect the different parts of the robot screws were used but there were some parts that they could not be connected via screws so we had to use aluminum welding.

For making the joints of the project to be able to move inside each other more easily and with out the affect of friction we made a use of bearing. There have been two different types of bearing which one of them is mad of two circles which roll inside each other this type of bearing is shown in the following figure:

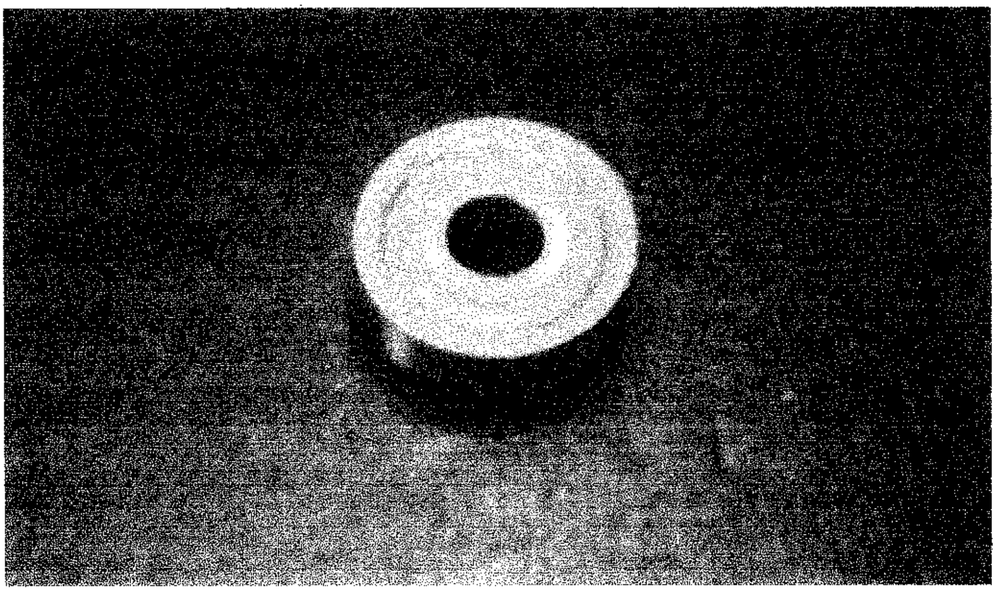


Figure 12 (bearing)

The second type of bearing which has been used in this project is made of two circular plates which are rolling on top of each other .this type of bearing is used to reduce the friction between those plates which are rolling on each other the following figure shows this type of bearing.

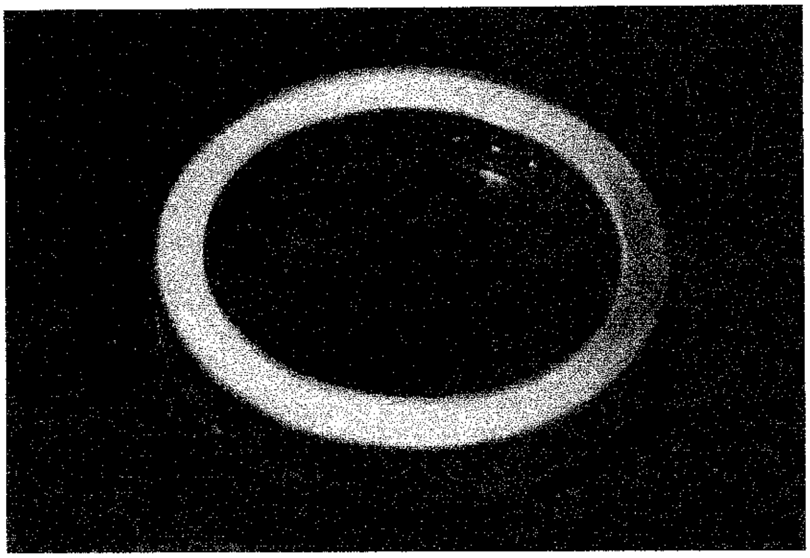


Figure 13 (bearing)

The next issue to be discussed is the thickness of the plates which have been used in implementation of this robot. As we know if the aluminum plates are not thick enough later on we will face the twisting problem as the different parts of the robot move. On the other hand if the plates are very thick their weight will be creating problems as they will produce unwanted torque on the robot due to the affect of the gravity. As a result we have chosen a thickness of 0.4 cm to make sure that we won't face any problem in the future steps.

Gear design was another issue in the project as we know in order to transfer the power from the motor shaft into the arms we had to use either belting or gear sets, which we have chosen the gear sets. Gears are designed according to the criteria of gear design to indicate the pressure angle, external diameter, and the number of the tooth. The pressure angle will be designed according to the amount of the pressure which is applied to each tooth and the number of the tooth will indicate the reduction ratio. For this particular project the pressure angle has been chosen equal to 20 degree and the module number is equal to 1.5.

CHAPTER 4

RESULTS AND DISCUSSION

4.1Electrical aspects of the robot:

As we discussed in the previous parts of this report the robot will be using the power which is provided by its own power supply. The power supply of this robot have been implemented and working up to this point. The following figure is shown in the following figure:

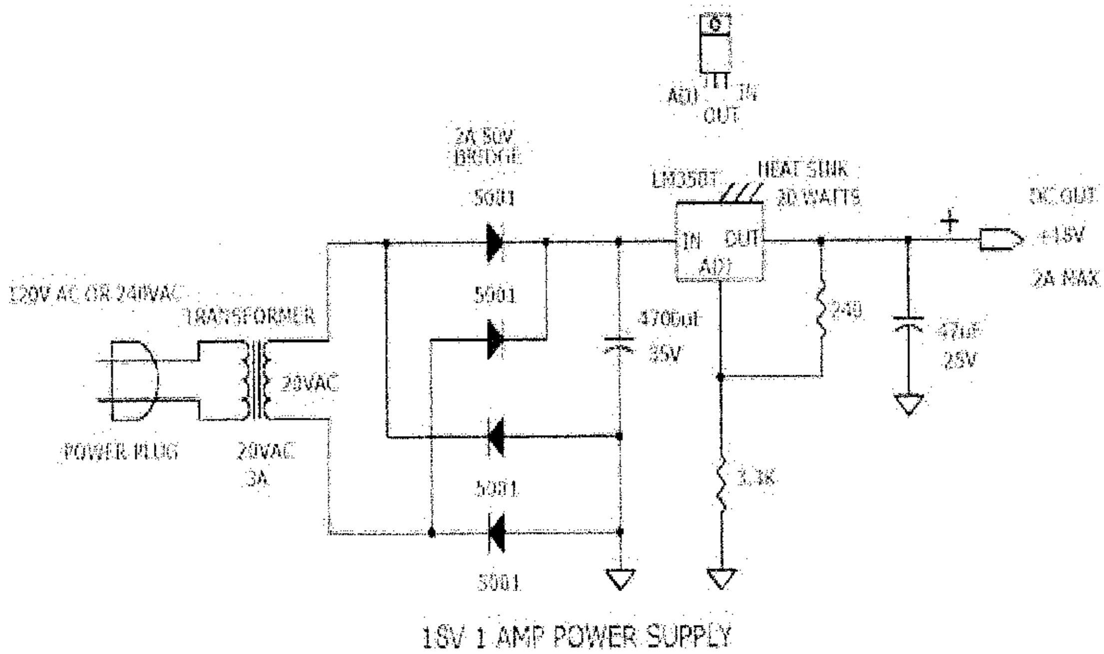


Figure 14 (power supply circuitry)

Next electrical aspect of the robot which has been implemented successfully at this point is the DC motor drivers by using the H-Bridges. These H-Bridges are run by the micro controller and proper PWM which will be briefly shown after this part. The following figure illustrates the circuitry which has been implemented for the DC motor driver:

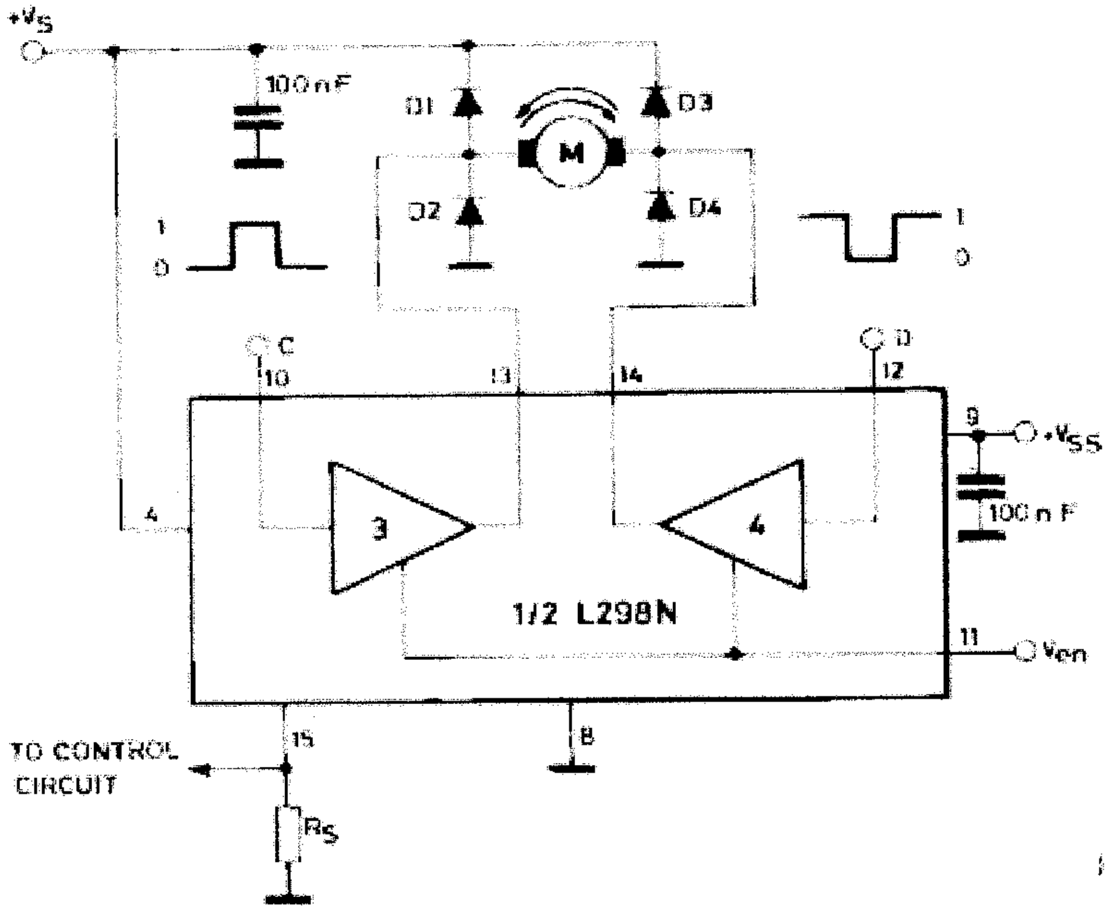


Figure 15 (motor driver)

As we know the encoders will provide a wave form for the microcontroller as an input. The microcontroller will count the number of the pulses to indicate the approximate position of the robotic arm. The function which will carry out this task has been produced and tested. The following paragraph is the CCS-PIC code which will perform the capture and compare task:

```
#include <16F877.h>

#use delay (clock=4000000)

#fuses HS, NOWDT
```

```
Int overflow_count;

Unsigned int32 start_time, end_time;

Int32 pulse_ticks;

#int_TIMER1_isr ()

{

++overflow;      //increment whenever an overflow occurs

}

#int_CCP1

CCP1_isr ()

{

End_time = ccp_1;  //read captured timer ticks

// check for the pulse time (in ticks) accounting for any overflow

// that may have occurred

Pulse_ticks = (0x10000* overflow_count) - start_time+end_time;

If (Pulse_ticks< 2500) output_low(pin_D0);

Else output_high(pin_D0);

// now we set up for the next pulse

start_time=end_time; // end time of this pulse is the start for the next pulse

overflow_count= 0; //clear over flow counts

}

Void main ()

{
```

```
Setup_ccp1(CCP_CAPTURE_RE);

Setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);    //set the timer1 to run at system
                                            //colock/4

Enable_interupts (INT_TIMER1); // un mask timer1 interupts

Enable_interupts (INT_CCP1); // unmask capture event interrupts

Enable_interupts (global); // enable all unmasked interrupts

While (1);

}
```

The next part which have been accomplished up to know is the PWM program which is providing the appropriate output signal with a proper duty cycle to be fed into the DC motor- driver's input in order to run the DC motor perfectly. Here is the program which is accomplishing this task in each micro controller:

```
#include <16F877.h>

#use delay (clock=4000000)

#fuses HS, NOWDT

Int duty_cycle, control_bits, hold_value;

Void main ()

{

Port_b_pullups (TRUE);

Setup_timer_2(T2_DIV_BY_1,99,1); // enable timer2, PR2=99, prescaler=1

Setup_ccp1(CCP_PWM); // enable pwm mode

While (1);
```

```
{
hold_value = (~input_b( ) & 0x7); //get the pin value
While ( (control_bits = (~input_b( )) & 0x7 )) == hold_value);
// wait for the change
Switch (control_bits)
    {
    Case 0x1 : duty_cycle = 35;
        Break;
    Case 0x2 : duty_cycle = 59;
        Break;
    Case 0x3 : duty_cycle = 94;
        Break;
    Default : duty_cycle = 0;

    }
CCP_1 = duty_cycle ;
}
}
```

As we have mentioned during the objectives of this project the project is about to be controlled manually and autonomous as well. In order to accomplish this objective we have made a use of a handle controller for the manual mode which will enable the robot to be controlled manually via the handle controller. The handle controller which have been designed and fabricated for this robot will provide 18 working keys which not all of them is being used for the time being of this project. But the number of the keys will enable the designer to develop the robot to some more complicated functionality ore even develop the number of DOF in the further steps.

The handle controller is made of a PS2 handle which is providing the user with a comfortable grasp system but it could be mentioned that for the sake of copyright none of the circuitry of the original handle have been reused. In this part we have taken all the circuitry of the handle out and then we designed our own circuitry for the handle in order to transfer the proper commands from the handle controller into the master micro controller.

The signal conditioner circuit which is located in between the handle and micro controller will change the thristate commands which are combinations of (high, low, high impedance) into the (high and low) which is understandable for the micro controller. This circuit also will be performing as repeater to eliminate the affect of signal attenuation which might be caused by the long wire which is connecting the handle controller to the circuitry.

In the following figures you will see the circuitry of the signal conditioner and its PCB design as well as its own picture.

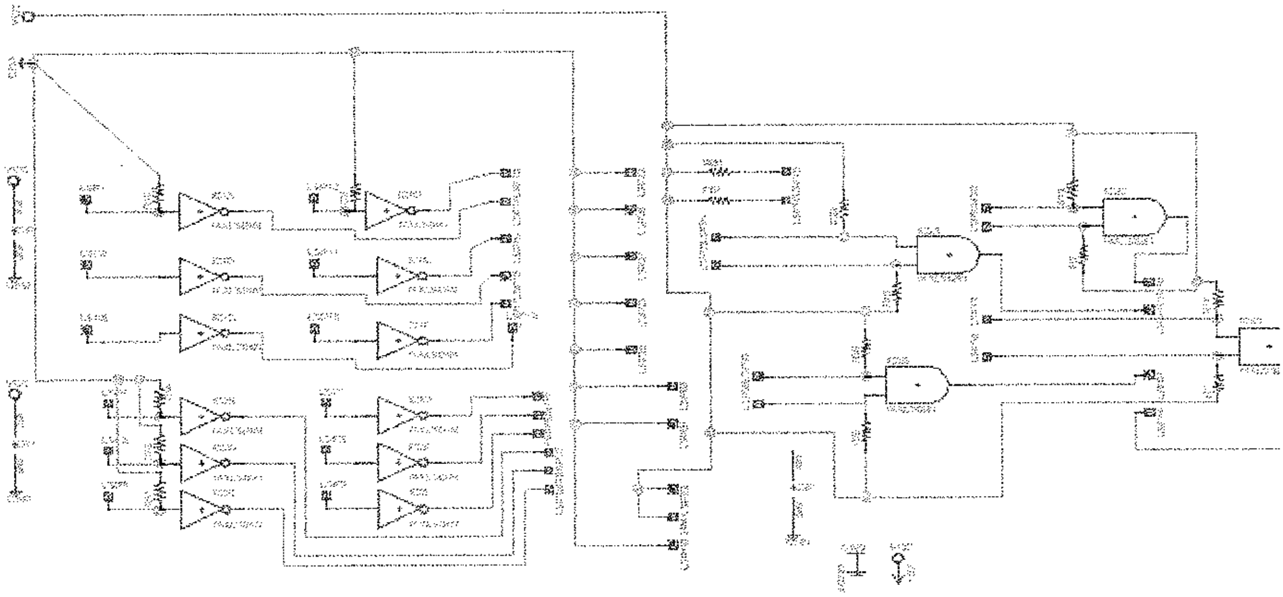


Figure 16 (signal conditioner)

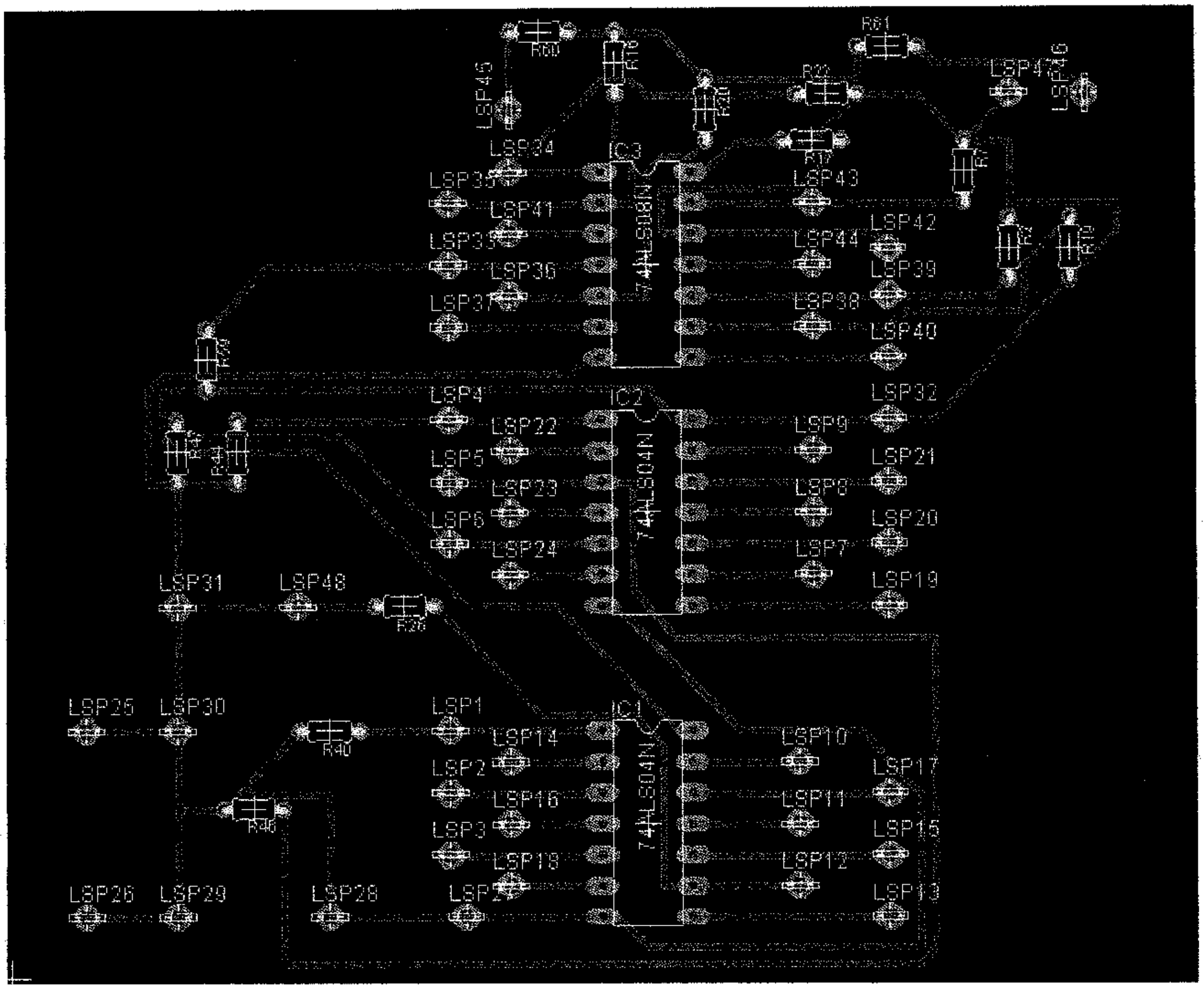


Figure 17 (signal conditioner PCB design)

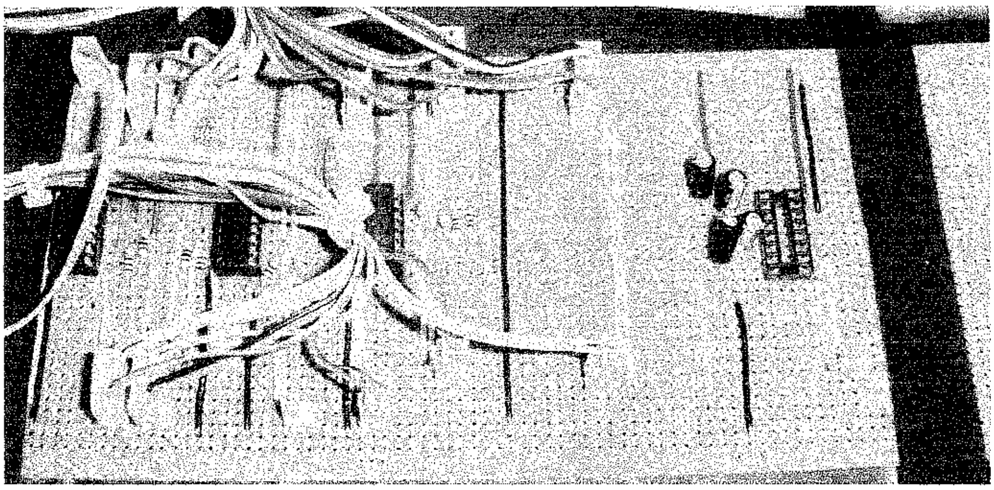


Figure 18 (signal conditioner circuitry)

Next issue to be disscused is the manual main controller. In this section the signal which is from the type of active low will be interperted and according to the result of the proccess which has performed the right angle and also the proper H-Bridge will be activated to enable the robot to move its certaine joints according to the following command which will be given to the microcontroller through the handel controller and the signal conditioner by the user. The program which is being run inside the microcontroller is presented in the following pragraph:

```
#include <16F877.h>

#use delay(clock=8000000)

#fuses NOWDT,HS,NOPUT,NOBROWNOUT,NOPROTECT,NOLVP

//#use standard_IO(D)

//#use i2c(MASTER, SDA=PIN_C4, SCL=PIN_C3)


//c 0-1 d1 3-4 d2 6-7 d3

int8 xstc =0; //sxtc has 4 state 0- {x} no action 1- {s} "digree 1" 2- {t} "digree 2" 3-
{c} "digree 3"

int8 lurd = 0; //lurd has 5 state 0- no action 1- {l} left 2-{u} move up 3-{r} move right
4-{d} move down

int8 fs = 0;  //fs has 2 state 0- {s} slow 1- {f} fast

int8 sts = 0;  //sts has 2 state 0-{st} stop 1- {s} start

int8 master= 0; //This is master number 0 this value will set the msb

int8 kosmac = 0;

int8 cmac = 0;
```

```
//checkf is to check and set the speed of program

void checkf(){

  int8 pin;

  pin = ~(INPUT_D()) & 0x03; //mask 0000 0011 of port D

  if(pin == 1){

    fs = 1;           // if it's 1 it means fast

    //ccp_1 = 100;

  }else if(pin == 2){

    fs = 0;           //if it's 2 it means slow

    //ccp_1 = 64;

  }

  if(xstc == 0)

  {

    fs = 0;           //prior to all other action if the digree is not selected the program is
in slow mode

    //ccp_1 =0;

  }

}


//checks is to check stop and start

void checks(){
```

```
  int8 pin;
  pin = ~(INPUT_B()) & 0x30; //mask 0011 0000 of port B
  if(pin == 0x10){
    sts = 0;            //if it's 0x10 means stop
  }else if(pin == 0x20){
    sts = 1;            //if it's 0x20 means start
  }
  if(xstc == 0)
  {
    sts = 0;            //prior to all other action if the digree is not selected the program
is in fast stop mode
  }
}

//checkl is to check the direction
void checkl(){
  int8 pin;
  pin = ~(INPUT_B()) & 0x03; //mask 0000 0011 of port B
  //pin1 = ~(INPUT_D()) & 0x0C; //mask 0000 1100 of port D
  if(xstc == 0)
  {
```

```
    lurd = 0;


  }else if(xstc == 1){
                    // if user select the first degree it only move right or left
    if(pin == 0x00){
      lurd = 0;        //if it's 0x00 none of them pressed free wheiling
    }else if(pin == 0x01){
      lurd = 2;        //if it's 0x04 means up
    }else if(pin == 0x02){
      lurd = 1;        //if it's 0x08 means down
    }
  }else if(xstc == 2){
                    // if user select the second or third degree it only move up and
down
    if(pin == 0x00){
      lurd = 0;        //if it's 0x00 none of them pressed free wheiling
    }else if(pin == 0x01){
      lurd = 2;        //if it's 0x01 means up
    }else if(pin == 0x02){
      lurd = 1;        //if it's 0x02 means down
    }
```

```
 }else if(xstc == 3){
                        // if user select the second or third degree it only move up and
down
    if(pin == 0x00){
      lurd = 0;         //if it's 0x00 none of them pressed free wheiling
    }else if(pin == 0x01){
      lurd = 2;         //if it's 0x01 means up
    }else if(pin == 0x02){
      lurd = 1;         //if it's 0x02 means down
    }
  }
  lurd = lurd * sts; //ignore direction if the program is not started
}


//checkx is for digree selection
void checkx(){
  int8 pin;
  pin = ~(INPUT_D()) & 0xF0; //mask 1111 0000  of port D d4 t d5 x d6 c d7 s
  if(pin == 0x10){
    xstc = 2;         //if it's 0x10 means digree 2
    //dad = 0xB0;
```

```
  }else if(pin == 0x20){
    xstc = 0;
    sts = 0;   //if it's 0x20 means fast stop
    //dad = 0x00;
  }else if(pin == 0x40){
    xstc = 3;             //if it's 0x40 means digree 3
    //dad = 0xC0;
  }else if(pin == 0x80){
    xstc = 1;             //if it's 0x80 means digree 1
   // dad = 0xA0;
  }
}

void moveit()
{
if(cmac == 0){
  kosmac = xstc;
}else{
  xstc = kosmac;
}
if(xstc ==0){
```

```
    output_low(PIN_C0);
    output_low(PIN_C1);
    output_low(PIN_C3);
    output_low(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
}else if(xstc == 1){
  if(sts == 0){
    output_low(PIN_C0);
    output_low(PIN_C1);
    output_low(PIN_C3);
    output_low(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
  }else if(sts == 1){
    if(lurd ==0){
      ccp_1 = 0;
      cmac = 0;
      output_low(PIN_C0);
      output_low(PIN_C1);
      output_low(PIN_C3);
```

```
  output_low(PIN_C4);

  output_low(PIN_C6);

  output_low(PIN_C7);

}else if(lurd == 1){

  cmac = 1;

  ccp_1 = 64+(fs*36);

  output_high(PIN_C0);

  output_low(PIN_C1);

  output_low(PIN_C3);

  output_low(PIN_C4);

  output_low(PIN_C6);

  output_low(PIN_C7);

}else if(lurd == 2){

  cmac = 1;

  ccp_1 = 64+(fs*36);

  output_low(PIN_C0);

  output_high(PIN_C1);

  output_low(PIN_C3);

  output_low(PIN_C4);

  output_low(PIN_C6);

  output_low(PIN_C7);
```

```
    }
  }
}else if(xstc == 2){
  if(sts == 0){
    output_low(PIN_C0);
    output_low(PIN_C1);
    output_low(PIN_C3);
    output_low(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
  }else if(sts == 1){
    if(lurd ==0){
      cmac = 0;

      output_low(PIN_C0);
      output_low(PIN_C1);
      output_low(PIN_C3);
      output_low(PIN_C4);
      output_low(PIN_C6);
      output_low(PIN_C7);
      ccp_1 = 0;
```

```
    }else if(lurd == 1){
      cmac = 1;
      ccp_1 = 64+(fs*36);
      output_low(PIN_C0);
      output_low(PIN_C1);
      output_high(PIN_C3);
      output_low(PIN_C4);
      output_low(PIN_C6);
      output_low(PIN_C7);
    }else if(lurd == 2){
      cmac = 1;
      ccp_1 = 64+(fs*36);
      output_low(PIN_C0);
      output_low(PIN_C1);
      output_low(PIN_C3);
      output_high(PIN_C4);
      output_low(PIN_C6);
      output_low(PIN_C7);
    }
  }
}else if(xstc == 3){
```

```
if(sts == 0){
  output_low(PIN_C0);
  output_low(PIN_C1);
  output_low(PIN_C3);
  output_low(PIN_C4);
  output_low(PIN_C6);
  output_low(PIN_C7);
}else if(sts == 1){
  if(lurd ==0){

    output_low(PIN_C0);
    output_low(PIN_C1);
    output_low(PIN_C3);
    output_low(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
    cmac = 0;
    ccp_1 = 0;
  }else if(lurd == 1){
    cmac = 1;
    ccp_1 = 64+(fs*36);
```

```
      output_low(PIN_C0);

      output_low(PIN_C1);

      output_low(PIN_C3);

      output_low(PIN_C4);

      output_high(PIN_C6);

      output_low(PIN_C7);

    }else if(lurd == 2){

      cmac = 1;

      ccp_1 = 64+(fs*36);

      output_low(PIN_C0);

      output_low(PIN_C1);

      output_low(PIN_C3);

      output_low(PIN_C4);

      output_low(PIN_C6);

      output_high(PIN_C7);

    }

  }

}

}

//main program

void main(){
```

```
int8 pin=0;            //pin is the value that will send to slave

 setup_timer_2(T2_DIV_BY_1,99,1);

 setup_ccp1 (ccp_pwm);

 while(1)

 {

   checkx();         //wait untill user select any digree exept X

   checkf();       //wait untill user select the speed

   checks();       // wait untill user select start or stop

   checkl();    //wait untill user select rotation

   moveit();


//////////////////////////////////////////////////////////////////////////////////////////

                         //

                         //  the program out put is M S1 S0 SP D2 D1 D0 ST

                         //

                         //  1- M master 0-manual 1- automated (0*128 = 0)

                         //  2- S1 S0 is Slave address 0- none 1- digree1 2-
digree2 3- digree3 (S1 S0)* 32 bit 5 and 6

                         //  3- SP is to select the speed of movement 0- slow 1-
fast (SP) * 16 bit 4

                         //  4- D2 D1 D0 is direction of movement 0- free whiling
1- left 2- up 3- right 4- down (D2 D1 D0)*2
```

```
//                                              bit
1,2,3
                    //  5- ST Start/ Stop bit 0- means fast stop 1- start (ST)
* 1 bit 0
                    //

//////////////////////////////////////////////////////////////////////////////////////
  }
}
```

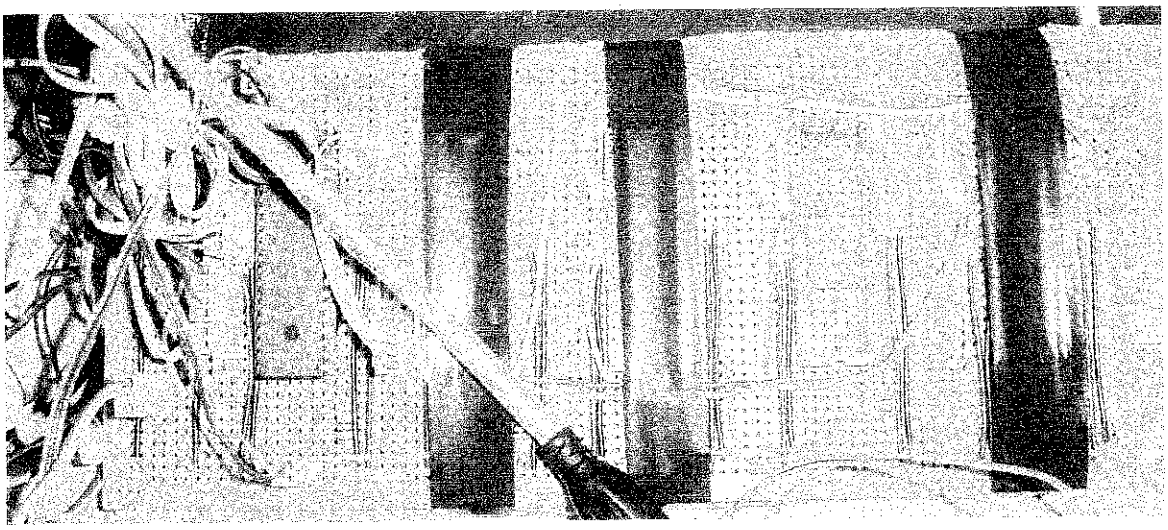


Figure 19 (manual controller)

After all we have arrived to the autonomous part of this robot which is made of one master micro controller along whith its slave. In this section the master micro controller will guide the encoders into its CCP_1 port which is performing the capture and compare job to finde the position of the robot and specificly the arm which is moving in that time. As we know we are having three encoders but only one CCP_1 port available. This problem could be solved by hiering at least three number of the micro controllers but redirecting the correct encoder through the relays will cost less than highering the greater number of micro controllers.

As the correct degree have been chosen by microcontroller the proper command which is being sent from computer into the micro contoller will be interperted and then accordingly the proper PWM will be sent to the slave micro controller and the slave wil start generating the requested PWM till it recive a new command from master microcontroller.

This part could be considered as a semiautonomous. Because a certaine series of commands can not be performed serially. The user needs to insert the joint number and also direction and the degree of movement for each part in order to move.

In order to make the robot fully autonomous we need to develop the inverse kinemethics algorithem either inside the autonumous mastaer or inside the computer. But as we were limited in term of time we left this part for the future development of robot.

In the following figures you can observe the circuitry of the autonomous part of this robot along with the MAX232 which has been used to establish a communication bridge between robot and computer. Also the selecting circuitry for the encoders is shown in the following figurs.

After all we have arrived to the point which makes this robot to be different from the existing versions in the market and industry. In this project we designed our robot in a way that it could be controlled via the internet and even wirellessly. Inorder to atchive this target we made a use of DATA AND COMPUTER NETWORKING abillities to fully controll the computer which is connected to the robot. as a result now our robot is able to be controlled either from the work station or through the internet. The importance of this application will be more observable when it is applied to a line of production which is a combination of few robots from this type while they are being controlled through a single main frame inside the controll room. Again I mention that this technology can be applied on the considerable number of this robots which are working togather on a production line by using the intenal network.

Another application of this technology will apear when the company which is producing this robot wants to trouble shoot the controll algorithem which is being run on the robotic production line in the user factory. This trouble shooting can be performed even from another continent. Like the robot producer factory is in germany and the user factory is located in malaysia. The trouble shooting could be performed real time with out making the engineers to travel to any place.

In the folowing paragraphs you can observe the program inside the autonomous master and its slave:

Aoutonomous master:

```
// computer will send digree\t direction\t angle

//this program will give pwm value to slave and the direction set by this program

// 1- digree also will select from here c0,c1 digree 1, c3,c4 digree 2, c6,c7 digree 3

// 2- direction 1  means c0+ digree (c3 digree 2,c5 digree 3) 2 will make other pin high

// 3- angle each palse from encoder is 0.36 digree palse = angle/0.36

//   if angle < 0.9* angle master-> slave send = 01 => pwm = 98 and

//   if angle > 0.9* angle master-> slave send = 10 => pwm = 75 and

//   if angle == angle master -> slave send = 00 => pwm = 0 and full stop

// when computer send this data is it set before set the pwm for slave the program send ack

//   and computer need to accept it


//every 24 digree is one second and 1/24 second = 1 digree so 100/1 = ?/0.04 ==> 4 =1 ? = 360

#include <16F877.h>
```

```
#use delay(clock=8000000)

#fuses NOWDT,HS,NOPUT,NOBROWNOUT,NOPROTECT,NOLVP

#USE RS232 (baud=9600,parity=N,xmit=PIN_D3,rcv=PIN_D2,stream=
o_stream,bits=8)

int sec, min, hour, millisec, millisecon;

//int overflow_count;

//unsigned int32 start_time, end_time;

//int32 pulse_ticks;

int32 cp = 0;

int8 con =0;

char dt[4];

int8 dtc =0;

int8 what;

int8 dig,dir,ang;

char com[10];

//int8 numd[5];

//int8 numc = 0;


#int_TIMER0

void TIMER0_isr(void)

{
```

```
set_timer0(6);

if (millisec++ == 51)

    {

   millisecon = millisecon+1;

   millisec=0;

    }


}


//#int_TIMER1

//TIMER1_isr()

//{

  //++overflow_count; //increment whenever an overflow occurs 16s

//}


 /*#int_CCP1

 CCP1_isr()

 {

 cp++;

 if(cp>3600000000)

   cp =0;
```

```
  //end_time = CCP_1;  //read captured timer ticks

  //check for pulse time (in ticks) accounting for anyoverflow that may have
occurred

  //pulse_ticks = ((0x10000 * overflow_count) - start_time + end_time)/200;

  //if ((cp/40) < 255){

    //output_d(cp/40); //light LED if too fast

    //output_low(PIN_C7);

  //}

  //else

  //output_high(PIN_C7); //extinguish LED if speed is OK


  //now set up for the next pulse

  //start_time = end_time;  //end time of this pulse is the start time for the next one

  //overflow_count = 0;  //clear overflow counts

}*/



char int2chr(int chr){

char ch[] = {'0','1','2','3','4','5','6','7','8','9'};

return ch[chr];

}
```

```
void showdt(){
  int8 i, val = 0;

  for(i=0;i<dtc;i++){
    val = val*10 + (dt[i]-48);

  }
  fputc(val , o_stream ) ; // number in ascii start from 48 =0
  switch(what++)
  {
    case 0:
      dig  = val;
      fputc('d' , o_stream ) ;
    break;
    case 1:
      dir  = val;
      fputc('i' , o_stream ) ;
    break;
    case 2:
      ang  = val;
```

```
        fputc('a' , o_stream ) ;
      break;
   }
}

void datag(char data){
if (data != '\t'){
   dt[dtc++] = data;
}else{
   //val =0;
   showdt();
   fputc('+' , o_stream ) ;
   dtc =0;
}
}

 void selectdig()
 {
   switch(dig){
```

```
case 0:
  output_low(PIN_C0);
  output_low(PIN_C1);
  output_low(PIN_C3);
  output_low(PIN_C4);
  output_low(PIN_C6);
  output_low(PIN_C7);
break;
case 1:
  output_high(PIN_A0);
  output_low(PIN_A1);
  output_low(PIN_A2);
  if(dir == 0){
    output_low(PIN_C0);
    output_low(PIN_C1);
    output_low(PIN_C3);
    output_low(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
  }else if(dir == 1){
    output_high(PIN_C0);
```

```
    output_low(PIN_C1);
    output_low(PIN_C3);
    output_low(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
  }else if(dir == 2){
    output_low(PIN_C0);
    output_high(PIN_C1);
    output_low(PIN_C3);
    output_low(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
  }
break;
case 2:
  output_high(PIN_A1);
  output_low(PIN_A0);
  output_low(PIN_A2);
  if(dir == 0){
    output_low(PIN_C0);
    output_low(PIN_C1);
```

```
    output_low(PIN_C3);
    output_low(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
  }else if(dir == 1){
    output_low(PIN_C0);
    output_low(PIN_C1);
    output_high(PIN_C3);
    output_low(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
  }else if(dir == 2){
    output_low(PIN_C0);
    output_low(PIN_C1);
    output_low(PIN_C3);
    output_high(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
  }
break;
case 3:
```

```
output_high(PIN_A2);
output_low(PIN_A1);
output_low(PIN_A0);
if(dir == 0){
  output_low(PIN_C0);
  output_low(PIN_C1);
  output_low(PIN_C3);
  output_low(PIN_C4);
  output_low(PIN_C6);
  output_low(PIN_C7);
}else if(dir == 1){
  output_low(PIN_C0);
  output_low(PIN_C1);
  output_low(PIN_C3);
  output_low(PIN_C4);
  output_high(PIN_C6);
  output_low(PIN_C7);
}else if(dir == 2){
  output_low(PIN_C0);
  output_low(PIN_C1);
  output_low(PIN_C3);
```

```
        output_low(PIN_C4);

        output_low(PIN_C6);

        output_high(PIN_C7);

      }

    break;

  }

}


void slave(){

//  while((cp*0.36)<(ang*0.9)){

  while(millisecon<(ang*0.9)){

    output_high(PIN_D0);

    output_low(PIN_D1);

  }

  while(millisecon<(ang)){

    output_low(PIN_D0);

    output_high(PIN_D1);

  }

    output_low(PIN_D0);

    output_low(PIN_D1);

}
```

```
int moveit(){
  if(what>1)
  {
    what = 0;
    selectdig();
    millisec = 0;
    millisecon =0;
    slave();
    cp=0;
    return 1;
  }else{
    return 0;
  }
}


void main()
{
  char ch;
  char j=0,i;
```

```
//lcd_init();
set_tris_b ( 0xff ) ;
//setup_ccp1(CCP_CAPTURE_RE);
//setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);//set timer1 to run at system
clock/4
//enable_interrupts(INT_TIMER1); //unmask Timer1 overflow interrupt
//enable_interrupts(INT_CCP1);    //unmask capture event interrupt
setup_counters(RTCC_INTERNAL,RTCC_DIV_8);
enable_interrupts(INT_TIMER0);
enable_interrupts(global);    //enable all unmasked interrupts
output_low(PIN_C0);
output_low(PIN_C1);
output_low(PIN_C3);
output_low(PIN_C4);
output_low(PIN_C6);
output_low(PIN_C7);
fputc('|' , o_stream ) ;
      while (1)
      {
    while(1){
    if((com[i++]=getc())=='\n')
```

```
  break;
}
i--;
fputc((i+48) , o_stream ) ;
what =0;
for(j=0;j<i;j++)
  datag(com[j]);
fputc( com[j], o_stream ) ;
i=0;
while(1){
  if(moveit() == 1){
    output_low(PIN_C0);
    output_low(PIN_C1);
    output_low(PIN_C3);
    output_low(PIN_C4);
    output_low(PIN_C6);
    output_low(PIN_C7);
    fputc('\n' , o_stream ) ;
    fputc('r' , o_stream ) ;
    fputc('o' , o_stream ) ;
    fputc('t' , o_stream ) ;
```

```
        fputc('a' , o_stream ) ;
        fputc('t' , o_stream ) ;
        fputc('i' , o_stream ) ;
        fputc('o' , o_stream ) ;
        fputc('n' , o_stream ) ;
        fputc(' ' , o_stream ) ;
        fputc('d' , o_stream ) ;
        fputc('o' , o_stream ) ;
        fputc('n' , o_stream ) ;
        fputc('e' , o_stream ) ;
        fputc('\n' , o_stream ) ;
        break;
      }
    }
  //numg(cp);
  //delay_ms(1000);
 }
}
```

Autonomous slave:

```
#include <16F877.h>
#use delay(clock=8000000)
```

```
#fuses NOWDT,HS,NOPUT,NOBROWNOUT,NOPROTECT,NOLVP

#use fixed_io(c_outputs = PIN_C2)


void main()

{

  int8 pd0=0;

  setup_timer_2(T2_DIV_BY_16,250,1);

  setup_ccp1(CCP_PWM);

  set_tris_d(0xFF);


  while(1)

  {

    if(pd0 != (input_d() & 0x03)){

    pd0 = input_d() & 0x03;

    if(pd0 == 0x00){

      ccp_1 = 0;

    }else if(pd0 == 0x01){

      CCP_1 = 98;

    }else if(pd0 == 0x02){

      CCP_1 = 75;

    }
```

```
    //CCP_1 = 98;

    }

  }

}
```

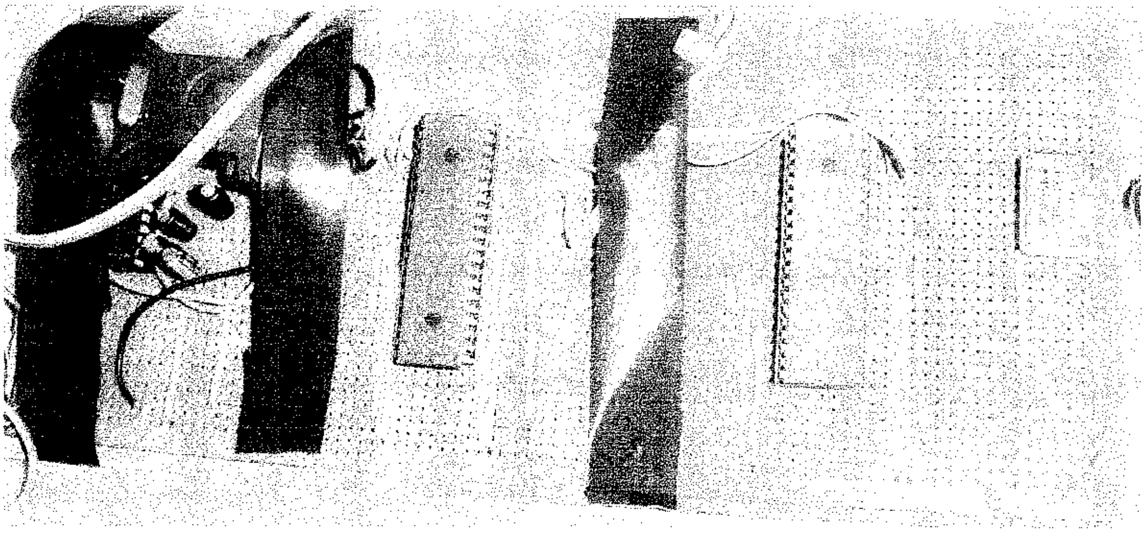


Figure 20 (autonomous controller)

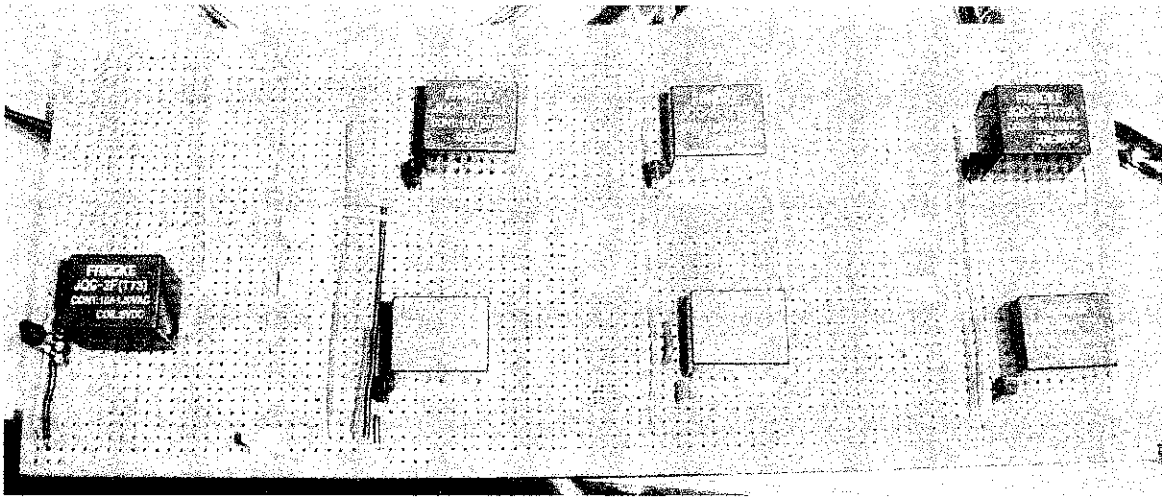


Figure 21 (relay circutry)

4.2 Mechanical aspects of the robot:

The mechanical part of this project was the main concern for the FYPI. Most of the parts of the project have been implemented and manufactured by now. All of these parts have the 3D Auto-Cad drawing which is shown in the following figures:

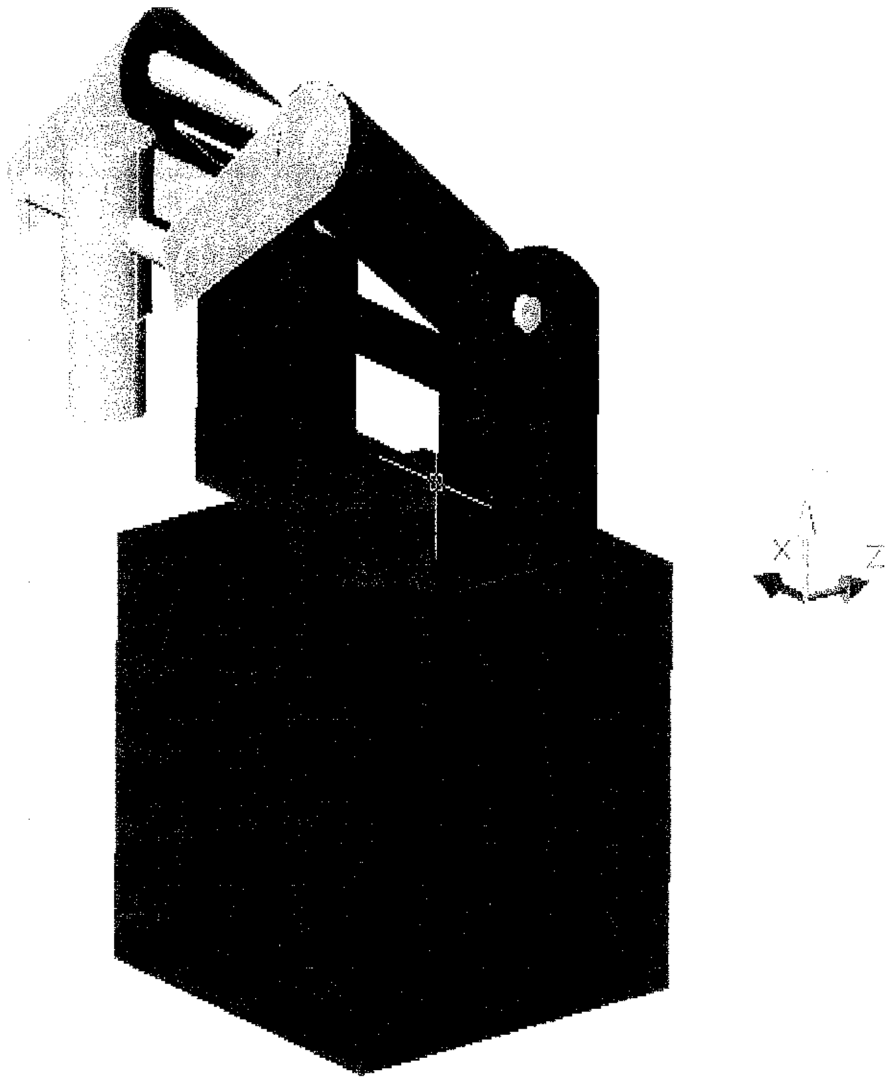


Figure 22 (3D Auto-Cad drawing1)

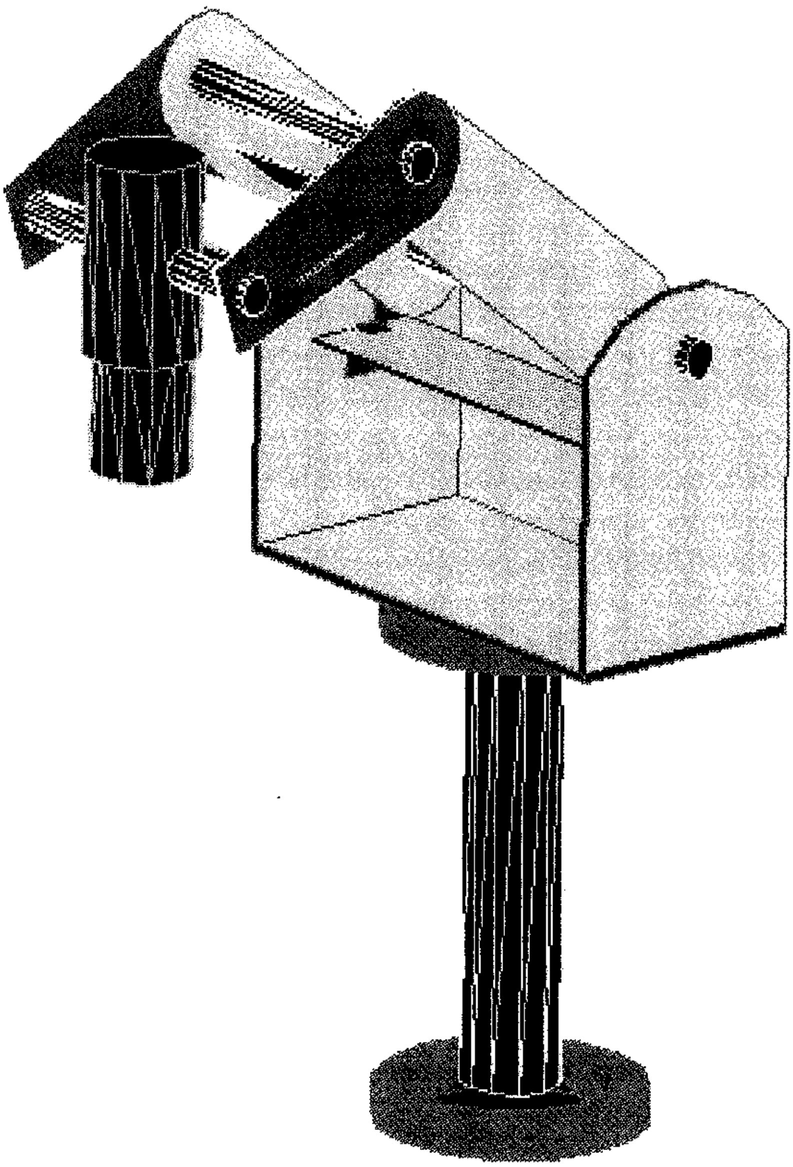


Figure 23 (3D Auto-Cad drawing2)

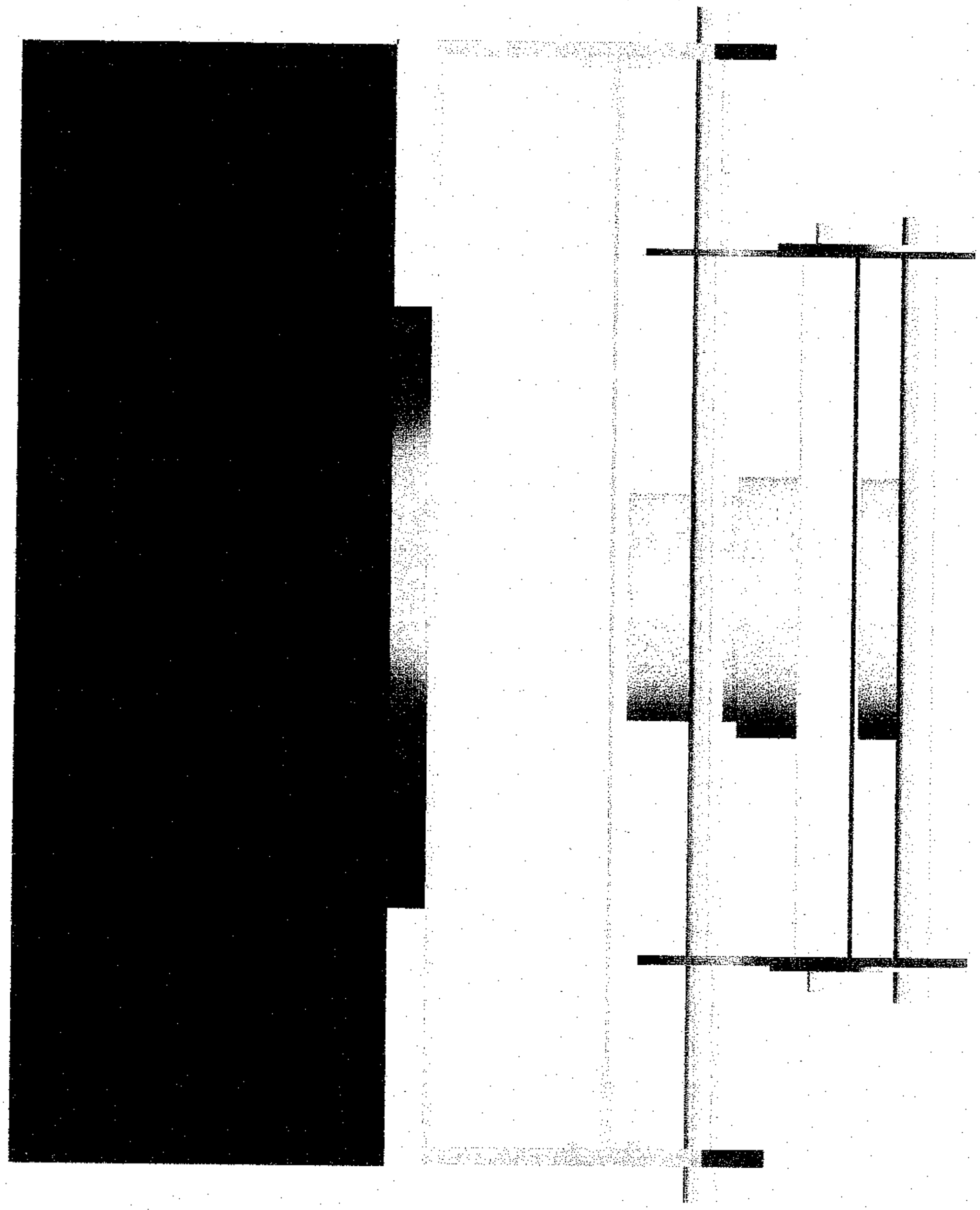


Figure 24 (3D Auto-Cad drawing3)

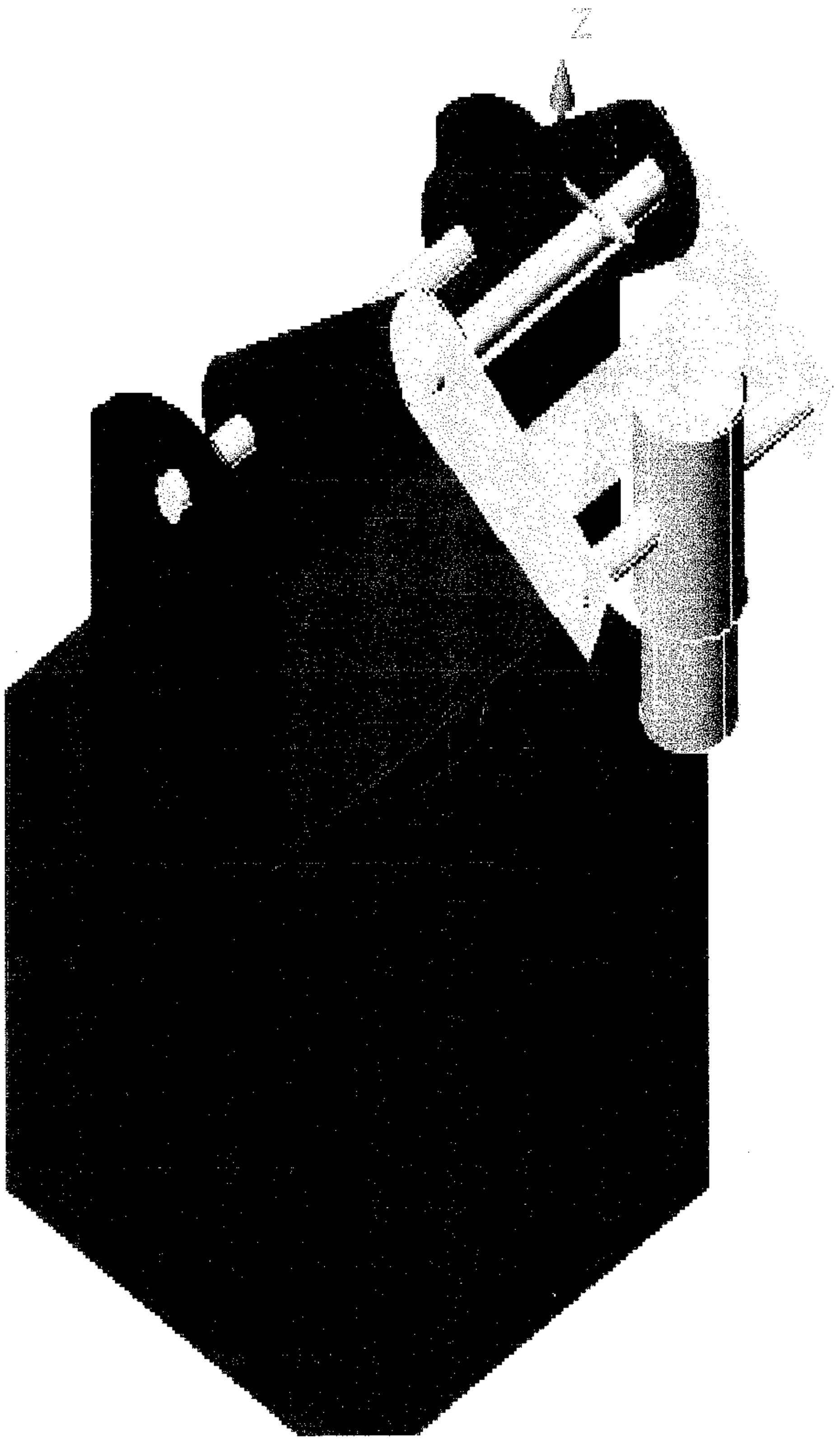


Figure 25 (3D Auto-Cad drawing4)

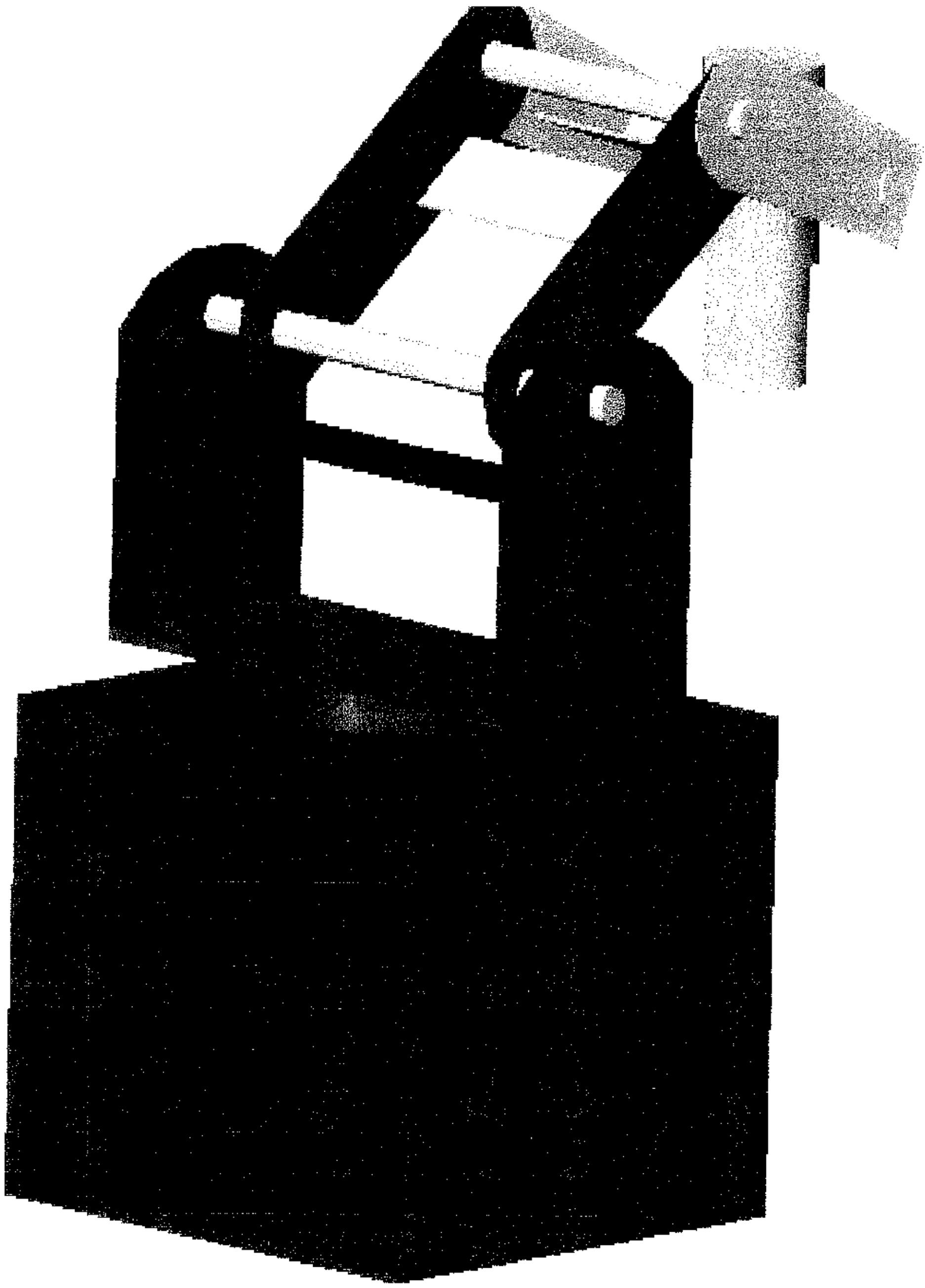


Figure 26 (3D Auto-Cad drawing5)

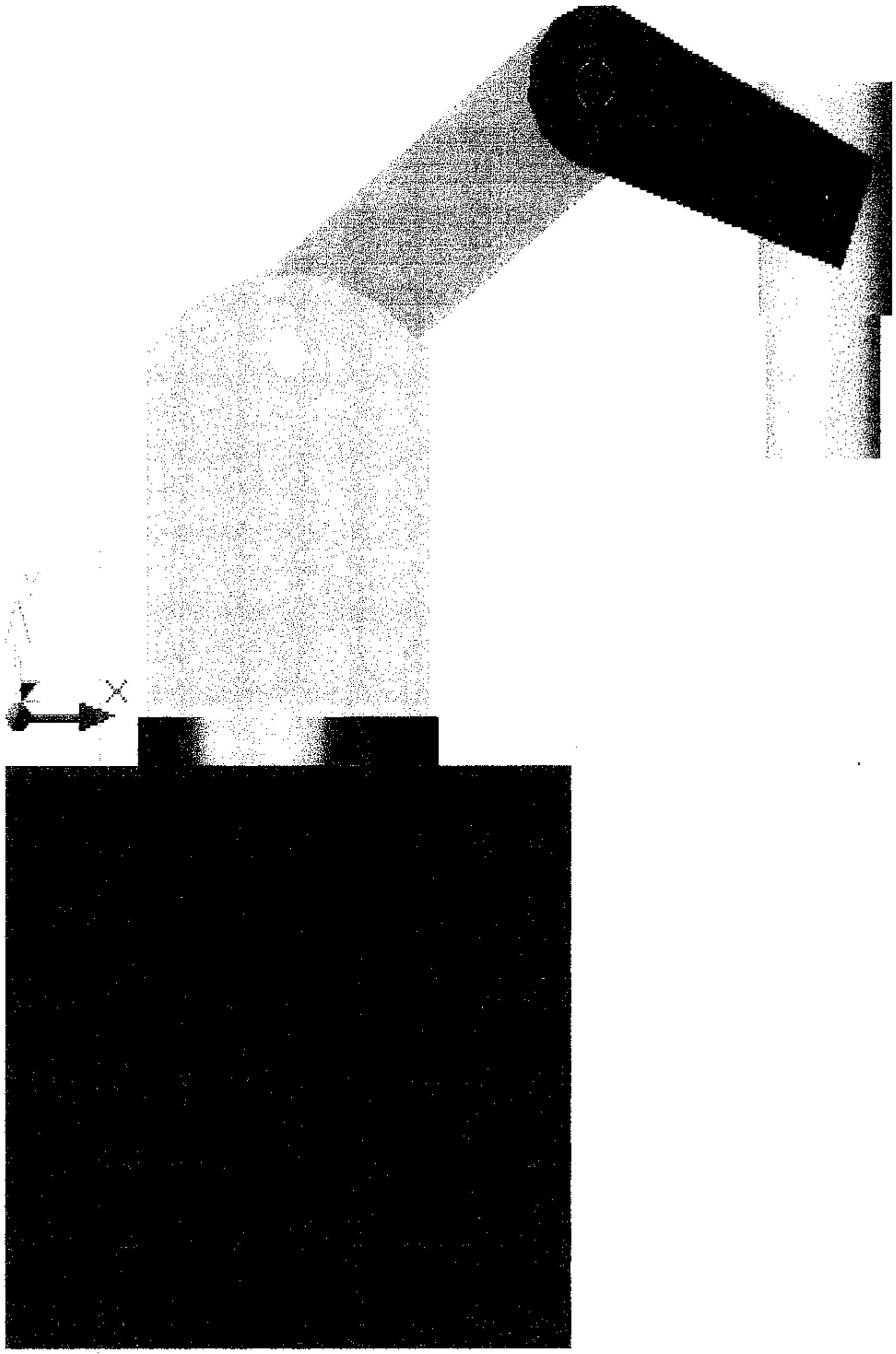


Figure 27 (3D Auto-Cad drawing6)

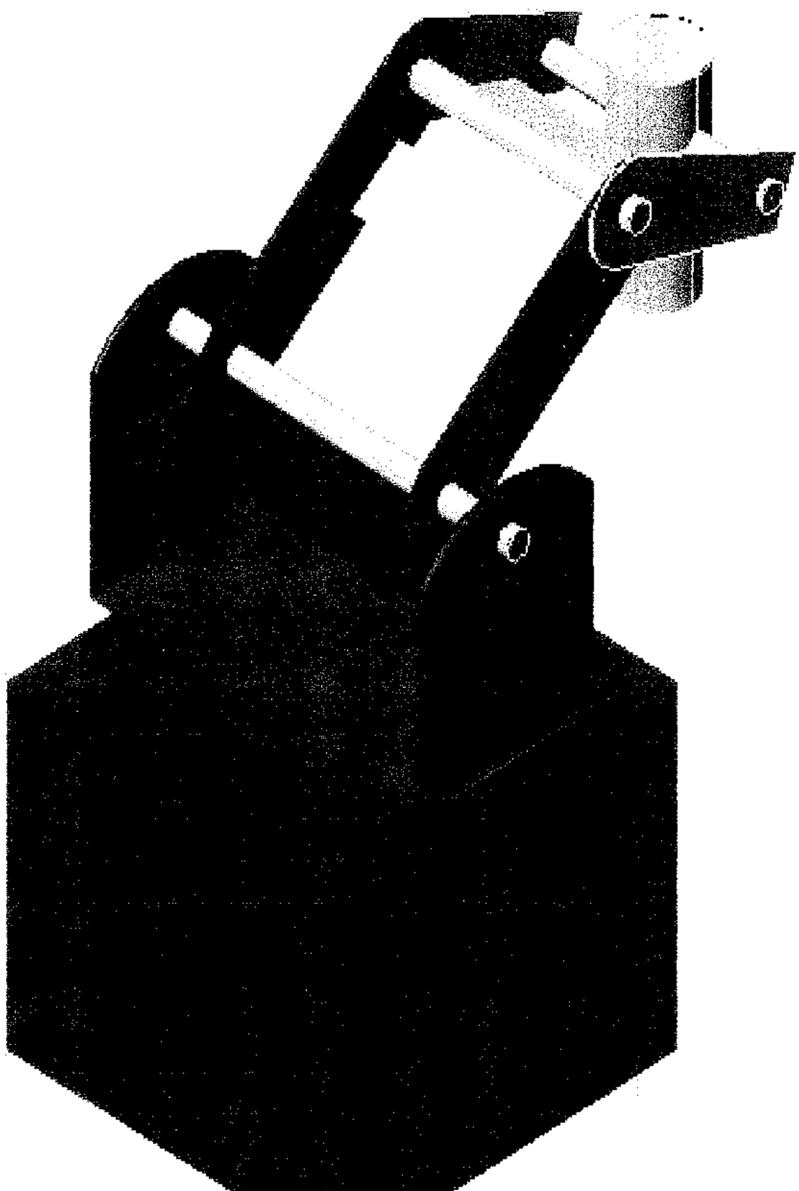


Figure 28 (3D Auto-Cad drawing7)

The following figures are the manufactured parts of the robot:

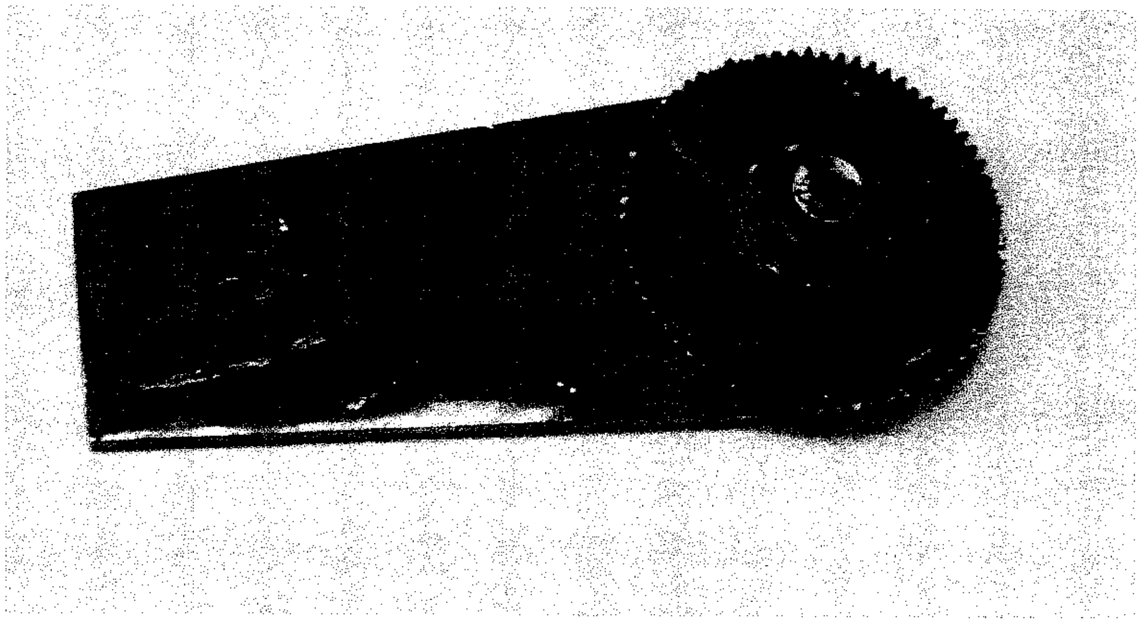


Figure 29 (work piece1)

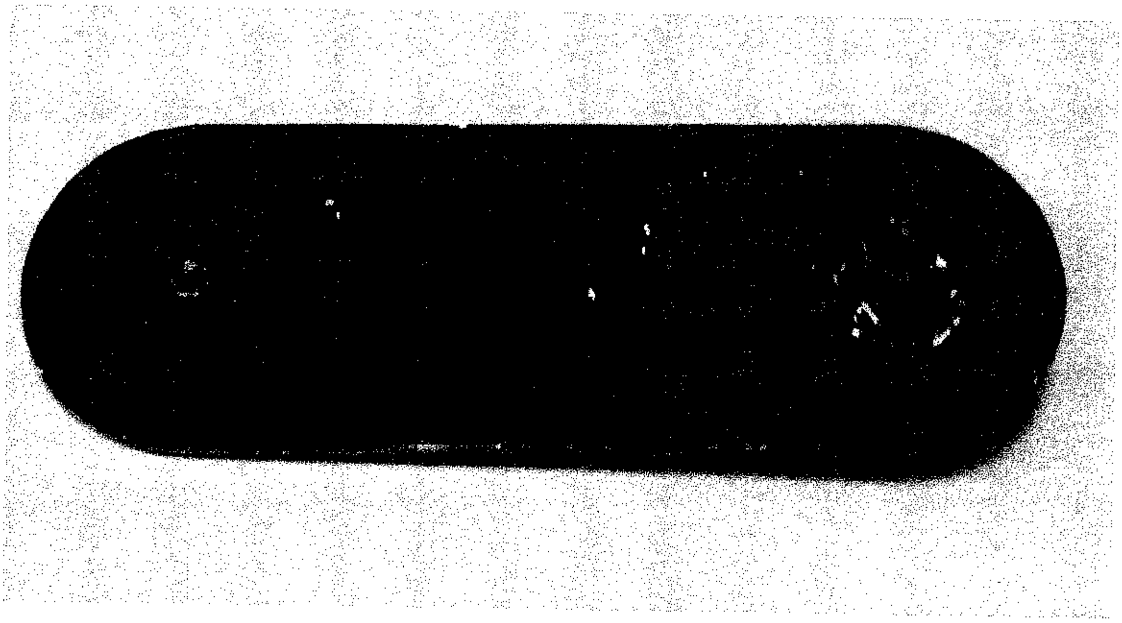


Figure 30 (work piece2)

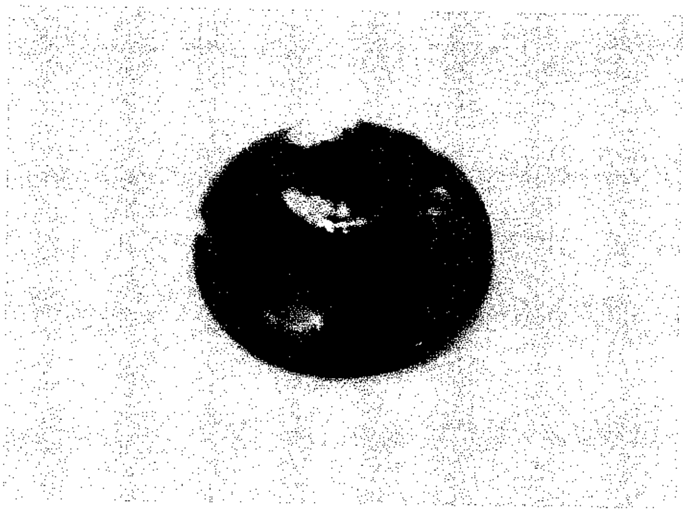


Figure 31 (work piece3)

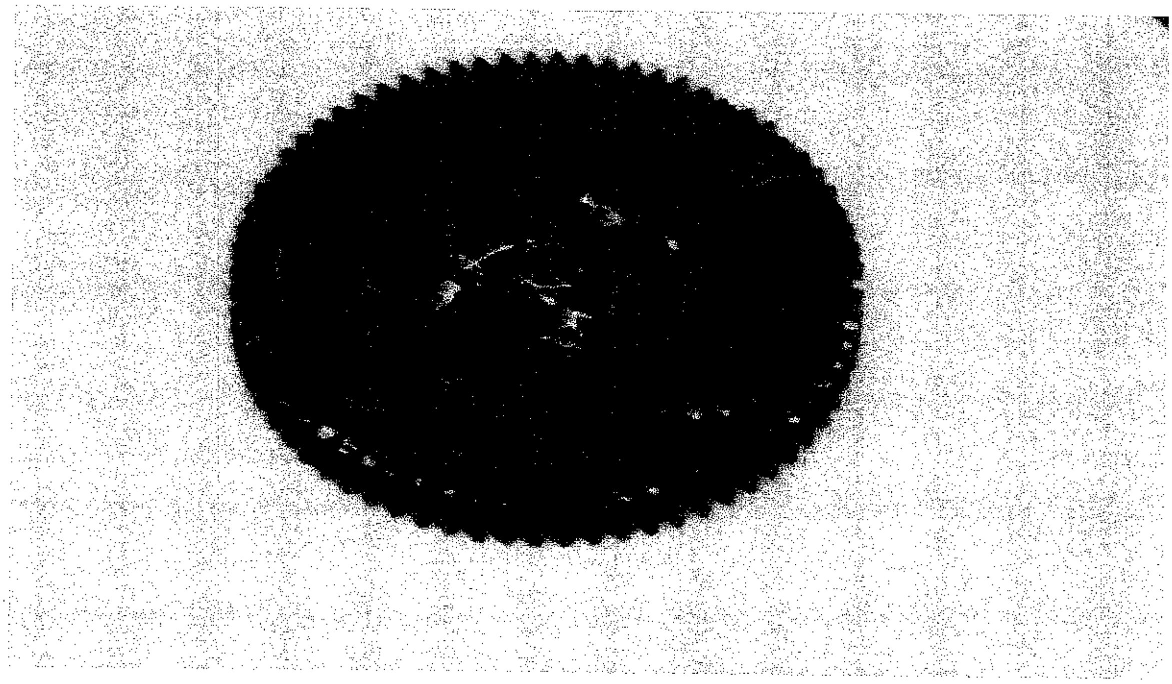


Figure 32 (work piece4)

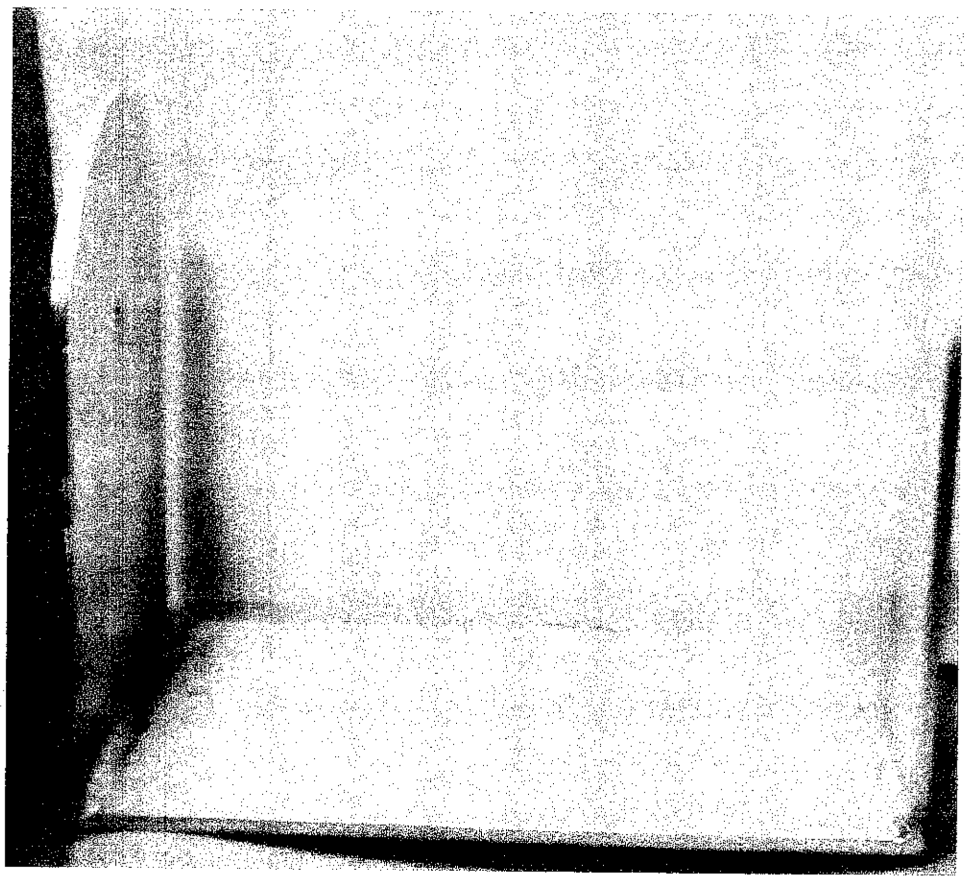


Figure 33 (work piece5)

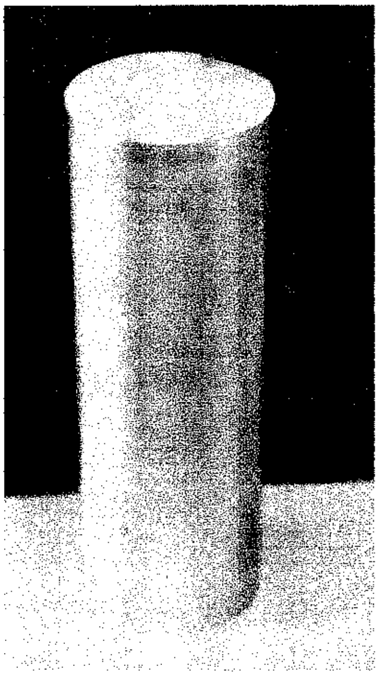


Figure 34 (work piece6)

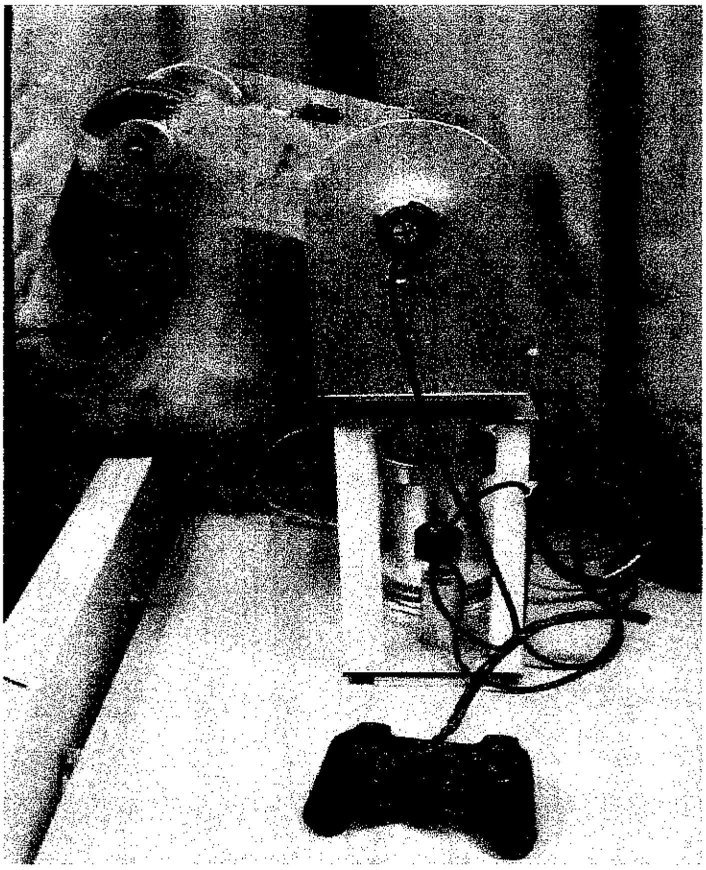


Figure 35 (robot view number one)

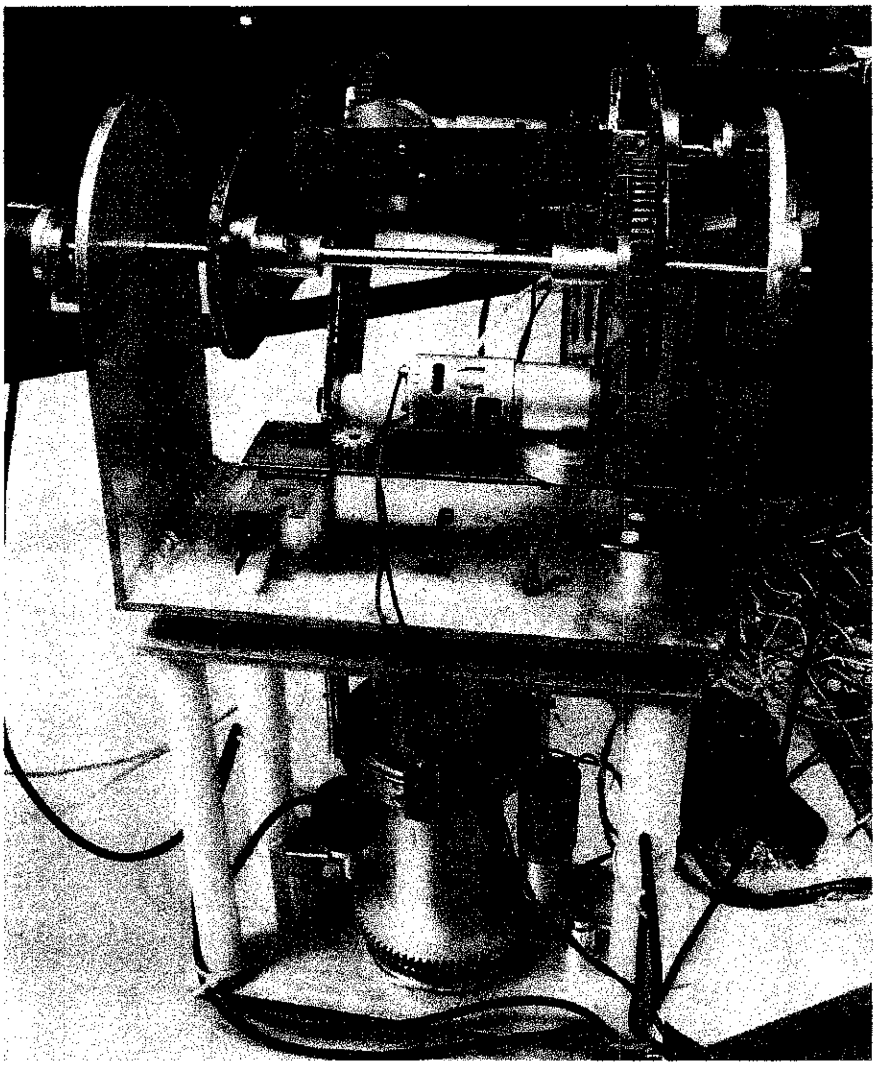


Figure 36 (robot view number two)

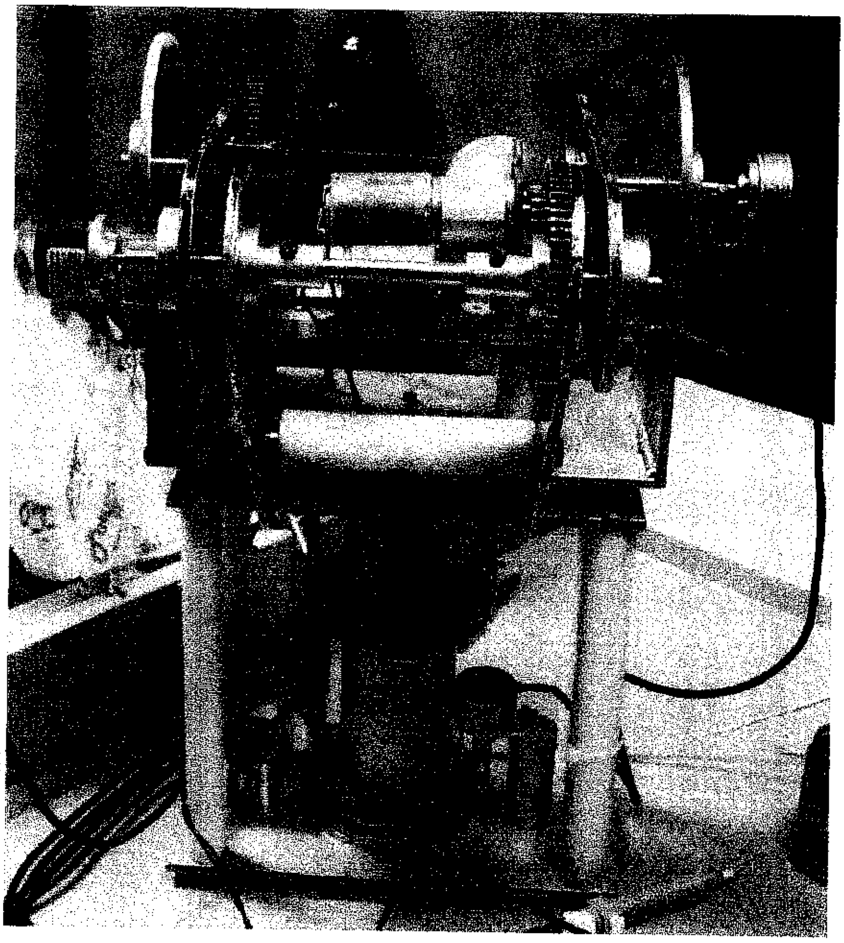


Figure 37 (robot view number three)

CHAPTER 5

CONCLUSION AND RECOMMENDATION

Conclusion and Recommendation

As we have seen the robot which is to be implemented in this project is having five degree of freedom and it is using its own power regulator for providing the electric power.

The way that we are using to control the speed of the DC motor is by use of PWM and H-Bridges to provide enough magnitude of current for the motor to run correctly and with desired speed. Microcontroller and computer will be used to control this robot and they will provide graphical user interface. Up to now too many steps toward manufacturing the robot have been taken which includes circuit design and microcontroller programs, mechanical manufacturing and so on. In fact choosing the right motors and sensor have been the main concern in the preliminary steps of this project as they will be playing a very important role in the future steps which will be taken in the FYPII project.

Recommendations:

As recommendation efficiency and accuracy of this robot could be enhanced by providing the ability of measuring the weight of the load which is to be treated.

Robot could be enhanced by adding the ability of image processing to the robot. This could simply accomplished by adding a camera and then a very simple image processing in two dimension to make the robot to recognize between round and square loads and then locate them in different boxes.

References

[1] Fundamentals of robotics David D. Ardayfio

[2] Building robot drive trains Dennis Clarckan Michael Owings

[3] Robot builder's BONANZA Gordon Mccomb

Appendix A

CNC codes:

```
(PROGRAM NAME - 2504073)

(DATE=DD-MM-YY - 25-04-07 TIME=HH:MM - 17:13)

N100G21

N102G0G17G40G49G80G90

(8. FLAT ENDMILL TOOL - 8 DIA. OFF. - 8 LEN. - 8 DIA. - 8.)

N104T8M6

N106G0G90G59X-150.Y79.A0.S2500M3

N108G43H8Z50.M8

N110Z10.

N112G1Z-.483F60.

N114X150.F500.

N116G2X154.Y75.R4.

N118G1Y-75.

N120G2X150.Y-79.R4.

N122G1X-150.

N124G2X-154.Y-75.R4.

N126G1Y75.

N128G2X-150.Y79.R4.
```

```
N130G1Z-.967F60.
N132X150.F500.
N134G2X154.Y75.R4.
N136G1Y-75.
N138G2X150.Y-79.R4.
N140G1X-150.
N142G2X-154.Y-75.R4.
N144G1Y75.
N146G2X-150.Y79.R4.
N148G1Z-1.45F60.
N150X150.F500.
N152G2X154.Y75.R4.
N154G1Y-75.
N156G2X150.Y-79.R4.
N158G1X-150.
N160G2X-154.Y-75.R4.
N162G1Y75.
N164G2X-150.Y79.R4.
N166G1Z-1.933F60.
N168X150.F500.
N170G2X154.Y75.R4.
N172G1Y-75.
```

```
N174G2X150.Y-79.R4.
N176G1X-150.
N178G2X-154.Y-75.R4.
N180G1Y75.
N182G2X-150.Y79.R4.
N184G1Z-2.417F60.
N186X150.F500.
N188G2X154.Y75.R4.
N190G1Y-75.
N192G2X150.Y-79.R4.
N194G1X-150.
N196G2X-154.Y-75.R4.
N198G1Y75.
N200G2X-150.Y79.R4.
N202G1Z-2.9F60.
N204X150.F500.
N206G2X154.Y75.R4.
N208G1Y-75.
N210G2X150.Y-79.R4.
N212G1X-150.
N214G2X-154.Y-75.R4.
N216G1Y75.
```

N218G2X-150.Y79.R4.

N220G1Z-3.383F60.

N222X150.F500.

N224G2X154.Y75.R4.

N226G1Y-75.

N228G2X150.Y-79.R4.

N230G1X-150.

N232G2X-154.Y-75.R4.

N234G1Y75.

N236G2X-150.Y79.R4.

N238G1Z-3.867F60.

N240X150.F500.

N242G2X154.Y75.R4.

N244G1Y-75.

N246G2X150.Y-79.R4.

N248G1X-150.

N250G2X-154.Y-75.R4.

N252G1Y75.

N254G2X-150.Y79.R4.

N256G1Z-4.35F60.

N258X150.F500.

N260G2X154.Y75.R4.

N262G1Y-75.

N264G2X150.Y-79.R4.

N266G1X-150.

N268G2X-154.Y-75.R4.

N270G1Y75.

N272G2X-150.Y79.R4.

N274G1Z-4.833F60.

N276X150.F500.

N278G2X154.Y75.R4.

N280G1Y-75.

N282G2X150.Y-79.R4.

N284G1X-150.

N286G2X-154.Y-75.R4.

N288G1Y75.

N290G2X-150.Y79.R4.

N292G1Z-5.317F60.

N294X150.F500.

N296G2X154.Y75.R4.

N298G1Y-75.

N300G2X150.Y-79.R4.

N302G1X-150.

N304G2X-154.Y-75.R4.

```
N306G1Y75.

N308G2X-150.Y79.R4.

N310G1Z-5.8F60.

N312X150.F500.

N314G2X154.Y75.R4.

N316G1Y-75.

N318G2X150.Y-79.R4.

N320G1X-150.

N322G2X-154.Y-75.R4.

N324G1Y75.

N326G2X-150.Y79.R4.

N328G0Z50.

N330M5

N332G91G28Z0.M9

N334G28X0.Y0.A0.
```