# UNIVERSITI MALAYSIA PAHANG

## BORANG PENGESAHAN STATUS TESIS◆

**JUDUL: DESIGN AND DEVELOPMENT AUTOMATED GRIPPER CONTROL FOR ROBOTIC ARM**

**SESI PENGAJIAN: 2012/2013**

Saya, **ONG RONG SAN (880813-10-5087 )**

mengaku membenarkan tesis (Sarjana Muda / ~~Sarjana / Doktor Falsafah~~)* ini disimpan di perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis ini adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

| | | |
|---|---|---|
| ☐ | **SULIT** | (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972) |
| ☐ | **TERHAD** | (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi / badan di mana penyelidikan dijalankan) |
| √ | **TIDAK TERHAD** | |

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

**No 5, Jalan Kiambang 10,**
**Taman Gembira**
**42700 Banting**
**Selangor, Malaysia**

**EN KHAIRUL FIKRI BIN MUHAMAD**
(Nama Penyelia)

Tarikh: **19 JUNE 2012**

Tarikh: **19 JUNE 2012**

CATATAN: * Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.

◆ Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

# DESIGN AND DEVELOPMENT AUTOMATED GRIPPER CONTROL FOR ROBOTIC ARM

ONG RONG SAN

Thesis submitted in fulfillment of the requirements
for the award of the degree of Bachelor of Mechatronic Engineering

Faculty of Manufacturing Engineering
UNIVERSITI MALAYSIA PAHANG

JUNE 2013

**UNIVERSITI MALAYSIA PAHANG**

**FACULTY OF MANUFACTURING ENGINEERING**

I certify that the project entitled "Design and Development Automated Gripper Control for Robotic Arm" is written by Ong Rong San. I have examined the final copy of this project and in my opinion; it is fully adequate in terms of scope and quality for the award of the degree of Bachelor of Engineering. I herewith recommend that it be accepted in partial fulfillment of the requirements for the degree of Bachelor of Mechatronic Engineering.

MR.ISMAIL BIN MOHD KHAIRUDDIN

Examiner                                        Signature

## SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality of this thesis is qualified for the award of the Bachelor of Mechatronic Engineering.

Signature :
Name : MR. KHAIRUL FIKRI BIN MUHAMAD
Position : LECTURER OF FACULTY OF MANUFACTURING
Date : 19th JUNE 2012

## STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature :
Name : ONG RONG SAN
MATRIC ID : FB10056
Date : 19th JUNE 2012

## DEDICATION

*I specially dedicate to my beloved parents*

*and those who have guided*

*and motivated me for this project*

*To my supervisor En.Khairul Fikri Bin Mohamad*

*and all my friend whose*

*give their support and encouragement*

**ACKNOWLEDGEMENT**

First and foremost, the deepest sense of gratitude to my parent , who guide and gave me the strength and support to complete this final year project 1 successfully. I would like to express my sincere gratitude to my supervisor Mr. Khairul Fikri Bin Muhammad for his continuous guidance, support and encouragement, which gave me huge inspiration in accomplishing this research. His practice of professional ethics and conducts which encourages me to become a confident and competent person to work individually as well as in groups with minimal supervision.

Besides that, I also took this opportunity to thank faculty staff who helped me in preparing the equipments. Not forgot to thank the panels who critics the outcome of this research besides providing some suggestions to improve the discussion and conclusion as well. I would also like to express my deepest appreciation to my parents whom always support and motivate me to complete this research and thesis. I also owe a depth of gratitude to my university friends who shared their knowledge and ideas that lead to the completion of this thesis.

Finally, to individuals who has involved neither directly nor indirectly in succession of this research with thesis writing. Indeed I could never adequately express my indebtedness to all of them. Hope all of them stay continue support me and give confidence in my efforts in the future. Thank you.

## ABSTRACT

The gripper is considered an important part of a robot, it has been wisely used in all kinds of all industries to increase the efficiency. Commonly it is designed as a mechanical hand by using the mechanical force in the automation system. It only can manipulate to the similar object, size and shape, because sequence motion is already compiled by a programmer. This makes user hard to manipulate and only can control the gripper manually. Thus to make the gripper can do the work in different conditions, the sensor and control systems are required for development automated gripper. The pressure force and infrared distance sensor use to measure the force and distance, with this device the gripper can contact with the work-piece without deformation. The signal from the sensor will transmit to the computer control by using a close loop system. This type of control system would able to analysis the data for control the position of the gripper. To prevent the object slip down from the gripper, and to grasp various types of object. In addition, the control system would able to control well in gripping force, the gripping force is a contact force per unit gripping are to clamp the object. This can gripper avoid the deformation of the work-piece because of the unconstant gripper force. Visual Basic (VB) programming language will used in the computer control system to control the overall movement of the gripper, because the VB programming ability to show the graphical user interface (GUI), to make the user more easy to control the sequence motion of the gripper. Besides the user also can control the gripper in a short period without using a complex formula to calculate the gripping force. Therefore data were analysed by the computer control system to control the gripping force and motion of the gripper. Through the programming abilities to show out the diagram about the position sensor, thus users can know the overall the condition of the gripper. Contribution gripper can apply to real life not only in the industry, it also can help people reduce the burden in daily life. Therefore the fundamental to design the gripper is important, it can show out the visualize force by using CATIA Finite Element Method (FEM) to the built the structure of the gripper. This can reduce the critical point in a particular area, and improve the reliability and endurance of the gripper. This all aspect must take into consideration to develop the automated gripper, these designs were fulfilled with the basic requirement believe in the future it able to perform multitasking for the robotic arm.

# ABSTRAK

Penggenggam dianggap sebagai bahagian penting dalam robot, ia telah digunakan dengan luasnya dalam semua jenis semua industri untuk meningkatkan kecekapan. Biasanya ia direka sebagai tangan mekanikal dengan menggunakan kuasa mekanikal dalam sistem automasi. Ia hanya boleh memanipulasi untuk object yang sama, saiz dan bentuk, kerana gerakan urutan sudah disusun oleh jure pengaturcara. Ini menjadikan pengguna sukar untuk memanipulasi dan hanya boleh mengawal penggenggam secara manual. Oleh itu untuk membuat penggenggam yang boleh melakukan kerja dalam keadaan yang berbeza, sensor dan sistem kawalan yang diperlukan untuk mencipta penggenggam automatik. Daya tekanan dan sensor jarak cahaya infra merah digunakan untuk mengukur daya dan jarak, dengan peranti ini penggenggam boleh mengambil barang kerja tanpa perubahan bentuk. Isyarat daripada sensor akan menghantar kepada kawalan komputer dengan menggunakan sistem gelung rapat. Ini jenis sistem kawalan akan dapat analisis data untuk mengawal kedudukan penggenggam, bagi tujuan untuk mengelakkan tergelincir daripada penggenggam, dan untuk memegang pelbagai jenis objek. Di samping itu, sistem kawalan juga dapat mengawal dengan baik dalam kuasa menggenggam, daya menggenggam adalah satu daya sentuhan seunit menggenggam untuk mengetatkan kawalan objek. Ini menjadi penggenggam dapat mengelakkan perubahan bentuk kerja keeping, disebabkan kuasa penggenggam yang tidak tetap. Visual Basic (VB) bahasa pengaturcaraan yang akan digunakan dalam sistem kawalan komputer untuk mengawal pergerakan keseluruhan penggenggam, kerana pengaturcaraan VB yang dapat menunjukkan maklumat kepada pengguna grafik (GUI), untuk membuat pengguna lebih mudah untuk mengawal pergerakan turutan penggenggam. Selain itu pengguna juga boleh mengawal penggenggam dalam tempoh yang singkat tanpa menggunakan formula yang kompleks untuk mengira tenaga mencengkam. Oleh itu data dapat dianalisis oleh sistem kawalan komputer untuk mengawal kuasa menggenggam dan gerakan penggenggam. Melalui pengaturcaraan boleh menunjukkan keluar gambarajah mengenai kedudukan sensor, oleh itu pengguna boleh tahu keadaan keseluruhan yang penggenggam. Sumbangan penggenggam boleh memohon digunakan dalam kehidupan sebenar bukan sahaja dalam industri, ia juga boleh membantu ramai orang untuk mengurangkan beban dalam kehidupan seharian. Oleh itu, asas untuk mereka bentuk penggenggam adalah penting, ia boleh menunjukkan keluar daya yang menggambarkan dengan menggunakan CATIA Finite Element Method (FEM) bagi membina struktur penggenggam. Ini boleh mengurangkan titik kritikal dalam bidang tertentu, dan meningkatkan kebolehpercayaan dan ketahanan penggenggam. Aspek semua mesti mengambil kira untuk membangunkan penggenggam automatik, reka bentuk ini telah dipenuhi dengan keperluan asas percaya pada masa akan datang ia mampu untuk melakukan pelbagai kerja untuk lengan robot.

## TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

## LIST OF SYMBOLS

| | |
|---|---|
| Fs(N) | Shear force |
| Fn(N) | Normal force |
| µ | Coefficient of the friction at the fingertip contact |
| Ibs | Force |
| Vout | Voltage Output |

## LIST OF ABBREVIATIONS

| | |
|---|---|
| ABS | Acrylonitrile Butadiene Styrene |
| ADC | Analog to Digital |
| ARM | Assistive Robotic Manipulator |
| CADCAM | Computer Aided Design Computer Aided Manufacturing |
| CATIA | Computer Aided Three Dimensional Interactive Application |
| COM | Component Object Model |
| DC | Direct Current |
| DMU | Digital Mockup Kinematics |
| EMF | Electromotive Force |
| FEM | Finite Element Method |
| GUI | Graphical User Interface |
| ICSP | In Circuit Serial Programming |
| LCD | Liquid Crystal Display |
| LED | Light Emitting Display |
| MCU | Microporgrammed Control Unit |
| PC | Personal Computer |
| PIC | Programmable Integrated Circuit |
| PID | Proportional Integral Derivativew |
| PTF | Polymer Thick Film |
| RPM | Revolutions Per Minute |
| RX | Receive |
| TX | Transmit |
| USB | Universal Series Bus |
| VB | Visual Basic |
| VOLT | Voltage |

## LIST OF APPENDICES

# CHAPTER 1

# INTRODUCTION

## 1.1 INTRODUCTION

Nowadays robot becomes an important machine due to its increasing use in industry. To improve work efficiency of the robot, gripper has been designed as a mechanical hand and integrated with the robot. It is used to grasp and place a material, tool or part from one site to another site.

The gripper becomes an essential part of a robot and it is wisely used in the automation field. Today, two-jawed gripper more and more frequently used in assembly tasks. It is totally different compared to gripper with a vacuum-based electromagnetic and gripper with electromagnets. Two-jawed gripper required mechanical force and gripper application to move the jaws to grasp an object.

Generally, the industrial robot gripper is a specialized device. It can only used to grip the object which similar in size, shape and weight of the repeated assembly process. In case, to grasp the objects which are different in size, shape and weight, manpower is needed to control the industrial robot gripper manually.

Therefore, sensors and control systems are play a major role in improving the automated gripper. The sensor will attach at the gripper to detect the position of objects to pick up and the environment, the data which detected by the sensor will send to the computer. While the computer control system of the gripper will analysis the data and adjust its subsequent action according to the actual state.

Gripping force is a contact force per unit gripping area to clamp the object. There are various kinds of properties object such as thin wall, soft and fragile. Different type of object required different degree of gripping force to clamp. The computer control system would able to find the suitable gripping force to hold the objects and avoid deformation.

The contribution of gripper shall not only restricted to the industries, likewise the use of gripper shall broaden to the daily life of the household. Thus, to design a proper gripper is important. Other than multitasking ability, criteria such as simplicity, user-friendly, durability, reliability, low cost and low maintenance shall take consider in designing a proper gripper

## 1.2 PROBLEM STATEMENT

a) Manual control needs experience and time consuming.
b) Large scale robot not suitable for portable tasking.
c) Complex calculation gripper force to grasp the object.

## 1.3 OBJECTIVE OF STUDY

a) To implement the control system in the motion of gripper.
b) To integrate control system with sensor.
c) To design and build a program for automated gripper.
d) To implement the computer control for the gripper.

## 1.4 PROJECT SCOPE

This study is focused on four project scope

a) To study the design of mechanical systems and driving mechanisms of the gripper.
b) Sensors attached to the gripper, sensors will send data through the electrical circuit and a parallel port to the computer.
c) Implement a computer control system for data analysis.

d) Data were analysed by the computer control system will control the gripping force and motion of the gripper.
e) Gripper performance based on the type and thickness of the object.

## 1.5 ORAGANIZATION OF THESIS

This thesis consists of five chapters. Chapter 1 presents an introduction. While Chapter 2 highlights the literature review of the article, journal and etc. Chapter 3 explains the methodology for the project. Chapter 4 shows the result of the experiment and the discussion regarding to the results. Finally, Chapter 5 concludes the project and provides recommendations for the project

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 INTRODUCTION

In this chapter, previous studies relevant to this project will be presented. The gripper is an important part of a robot, it is wisely used in all types of industries to reduce human burden and replace human beings to do dangerous task. In order to increase the efficiency of work, the design of gripper must take consider in all aspects to produce a functional and cost effective gripper.

There are various designs on gripper available at the current market such as hydraulic, pneumatic, and electric actuation. The design of gripper has now been more concern on safety and reliability (Redwan Alqasemi, 2007). More and more grippers are using two fingers in grasping objects, but they are still subject to lots of weaknesses.

The main objective of this project is to produce a gripper which has high dexterity and can assist households in their various daily living activities. This gripper is hoped to improve performance and usability. The gripping force of the gripper is set to about 45N (10 pound) of force (Rajiv Dubey, 2007).

## 2.2 FUNDAMENTAL OF GRIPPER

There are several types of gripping system available on the market. Gripping system is used the mechanical contact forces to grasp the object. According to Deaconescu (2011), it can be classified by the gripping modality of objects such as clamping, vacuum, suspension, adhesion, electromagnetically and etc. In addition, each type of gripper has a different number of gripping points which makes the gripper can achieve singular or multiple contact. This gripping system can be powered by electricity, pneumatics, hydraulics or combined.

The Gripper can replace the human hands, therefore gripper design becomes more and more advanced. Nowadays in the market is not only selling the two-jawed gripper, it has been developed to configuration with three jaws, four fingers, and five fingers. Gripper with multi-finger configuration able to grasp different geometry and different grasping point. This design is suitable for the assembly of automotive body parts (Deaconescu, 2011).  Assistive Robotic Manipulator (ARM) or known as Manus also has been redesigned by GertWillem et al to include high efficiency and high adjustable of driving mechanism into the gripper.

Most robots mounted two-jawed gripping system to perform the task. Two jaw grippers are widely used in the robot arm, because of the simplicity of configuration and easy to program. But the point of loading of two-jawed gripper must consider too. Point loading is the contact force per unit gripping area which results during the clamping. Imposed the point loading by gripper can damage the workpiece such as thin-walled, brittle, polished, super-finished or softly. These works-pieces are not able to withstand the gripping force and would be exposed to damage and deformation (Deaconescu, 2011).

## 2.3 DESIGN PROBLEM

A fundamental problem of design and operating mechanism of the gripper is important and shall be identified. This is to ensure that gripper can achieve full performance in all aspects of the gripper (Yeung et al, 2003). Design problem can be proposed such as the following steps:

a) Determine the basic characters of gripper, it consists of the dimension, shape, weight, surface contact and the material used to produce.
b) Evaluate the maximum grasping force required to grasp an object.
c) Implement the sensor to the jaws.

d) Implement the control system to the gripper.
e) Design the mechanism of the gripper.
f) Decide to use the DC motor to control the motion of the gripper.
g) The design cost, production cost, robotic application cost and maintenance cost.
h) Flexibility of the gripper, whether the gripper allows rapid replacement, is easy adjust or modify.

## 2.4 THE ACTUATOR FOR GRIPPER

Gripper normally consists of two jaws and it needs power for actuation. Figure 2.1 is a two-jawed gripper which can open and close to pick and place a range of objects. There are many kinds of gripper actuator such as hydraulic, electrical, pneumatic or combined, by deploying linear or rotating motors. Every actuator has its own its advantage and disadvantages.

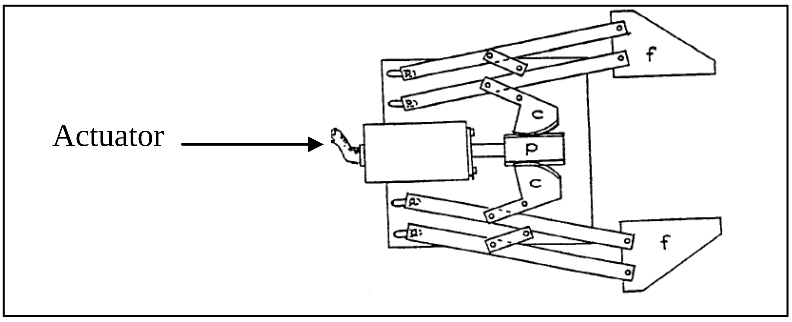



**Figure 2.1:** A scheme of mechanical design for two-finger grippers
Source: Andrea Deaconescu,(2011)

Pneumatic gripper is popular on the market now, because its compactness of the driving elements makes it easy to apply, low costs and easy maintenance. However, it is still subjected to control and compliance related problem.

The combination actuation will become electro-pneumatic or electro-hydraulic. The advantage of electro-pneumatic is rapid displacement and it is the most safety clamping systems. But its disadvantage is difficult to control the compliance problems related to driving position. (Deaconescu, 2011)

The electrical actuation is chosen as the actuator for gripper system, that is because it is using the computer programming to control. The electrical actuation consists the DC motors, stepper motor and servo motor. The servo motor and stepper motor are suitable to control the motion of gripper, both are received pulse of the movement. The stepper motor can control the movement more accurate and precise compare with servo motor, but need a lot of signal channels to input. Instead the servo motor only requires one channel signal to input to rotate, but the stepper motor can provide more accurate and precise movement compared to the servo motor.

## 2.5 THE MECHANISM AND CONTACT POINT OF GRIPPER

As stated earlier the mechanism of the gripper is to allow the jaw to close and open in parallel motion. The parallel motion can increase the contact surface to handle or grasp the object, in this way it can prevent object slipping out from the gripper. It has a surface contact angle that can be controlled, the efficiency of this mechanism will reduce the risk of objects being dropped. From Figure 2.2 on the next page is describing the humans using fingers to execute force control and fine motion act like as gripper. While the full arm act as robot arm to use gross motion control.

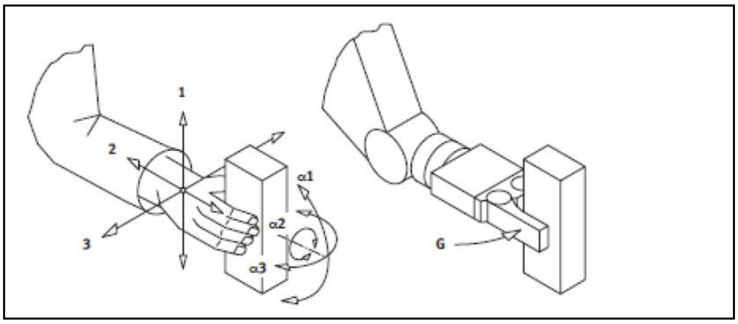



**Figure 2.2:** Motions to be conducted by a gripping system

Source: Salisbury, (1982)

The investigation of Salisbury (1982) stated that when grasp an object, the fingertip force is controlled, and it is likely to the contact forces. The contact forces inside the friction cone at the point of contact as shown below (Figure 2.3).

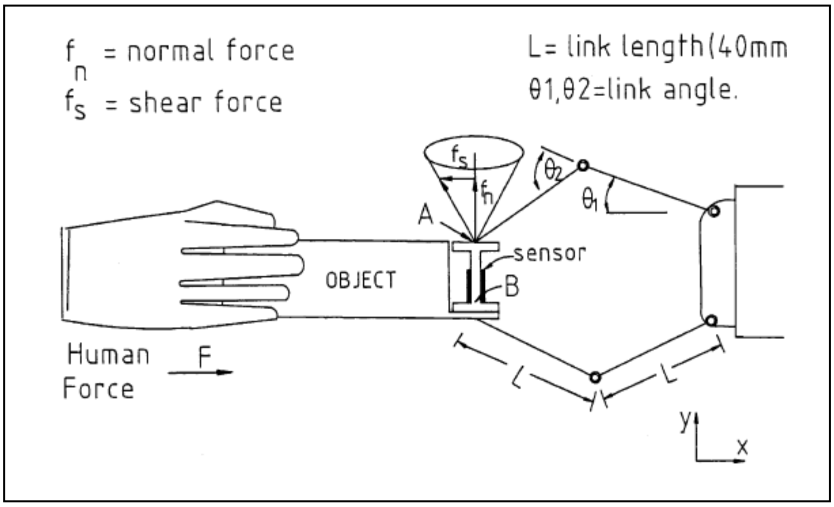



**Figure 2.3:** Forces acting on the object and fingers

Source: Salisbury, (1982)

As the object is symmetric and only two fingers are used for grasping, the forces on each finger are given by the following two equations.

The equation for $f_s$ (N):

$$f_s = \frac{F}{2} \tag{2.1}$$

The equation for $f_n$ (N):

$$f_n = \frac{F}{2\mu} \tag{2.2}$$

Where,

$f_s$ (N) = The shear force

$f_n$ (N) = The normal force

$\mu$ = Coefficient of the friction at the fingertip contact.

The contact between the finger and the object is assumed to be of the point contact type with friction. Refer to Figure 2.3 that the point of contact of the finger on the object was at the point ‘‘A’’. This is very important so that the normal finger force does not produce any bending moment on the sensory elements. In order to prevent the object from slipping, humans control the normal force, such as the shear force. It is inside the friction cone at the contact point (Ashish Dutta, 2001). As can be seen from Equation 2.2 the normal force required to prevent object slip should be greater than 1=ð2mÞ times the force applied by the human (F).

At the fingertip contact point, it functions as a normal force to prevent slipping the object. While the shear force is to estimate the slip and friction force. The importance of shear force is to control normal force and the orientation of the fingers to control the movement. However, on today's market still no have a sensor can estimate the shear force (Goro Obinata, 2001).

## 2.6 SENSOR

The market has many types of sensor, but shear force sensors still not available in the market. To solve the problem, the gripper can roughly divide into several conditions. While the gripper grasp the object is under stable condition, the control system for gripper can simplify to indirectly senses. The shear force on the fingertips required to calculate to know the position movement gripper. The gripping force can control more accurate and precise to grasp the object.

The shear force at the fingertips is considered one direction that means in the straight lines. The finger tips and object contact point are assumed as the friction. This task can study, while two humans move an object to one distance by using the two fingers to analyse the needs of the force and displacement, and the arm must kept stationary (Salisbury, 1982).

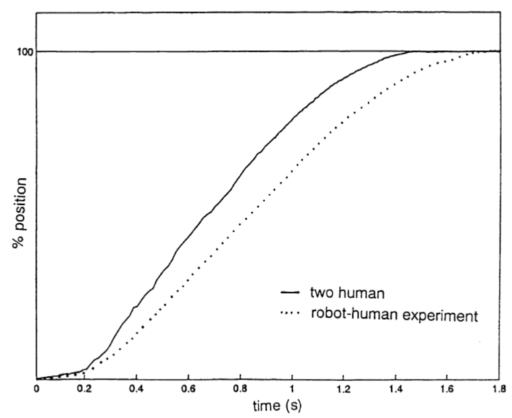



**Figure 2.4:** Initial results of robot–human cooperation experiment

Source: Ashish,(2001)

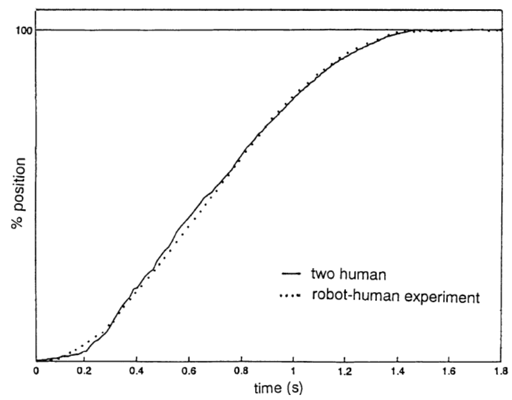



**Figure 2.5**: Result of robot–human cooperation
Source: Ashish,(2001)

Earlier experiments performed using experimental control set-up shows that the normalized error in the desired position finger tips in Figure 2.4. It shows that some corrections have been made to reduce the error in the actual position of the fingertip. This error can be reduced by slightly modulating the mass parameter in the model. Increase the value of the mass parameter means the compensation of the gain and phase lead in the higher frequency range. Therefore, this modulation of the mass function as low-pass dynamic compensation for experimental set-up. Corrected results are shown in Figure 2.5.

The second group of people to analyse various aspects of grasping such as internal force, force closure and the stability. The group consists Kerr and Roth (1985), Yoshikawa and Nagai (1991). The third group consists Westling and Johansson (1984), Howe, Popp, Akella, Kao and Cutkosky (1990), Howe and Cutkosky (1993). Who has analysed in term of physiological aspect are slipping force, shear force and etc. Embedded in the glove to analyse and study the friction, slip and shear force for fingertips.

In the past two decades, various aspects of the research have been done to understand and manipulate. Even have various types of sensors used in industry to measure pressure, acceleration, and it is, but for the shear force measurement sensors are still not available (J.G Webster, 1988).

The intelligent gripper is a combination mechanism gripper with the force and torque sensor (Dwivedi, Sharma and Sharifi, 2009). There are many types of sensors, one of that is a tactile sensor function to sense normal pressure, skin deformation and it is a common type of sensor has been used in robotics (Figure 2.6). It will send the output to the computer controller when the physical contact with the surface texture. Therefore, it can serve to control over the gripper and can detect slipping when grasp the objects at the finger contact (Ashish Duttaa and Goro Obinata 2011).

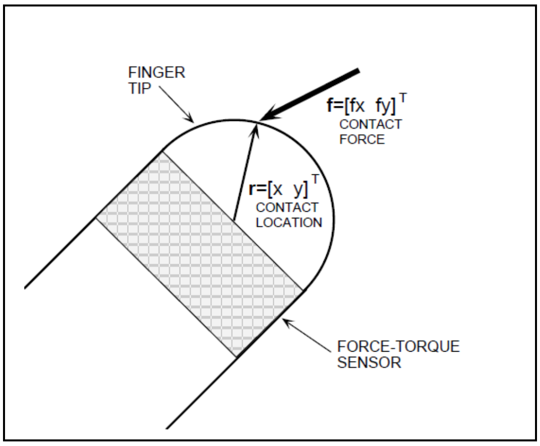



**Figure 2.6:** Sensing contact location with a force torque sensor

Source: J.G Webster, (1988)

### 2.6.1 Inductive proximity sensor

Inductive proximity sensor detects presence metallic objects. Oscillator generates a magnetic field changing around the winding coils are located on the device sensing face such as Figure 2.7 on the next page.

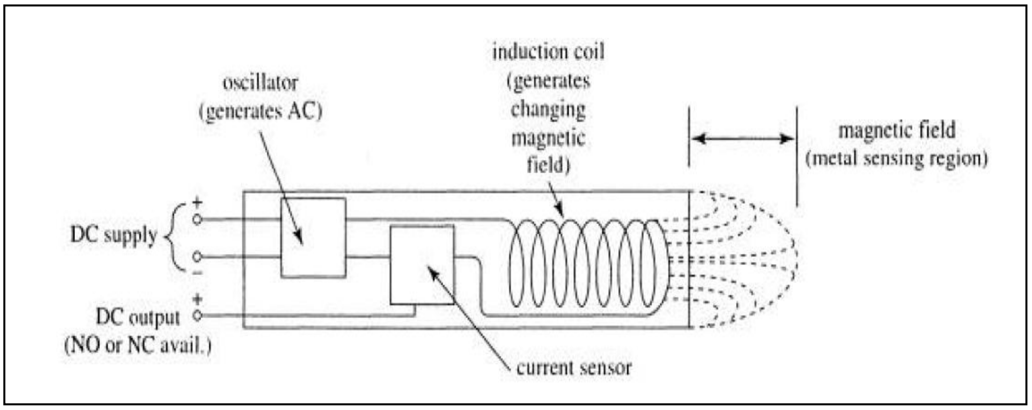



**Figure 2.7**: Schematic diagram of Inductive Proximity sensor
Source: Angelo Davalli, (1995)

The presence of metals in the operating area or metal sensing region, causes a reduction magnetic field in the inductive sensor (Angelo Davalli, 1995). Sensor detection circuit monitors the oscillator strength and trigger output from the output circuit when the oscillator reduces to a sufficient level. The operating range of the sensor depends on the size, shape and materials. Examples of inductive proximity sensor are shown in Figure 2.8

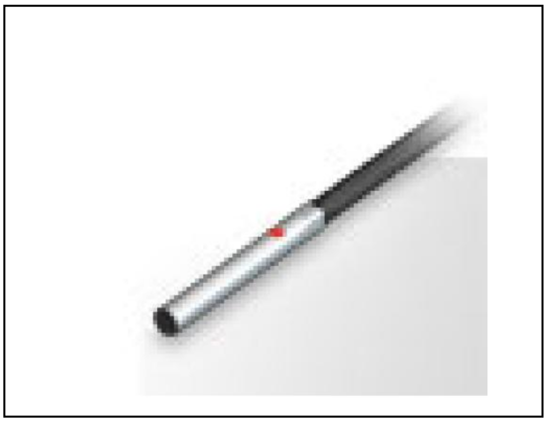


**Figure 2.8**: Inductive Proximity sensor
Source: Angelo Davalli, (1995)

### 2.6.2 Tactile Sensor System

The tactile sensor system consists of sensor arrays 6 by 14 sensors was shown on Figure 2.9.

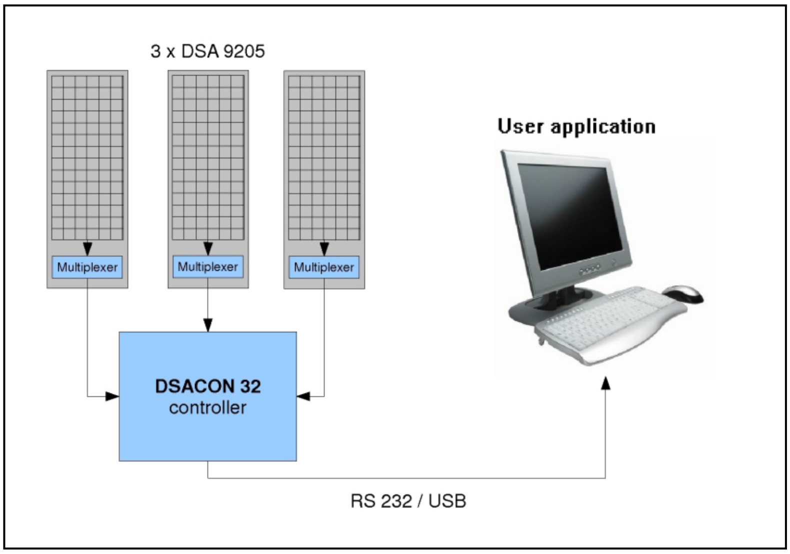



**Figure 2.9**: Tactile sensor system

Source: Andrea Tura, (2001)

It is digitalises measured data transducer is connected and then it can show out via a computer interface. It can use the parallel port RS232 allow serial interface or USB to connect the data. It consists of a microcontoller to read data from tactile matrices and to handle communication data. It uses a specific communication protocol, thus require specific skill and commands to the controller and translate the data series to show out pressure value. (Andrea Tura, 2001)

### 2.6.3 Force Sensing Resistor (FSR)

Force Sensing Resistor is a type of polymer thick film (PTF) device can decrease a resistance when an increase in the force applied to the active surface. Its sensitivity to force is optimized for use in human touch control of electronic devices (Claudio Lamberti, 2001). FSR are not a load cell strain gauge or load cell even almost same. FSR are not suitable for accurate measurements. The image of the FSR is shown on Figure 2.22.

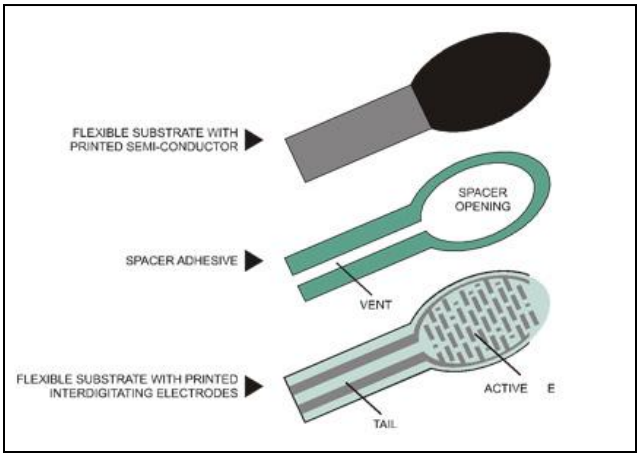



**Figure 2.10:** Force sensing resistor
Source: Claudio Lamberti, (2001)

## 2.7 CONTROL SYSTEM

The control system is device to command, manage and regulated the behaviour of another device and system. The gripper can implement the control system able to handle the various types of object (Ashish Duttaa and Goro Obinata, 2011).

The control system of gripper can be achieved using the feedback of both position and force has been applied. When the sensor detects the object and the gripper has a motion, the sensor will act as encoder to produce output, this positive feedback will sent back to the control system to decode the input signal and analyse data in Figure 2.11 on the next page.

This can be expressed by the following transfer function :

$$\tilde{X}(s) = F(s)/(Ms^2 + Cs) \tag{2.3}$$

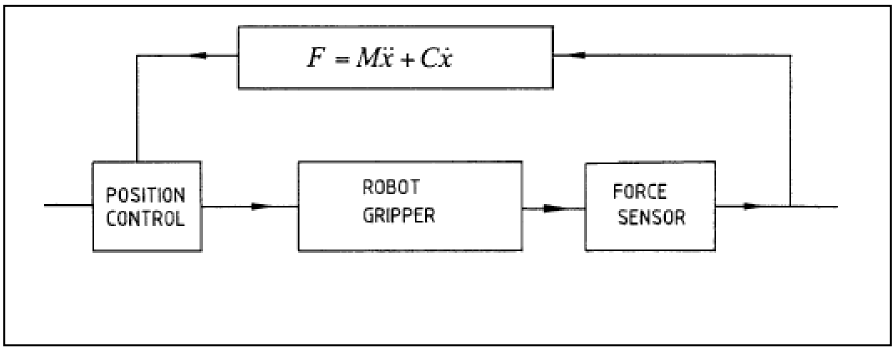



**Figure 2.11:** Impedance control system

Source: Goro Obinata, (2011)

For control more precisely to the gripper jaws, the sensor can attach at the gripper jaws to encode the signal. The signal was generated when detect the object and know the position of the gripper jaws, by this way the system can control the gripper jaws more accurate.

Besides the gripper force is important to avoid deformation of work pieces, various methods were considered to control the force. The strain sensor gauge load cell will mount on the jaw, to measure the pressure. Due to the of cost and convenience, it was utilized armature current as a determinant for proportional force control. Initial experiment was tested with servo motor revealed a good linear relationship between armature current and applied the force within the required range in Figure 2.12 on the next page.

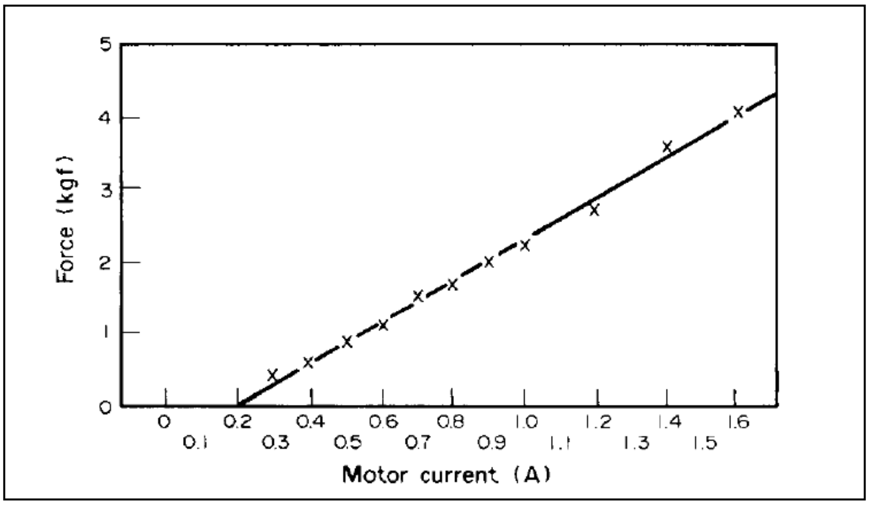



**Figure 2.12:** The relationship between applied force and motor current
Source: C.Melchiorri and G.Vassura, (1992)

Motor driver L293 is used to control the DC motor, its function is to decode input from PIC18F4550 microcontroller and provides power supply to DC motor. It is also able to produce the variable pulse through the output from the PIC18F4550 microcontroller, which is used to control the DC motor.

The feedback signal will derive from a wire resistor was mounting in series DC motor. The voltage drop across the resistor to measure by a differential amplifier in Figure 2.13 on the next page. The signal will pass active filter to remove harmonics. The resulting the sensor will use to provide the feedback signal to the input PIC18F4550 microcontroller and read at the computer.

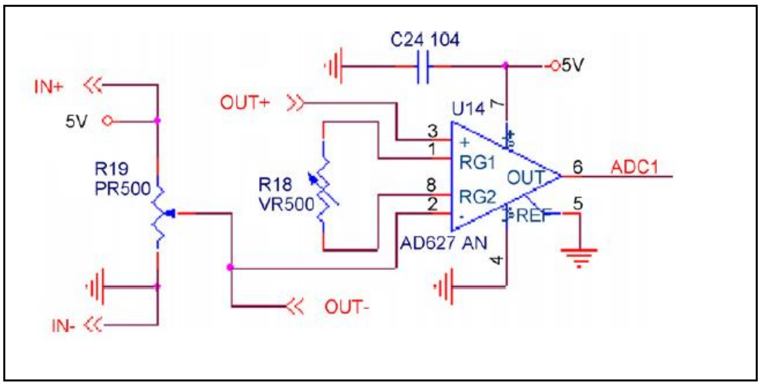



**Figure 2.13:** Electric circuit of amplifier

Source: Gab Soon Kim, (2007)

The computer parallel port has 8 outputs can be written output signal or read input signal. Some of the feedback will through the Opto-couplers, it can prevent the damage from the high voltage.

## 2.8 COMPUTER PROGRAMMING

The computer programming will be used to control the gripper is Visual Basic (VB). It will be used to control the whole movement of grippers. It also can read and write in to know the specific griping action at the appropriate point in programming cycle. So, the gripper can open or close at the desired position. Shimoga (1983), has mentioned implement control system and sensor to the gripper, it can detect the presence of different types of work pieces whether it is a hard object or brittle object. Then apply the suitable gripping force to hold the object.

The calibration cycle would call first, when the gripper jaws want to close under open condition. The sensor acts as an encoder to produce an output, it will generate the current profile to represent the motion of grippers. Encoder gradually increases the output voltage signal indicating the position of the gripper movement, until jaws close completely.

After closing, the voltage output will reach to maximum and it will convert the analog to digital (ADC) signal values to provide the feedback signal. As the gripper closes the jaws, the profile of the gripper is observing and compare with the previously saved profile, then computer able to know whether having the object or not.

The tactile sensor of the gripper is used to ensure the fragile object will not be damaged during grasping. After the jaw closes properly, continue to follow next programming instruction to open gripper when a command is received. Opening the gripper jaws, it will be ready to accept another command (Tedford, 1991).

## 2.9 CONCLUSION

There are several designs had been studied to produce the proper design for the automated gripper. Improvement the design to make it lighter and stronger material should be used to benefit the entire design. The inner contact of the gripper can cover with the soft rubber substance to ensure better gripping capability.

The driving mechanism was designed can increase the efficiency by instead of electric stepper motor, because it is more independent and simply to control the position compared to the electro-pneumatic. Besides the control system has been implemented, able control the movement of the gripper jaws to desired position and more accurate.

The improvement for design the new gripper can improve the reliability and efficiency, it believe able to give a lot of contributions to human at the future..

# CHAPTER 3

# METHODOLOGY

## 3.1 INTRODUCTION

The main objective of this project is the design and development automated gripper control for robotic arm. This project is aimed to design a robot purposely for pick and place, which is commonly used in the industry. This section will present about the methodology design and the development of gripper to fulfil the project objectives.

This chapter discusses the method for the design and development of used for automated gripper. For this project, gripper will be created and attached to the robot as a method to move the things from one place to another, according to the input program and the place to move it.

Design and development gripper consists of four main parts which is the mechanical part, the gripper, electrical part and a controller. Each part will be designed stage by stage. In this project, a DC motor is used to control the movement of the robot. The data will be programmed and executed in the computer control and the output will be sent to the robot so that the robot will grab things and move it according to the place wanted.

While for computer control, Visual Basic programming will be used as the fundamental to build the algorithm for automated gripper as well as the interfacing and coordination among the modules. Various sensors are used in order to make achieve automated gripper controlled. Gripper will also follow the instruction from the user.

Software integration required to combine signals from the sensors and motor controllers.

Control system implemented in computer control is used to make the custom measurement, control the stepper motor. All these elements combined together using programming to develop an automatic gripper control for robotic arm.

All these parts will be tested whole automation. Output acquired during development the automated gripper will be taken and analysed to get better results. Further investigation will be established to implement the most optimum solution for the automated gripper

## 3.2 METHODOLOGY FLOWCHART

The flowchart is used to show the process to make the automated gripper, the process begins from search the all information in journal and using the software to enhance the reliability for automated gripper

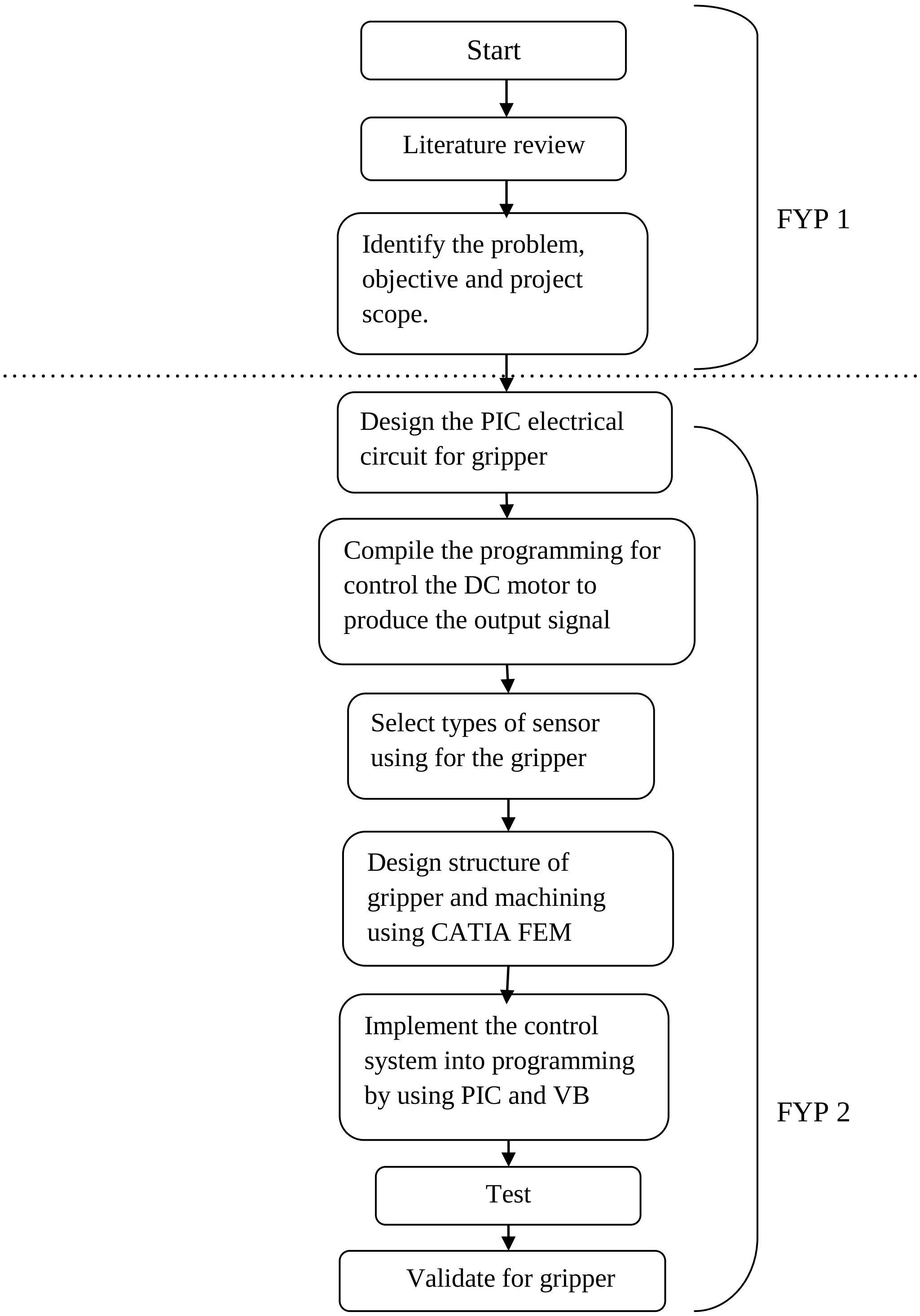



**Figure 3.1:** Flow chart of the methodology

## 3.3 DESIGN METHODOLOGY FOR AUTOMATED GRIPPER

This is the methodology the flow automated gripper function, from the beginner the automated gripper start to initialization type of object to grip until the control system sent out the output to apply suitable forces to grip the object.

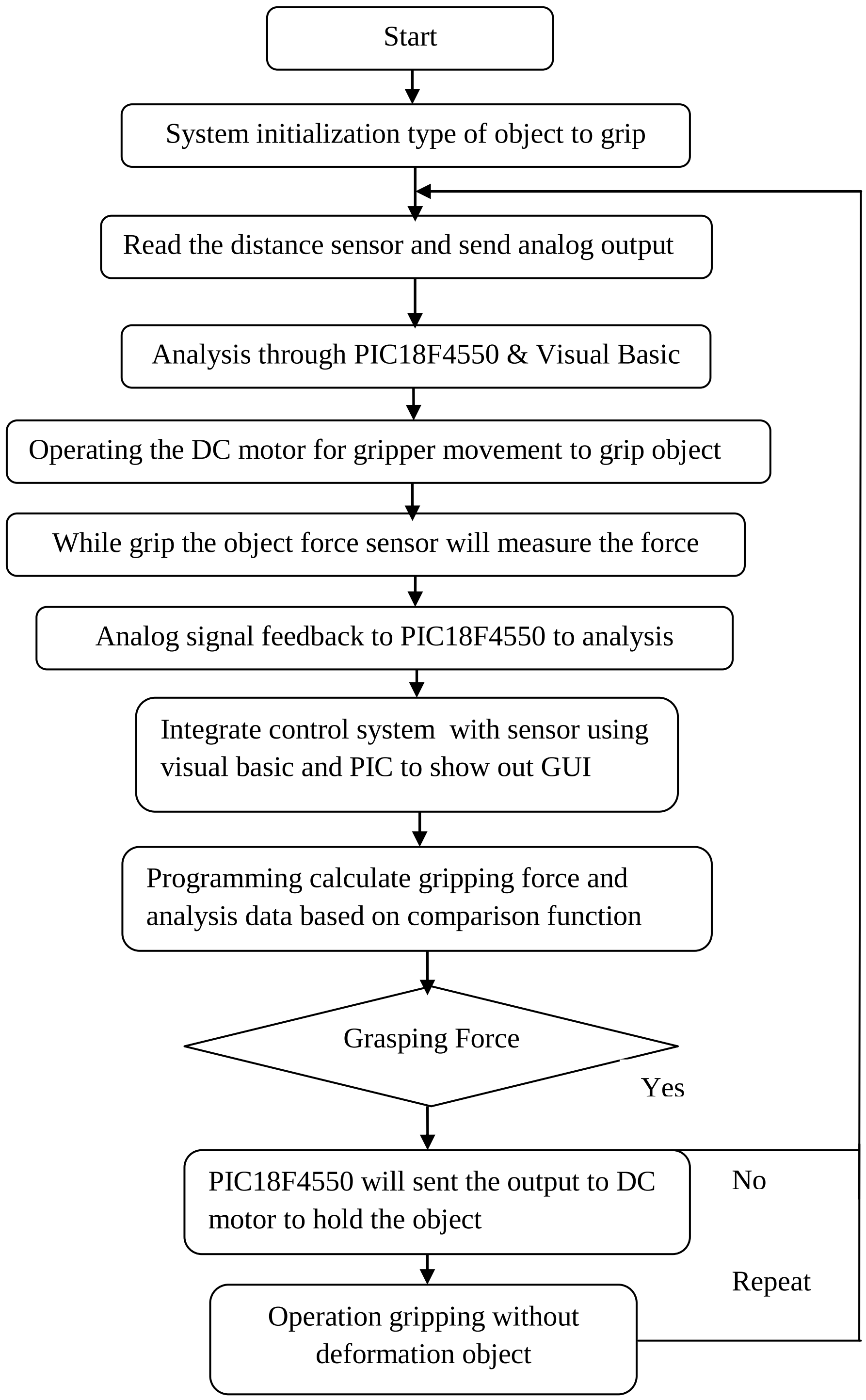



**Figure 3.2:** Flow chart for the control of the automated gripper.

## 3.4 DESIGN ELEMENT FOR GRIPPING FORCE

The basic gripper defines the interface for gripper kinematics to ensure compliance in all respects. Gripper jaw (finger) blanks are easy to install, equip with the clamping contours, and are available in aluminium, steel, a synthetic version. In terms of designing operating elements the gripper is important. When gripper contact with smooth high qualities surface work-piece will direct influence deformation, because of the gripping force was shown in Figure 3.3 and Figure 3.4 on the next page, thus the principle of gripping is important.

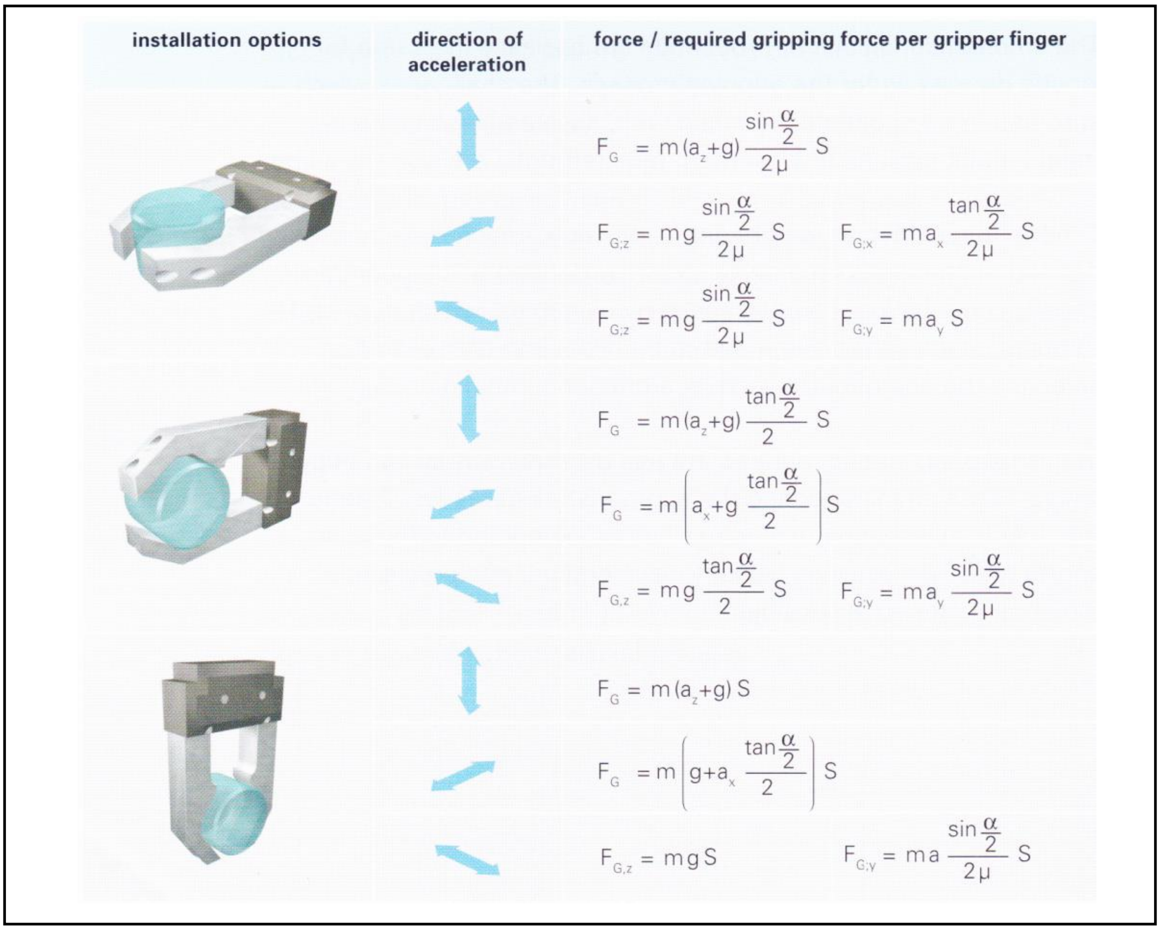



**Figure 3.3:** The gripping force and calculation

Source: Grippers in Motion (Springer)

**Relevant symbols and their meaning**

| | | | |
|---|---|---|---|
| a | acceleration | k | correction factor |
| $a_R$ | radial acceleration | l,L | lengths of links |
| | overall rotary specs | M | moment |
| $a_t$ | tangential acceleration | m | mass |
| $a_{NA}$ | emergency stop acceleration | p | normal pressure |
| $a_z$ | central acceleration | $P_o$ | over-pressure |
| A | plane | $P_u$ | under-pressure |
| B | magnetic induction | r | radius |
| D | diameter | s | distance |
| E | elasticity module | S | security factor |
| F | force | t | time |
| $F_C$ | Coriolis force | v | velocity, translation |
| $F_G$ | gripping force | $\alpha$ | jaw opening angle |
| $F_H$ | force to lift | $\beta$ | auxiliary angle |
| $F_{NA}$ | emergency stop force | $\varphi$ | friction angle |
| $F_S$ | force to fall | $\varsigma$ | opening angle |
| $F_R$ | resulting force | $\mu$ | rotary angle |
| $F_V$ | force to displace | $\mu$ | friction value |
| $F_Z$ | centrifugal force | $\omega$ | permeability |
| G | weight | $\dot{\omega}$ | angle velocity, rotary velocity |
| g | acceleration of the earth | | |

**Figure 3.4:** Symbol of gripping force.

Source**:** Grippers in Motion (Springer)

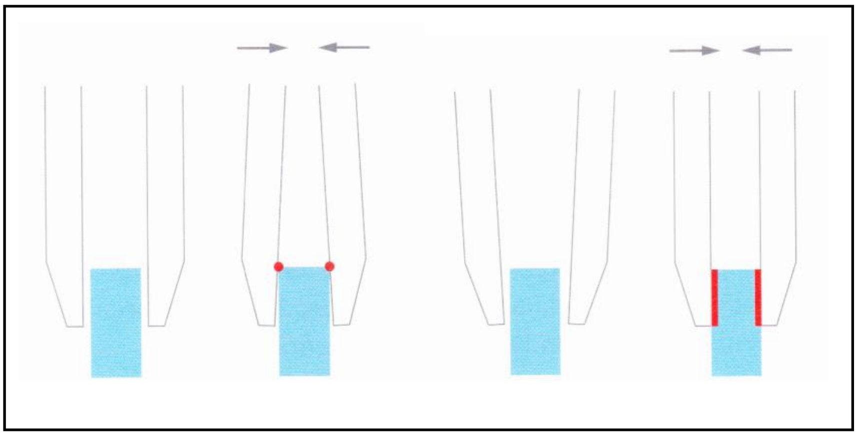


**Figure 3.5:** Face contact calculation of the gripper jaw.

Source**:** Grippers in Motion (Springer)

Special construction is necessary in this case because the high gripping forces and the length gripper jaw may extend the gripper jaw which again may lead to unwanted contact points, such as Figure 3.6 on the next page.

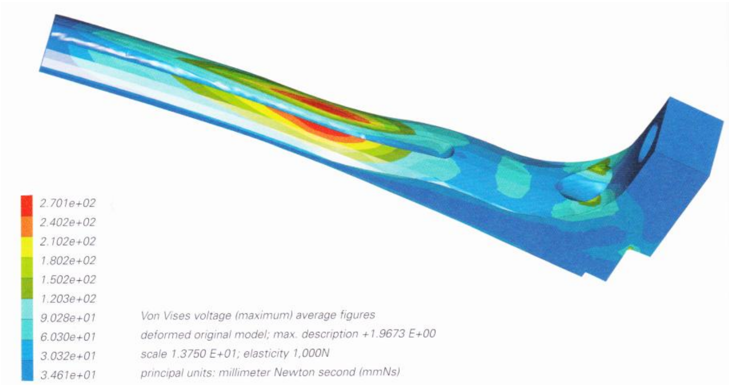



**Figure 3.6:** The gripper analysis by the CATIA FEM

Source**:** Grippers in Motion (Springer)

## 3.5 GRIPPER SENSOR

This part will show the types of sensor will be used for the design and development automated gripper. All types of sensors have a function to perceive and sent output to the computer control for data analysis. Schematic drawing showing several types of sensor have been used for the gripper.

Many sensing devices have been developed for robotic manipulation. All the specific function was listed down at table 3.1. A sketch of a robot hand with some of the most common types of contact sensor is shown in Figure 3.7. These sensors are FlexiForce A201 force sensor at the gripper jaw and analog infrared distance sensor.

| SENSOR | PARAMETER | PLACE |
|---|---|---|
| FlexiForce A201 | Pressure distribution | In outer surface of fingertip |
| Sharp Analog Infrared Distance Sensor | Position object | At the centre gripper jaw |

**Table 3.1:** The important parameters are detected by touch and sensors used to measure

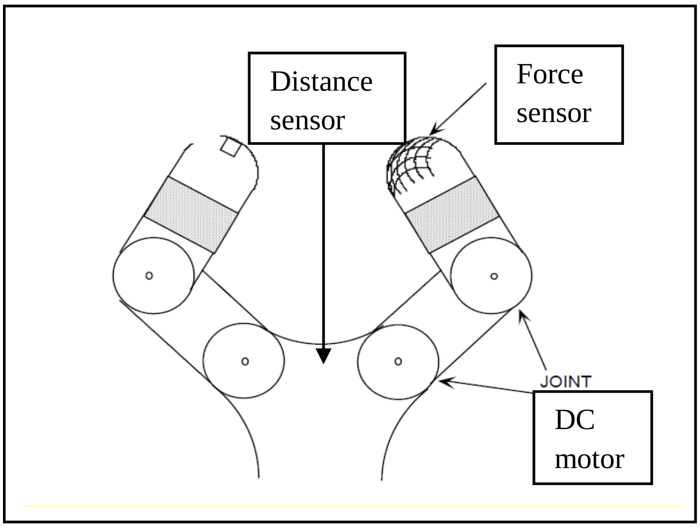



**Figure 3.7:** Schematic drawing of the robot jaw equipped with several types of sensing device for manipulation

### 3.5.1 Flexiforce A201 Sensor

The FlexiForce A201 sensor is an ultra thin, flexible printed circuit. The force sensor is constructed of two layers of substrate film. It consists a conductive material which is a silver colour. Adhesive is used to laminate the two layers of substrates together to form the force sensor. This active sensing area is defined by the silver circle on the top of pressure sensitive ink. This sensor is equipped with male square pins to be easily incorporated with a circuit. There are two outer pins are active and the centre pin in inactive was shown in Figure 3.8.


**Figure 3.8:** FlexiForce A201 sensor

The FlexiForce sensor single sensitive element force sensor  the act as a force sensing resistor in an electrical circuit. This sensor will attach to the contact surface of the fingertip. When the force sensor is unloaded, its resistance is high. As the object is in contact with the sensor which mean a force is applied to the sensor, the resistance will decrease in Figure 3.9. The resistance can be read by checking a multimeter voltage, then try to apply the force at the sensing area. The graph below shows the Force vs Resistance and Force vs conductance. The graph shown the conductance curve is linear in Figure 3.10, therefore this type of sensor suitable use to measure the pressure.

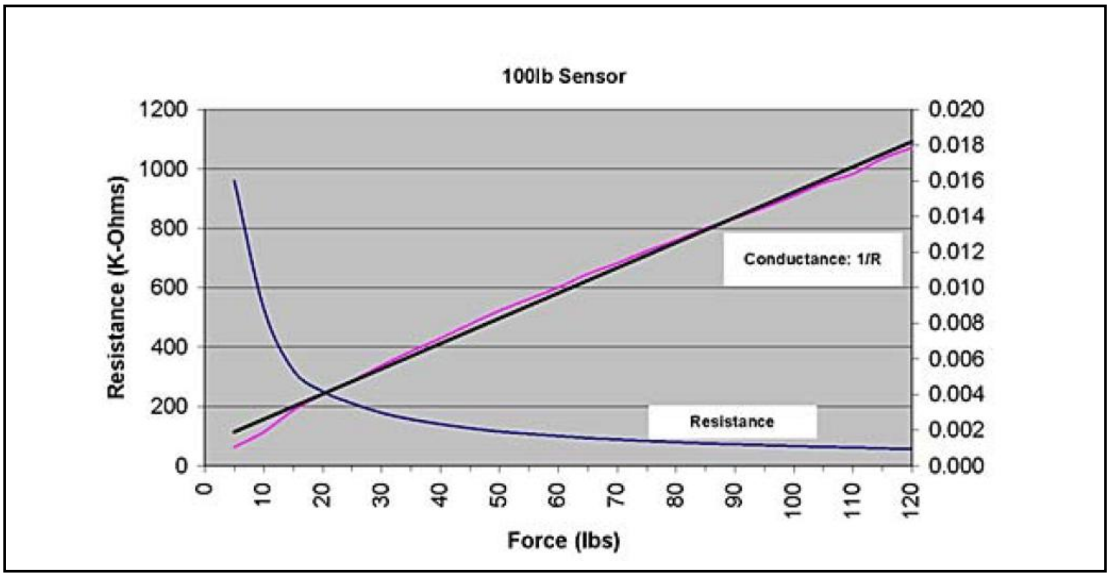



**Figure 3.9:** The graph shows the Force vs Resistance

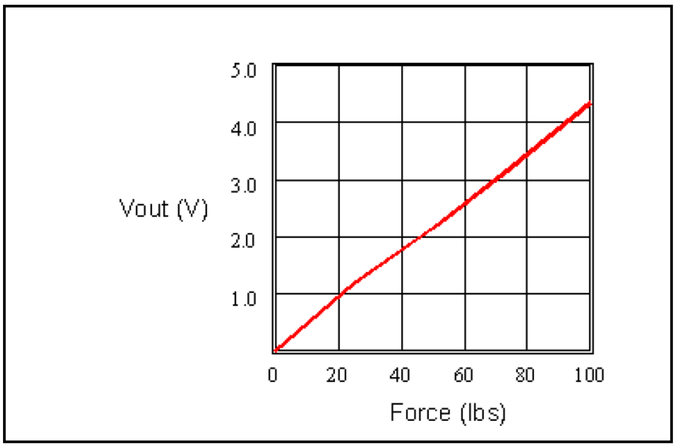



**Figure 3.10:** The graph for Force vs conductance

This kind of sensor is chosen as a pressure sensor at the gripper, because it can attach to many surfaces and can be combined with plastic or metal films to increase the stiffness. When the pressure compress on the surface, its electrical resistance would be changed based on the load. By this way can know the pressure value at the gripper tips when gripper holds the object.

### 3.5.2 Sharp Analog Distance Sensor (4-30cm)

Sharp Analog Distance sensors are used for application and development that require accurate distance measurements in Figure 3.11. This Analog infrared distance is more economical than other sensor such as sonar range finders. Beside it less influence on the colour of reflective object and unnecessary external control circuit Therefore this type of sensor provides better performance to detect and find the location of the object.

It's same like FlexiForce A201 sensor produce a single analog output. Therefore it needs a microcontroller to convert the analog to digital (ADC) for reading distance measurements. Mostly microcontroller is straightforward can read the analog value to interface at LCD screen. For this type of sensor it can operate from 4cm to 30cm, but make sure the power supply for whole the electrical circuit is enough if not will affect the result measurement. The Figure 3.12 on the next page below shows the graph Analog output voltage vs Distance to reflective object L (cm).


**Figure 3.11:** Sharp Analog Distance Sensor (4-30cm)

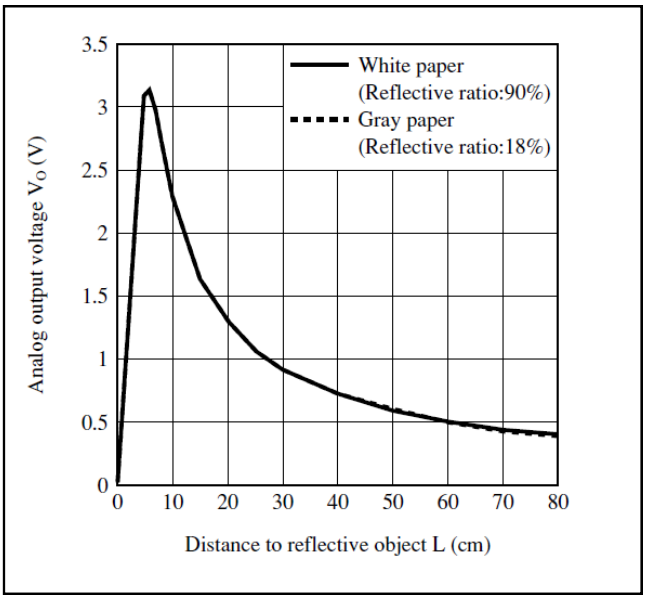



**Figure 3.12:** The graph Analog output voltage vs Distance to reflective object L (cm)

### 3.5.3 DC Gear Motor SPG30-300K

DC motors have been used in open-loop mechanical systems. It is a mechanically commutated electric motor receive the direct current (DC) source to operate in Figure 3.13 on the next page. It consists the stator is a stationary part and rotor is a rotating part. The current in the rotor is supplied through the switched commutator. This is how the relative angle between the stator and rotor magnetic flux is maintained near 90 degrees, which generate the maximum torque. The Figure 3.14 shows the working principle for DC motor.

Different connections in the field winding and armature winding will provide different inherent speed and torque regulation characteristics. DC motor speed can be controlled by changing a voltage applied to armature winding, because the speed of DC motor is directly proportional to the voltage supply as shown in the Figure 3.14 on the next page. DC motors are simple, robust and very reliable to use. It has played important role in robotics to operate gripper, its movement is much reliable than servo and stepper motor.

The SPG30-300k DC geared motor has been chosen to drive the gripper jaw. It has a specification of max 12 volt and having the speed of 12rpm, 300:1 gear ratio, current at 410mA and also 1176 N.m torque. This DC geared motor is chosen because it has a higher torque value compare to other SP30 series geared motor, even it has a low speed. For automated gripper the torque is more important than speed. Many advantages are achieved using this kind of motors, such as

a) Higher simplicity
b) Low cost
c) High reliability
d) High torque at low speeds


**Figure 3.13:** The DC Gear motor SPG30-300K

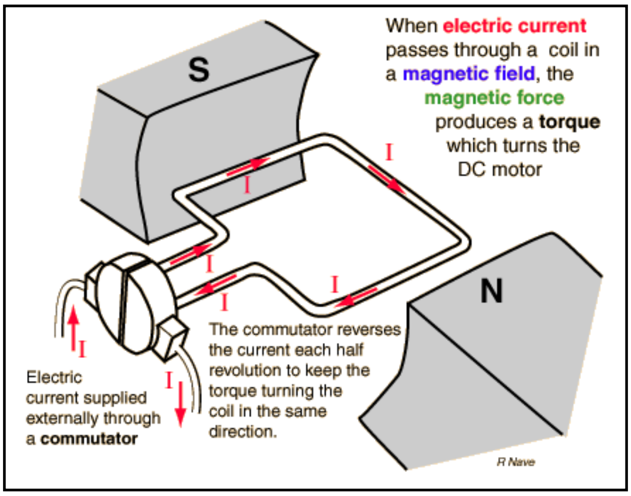



**Figure 3.14:** The working principle for DC motor

Source: Stepper Motor (Wikipedia)

### 3.5.4 DC Motor Connection Diagram

The driver to control DC motor is L293D, it’s an integrated circuit motor driver that bidirectional control in Figure 3.15 on the next page. It is used to connect between the microcontroller and a DC motor. Its function is receiving the signal from the microcontroller, then produces the output for DC motor. The reason it passes this circuit is given a power supply to DC motor and avoid the damage to the microcontroller.

This motor driver is limited to 600mA and handle the small motor, thus it is suitable for the SPG30-300K DC gear motor. But in reality it only can handle small current unless has fix with the heat sink to keep the case temperature down. To ensure this type of motor driver suitable use with DC motor, can run the motor and keeping a finger on the chip. If feel too hot on the chip, that mean this motor driver are not suitable for DC motor. L293 has an automatic thermal function which means the chip will shut down and stop working while too hot. By way means of controlling a DC motor operation have its advantage and disadvantage.

Advantages:

a) Increase or decrease the speed by adjusting the voltage.
b) Bidirectional rotating clockwise or anticlockwise.

Disadvantages:

a) Resonances can occur if not properly controlled.
b) Unable to operate at extremely high speeds.

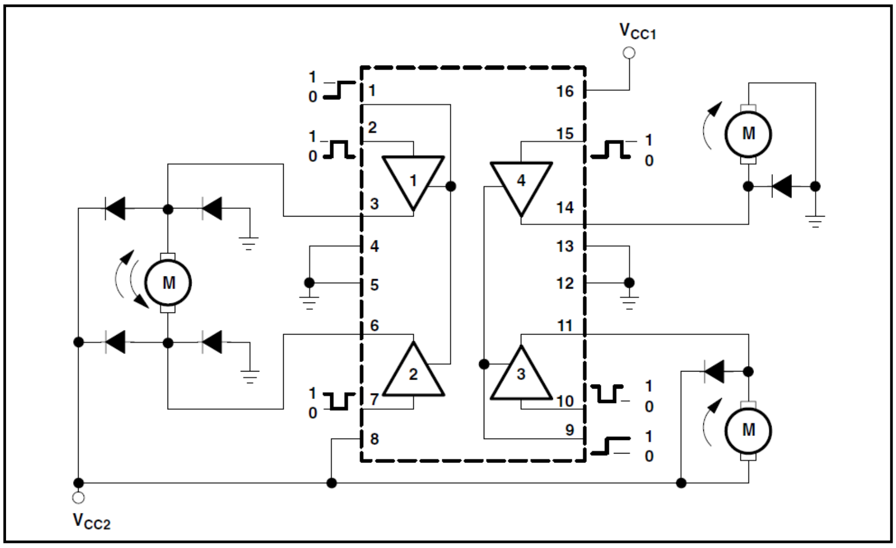



**Figure 3.15:** The working principle for L293D

Source**:** Motor Driver (Texas)

The connection for L293D motor driver and DC motor are shown in Figure 3.16. on the next page. Enable pin connect to +5V to ensure the driver is always enabled and ready to work. Then connect the direction pin which is pin 2 and pin 7 to change directions and turn the motor on and off. The principal to control the direction can refer the table 3.2. Put one high and the other low will rotate one direction. To reverse the direction can reverse the state pin for another direction. If put both pin high the motor will hold the position, to turn off the motor can put the both pin low.

| Enable | Pin 2 | Pin 7 | Function |
|---|---|---|---|
| High | High | Low | Rotate Clockwise |
| High | Low | High | Rotate counter Clockwise |
| High | Low/ High | High/Low | Fast Stop |
| Low | Either | Either | Slow stop |

**Table 3.2:** Principal to control bidirectional DC motor.

Source**:** data sheet (W. Durfee)

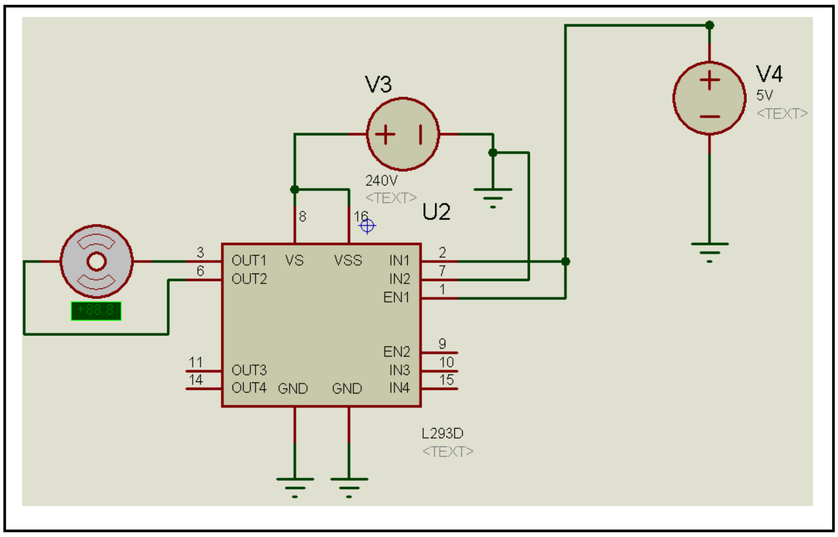



**Figure 3.16:** The connection for clockwise rotation.

Source**:** Protues simulation

### 3.5.5 PIC Microcontroller Start Up Kit (Sk40c)

SK40C board is designed to offer an easy to start board for the PIC user. All interfaces and program can be developed by the user. This board comes with the basic element for the user to begin project development, it equips with plug and use feature 33 input output pins which are nicely labelled to avoid missing connection by users in Figure 3.17 and table 3.3 on the next page. It also provided connector for UIC00 model a low cost USB ICSP PIC Programmer for simple and fast method to load the program. SK40 comes with

- 5V voltage regulator
- Maximum current is 0.5A
- 20Mhz crystal oscillator
- Reset button
- 2X Programmable push button
- Connector for UIC00A
- On/ Off switch for main power
- DC adaptor socket as power input
- UART communication
- USB on board

- All the  necessities to eliminate users' difficulty in using PIC.

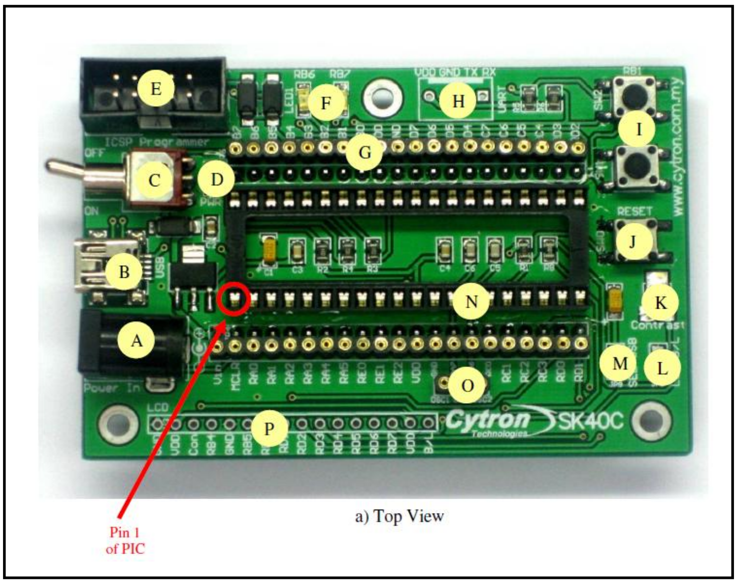



**Figure 3.17:** The PIC Microcontroller Start-up Kit (SK40C)

| Label | Function | Label | Function |
|---|---|---|---|
| A | DC power adaptor socket | I | Programmable Push Button |
| B | USB Connector | J | Reset Button |
| C | Toggle Switch for power supply | K | LCD contrast |
| D | Power indicator LED | L | JP8 for LCD Backlight |
| E | Header pin and turn pin | M | JP9 for USB |
| F | LED Indicator | N | 40 pin IC socket for PIC MCU |
| G | Header pin and turn pin | O | Turn pin for crystal |
| H | UART Connector | P | LCD Display |

**Table 3.3:** Symbol for PIC Microcontroller Start up Kit (SK40C)

### 3.5.6 UIC00B USB ICSP PIC Programmer

UIC00B is a programmer designed to program popular flash PIC microcontroller which includes most of the PIC family. It can program 8 bit, 16 bit and 32 bit PIC MCU. On the board In Circuit Serial Programming (ICSP) connector offers flexible methods to load program. USB port is commonly available and widely used on

Laptop. Thus this device is a very convenient to use UIC00B. It has no required the external power since it connects at the computer USB port and take the power supply from there.

It's compatible with Microchip's PICkit 2. It's powered directly from USB port and no external power required for UIC00B to function. UIC00B also supports on board programming which eliminates the need of plug in and plug out of PIC MCU, this allows to modify the program without removing the PIC from the development board in Figure 3.18. Besides it also reliable and high speed programming.

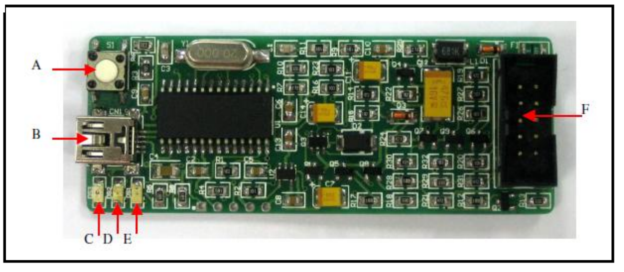



**Figure 3.18:** UIC00B USB ICSP PIC Programmer

### 3.5.7 PIC18F4550 Microcontroller

The microcontroller acts like the brain of the control system in Figure 3.19 on the next page. The microcontroller chip that has been selected for the purpose of controlling the DC motor and read the analog value from the force sensor and distance sensor. PIC18F4550 this chip is selected based on several reasons:

- Its size is small and equipped with sufficient output ports without having to use a decoder or a multiplexer.
- Its portability and low current consumption.
- It is a very simple but powerful microcontroller. Users would only need to learn 35 single word instructions in order to program the chip.
- It can be programmed and reprogrammed easily (up to 10,000,000 cycles) using the universal programmer.

- Enhanced flash program memory makes it deal for embedded control and monitoring applications that require periodic connection with a personal computer via USB for data upload and download.

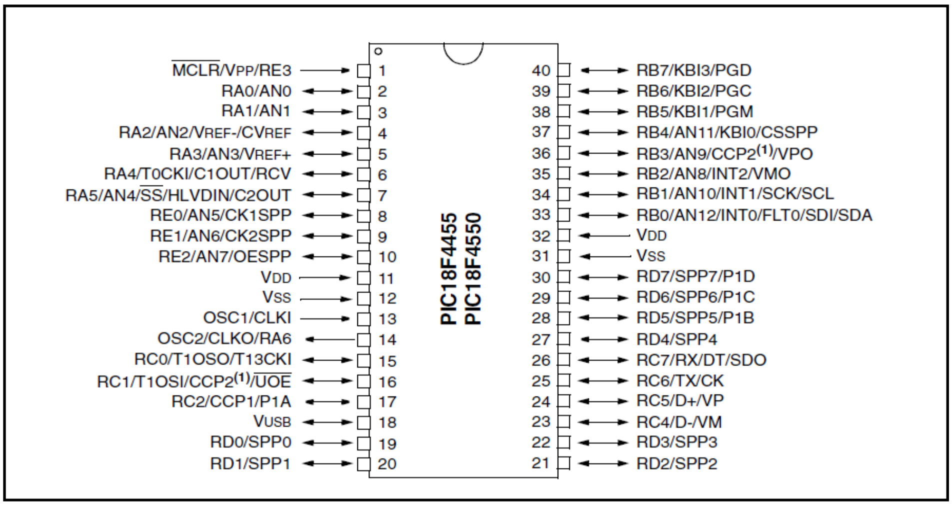



**Figure 3.19:** The PIC18F4550 Device

## 3.6 CONTROL SYSTEM

The control system is the application of control theory for regulation of process without direct human intervention. It is divided into two main type the open loop and the close loop.

### 3.6.1 Open Loop Operation

Open loop control means no feedback information is required for position control. This type of control eliminates the need for expensive sensing and feedback devices such as optical encoders. Commonly the speed of DC motor only can control by adjusting voltage supply, it’s one type of open loop system. One of the most significant disadvantages open loop system apply in DC motor is the speed and rotation hard to control precisely.

Stepping motors can be used in simple open-loop control systems, is generally adequate for systems that operate at low accelerations with static loads, but its torque has a limit. If a stepper in an open-loop control system is missing step due to over torques or excessive speed, all the stepper motor position is lost and the system must be reinitialized. Thus the stepper motor is not suitable for using at the automated gripper, since the gripper requires the high torque to grip the object.

### 3.6.2 Close Loop Operation

To solve the high torque problem, DC gear motor SPG30-300K is suitable to use. DC motors can be used in a closed loop system with the addition of the feedback encoder and the drive circuit. This improvement needs to expense additional cost.

In the close loop, a microcontroller will compare measured value from the force sensor and desired value set by the user. The close loop device use on automated gripper to send the feedback is:

- Force sensor to measure the force
- Analog infrared distance sensor to measure the position object.
- PIC18F4550 as a controller to analyse the data and act as close system
- L293D motor driver to control direction DC motor.
- DC motor to receive the command from the L293D motor driver to move the gripper jaw.

The position of the DC motor can correct back by using PIC18F4550, which receives the feedback from the encoder analog force sensor. After analysing by the PIC18F4550 the data will sent out to motor driver, the DC motor will rotate based on signal produce by motor driver.

The DC motor shaft can be positioned to any angle by sending a coded signal to L293D motor driver. As long as the coded signal exists on the input, the DC motor will rotate the position of the shaft. If the coded signal changes, the position of the DC motor shaft also will change. To build the close loop for DC motor three basic components must have L293D motor driver, force sensor and control circuit with can use the

PIC18F4550 microcontroller. Closed loop control may be important for high accuracy, especially when the gripper involves a variable load to grip.

The force sensor is an important sensor for control system, when the force sensor detects the force is similar to the value force has been set by the user. That means the position is correct, the DC motor will hold the position. If the force being detects is less than the value force set by the user, it will turn the rotate clockwise direction slowly until the force being detected are similar the force value set by the user. Vice versa if the force being detected is over the value force set by the user, the DC motor will rotate counterclockwise until the position of the gripper is correct.

### 3.6.3 Block diagram for control system

This is part of a project control system for automatic gripper. It is used for measuring and control the gripper. A block diagram of the control system has been implemented in a controller, it can analyse the data upon receiving feedback from the sensors can refer to the Figure 3.20.

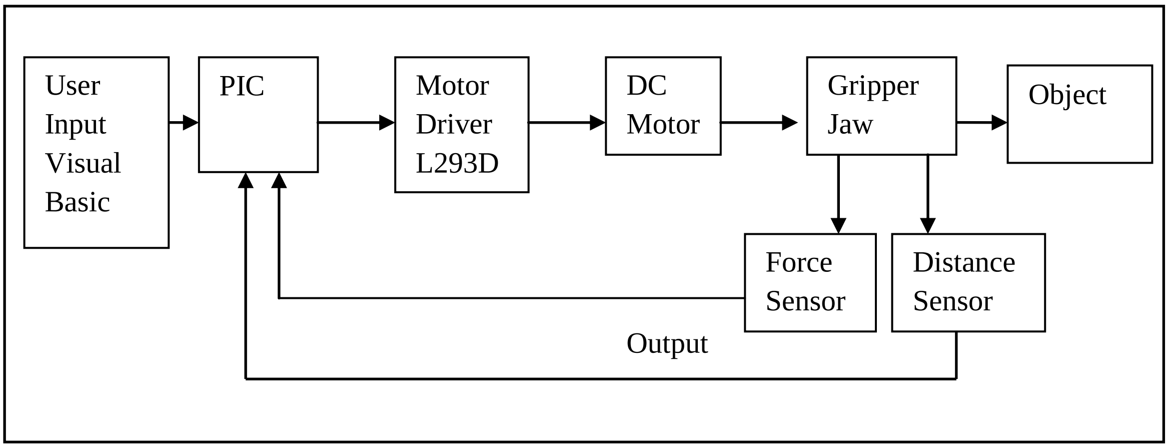



**Figure 3.20:** Block diagram control system for automated gripper.

The control system can help to analyse the data upon receiving feedback from the sensors. Therefore gripping force and position of the gripper can be controlled precisely to prevent work piece deformation. The Figure shows a block diagram of control system for automatic gripper.

Operating systems can be divided into three stages:

1. Initialize the Loop
2. Motion Profile
3. Motion Programming

The first stage, initialize the loop, ensure that the DC motor following ordered position. This is done by using a limit switch to initialize the gripper for the purpose back to the origin position. The motion profile is the generation function to the desired position. This function explains where the motor should be, when receive every command. Profile and initialize the loop are independent functioning. Profiling function determines where the motor should and initializes the loop ask the gripper to initialize and back to ready position. The highest level is motion control programs, the data can be stored in the PIC microcontroller such as desire force value has been set by the user. This program performs the task based on the force and distance sensor has been detected.

The gripper can operate as follows.

1. The user gives instructions to the computer control through the Graphical User Interface (GUI).
2. Computer control will send the pulse to PIC18F4550 microcontroller to set the maximum distance to operate and set the desired force to grip.
3. The user may also select the type of the material which has been set the desired force value at the microcontroller previously.
4. The alternative way pushes the push button near the microcontroller box in Figure 3.21.

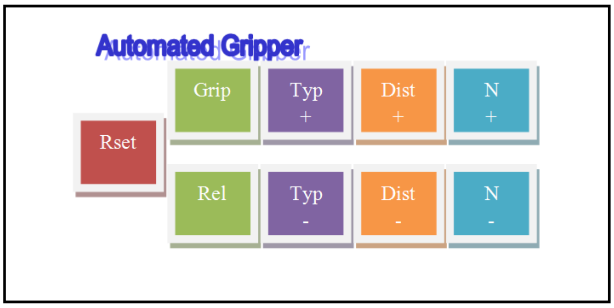



**Figure 3.21:** Push button mode for automated gripper.

5. After the analog infrared distance sensor detects the object within the maximum range. The analog output will received by the microcontroller, thus microcontroller sends the output to the DC motor driver.
6. The DC motor driver will drive the DC motor to move the gripper.
7. When the gripper start moves, force sensor is ready to detect the pressure.
8. The Gripper will grasps unknown object by control the DC motor and follow reference value 2N. The reference value is the desired value set by the user.
9. Force sensor will perceive the appearance of force while grip the object, force sensor would send the feedback to the microcontroller.
10. The microcontroller will calculates the grasping force object convert the analog signal to force value.
11. Microcontroller will make a comparison base of 3 function more than >, less than <, same =.
12. If the force sensor detects is below 2 N, the microcontroller would judge that the gripper does not grasp the objects properly. Which proves the object slides down between the gripper from the force sensor or not yet grip. Repeat the steps 7 to 12 with reference value 2N.
13. Until the weight is 2N, microcontroller control will prove an object that has been grasped.
14. Grasping force is calculated based on feedback from the force sensors.
15. Thus gripper will able to apply the suitable gripping force to grip the object.

## 3.7 DEVELOPMENT OF GRAPHICAL USER INTERFACE (GUI)

One of the scope of the project is to focus on the gripper control with graphical user interface, other areas such as controlling the sensor will not be addressed by the programs developed.

Since the memory of the Programmable Integrated Circuit (PIC) or microcontroller have a enough of the memory and performance, but it still some limitation can’t show the Graphical User Interface (GUI). The computer controller PC based are more preferred to use, because unlimited for capable of storage and performance to programming. Besides it also can use the Visual Basic (VB) to programming, to create the GUI user interface are friendly to user. The user can define the gripper motion sequence and angle to control the gripper without compile complex programming.

Another reason using visual basic to programming is the software can show the value as Figure 3.22 and 3.23 to analysis the data for distance sensor and force sensor. Since the user's host computer will be dedicated to communicate another computer, it is assumed to be realistic to allow the host computer to control the motion sequence of the gripper using by the VB.

### 3.7.1 Program Initialisation

The purpose of the initialization sequence when the program loads are to assign values to variables assigned in the program, such as the resolution of each motor, measurement force sensor of the gripper, define the base address for communications ports, and to read in data.

The final part of the initialization contains the set up of the Inpout32.dll library, this is to inform the Visual Basic program of the functions, and their names, that the library performs. The functions performed by the library are given the function names of ‘Inp’ and ‘Out’.

### 3.7.2 GUI Automated Gripper

The software used to control the two-fingered gripper computer developed using Visual Basic. This software is used to control the movement of the gripper. A screenshot of the developed software is shown in Figure 3.22 on the next page.

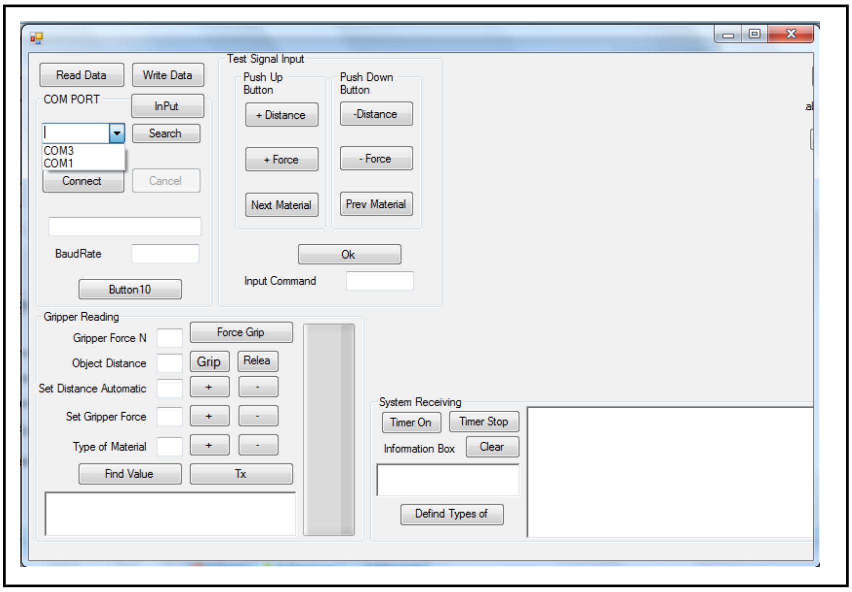



**Figure 3.22:** Screenshot of computer controller through GUI for the two jaw Gripper.

Visual basic application program scans and list the available communication port on a personal computer running the software. After scanning is complete the software allows connection options within the framework of communication port. This frame is used for the selection of the serial port used for communication with the parallel port. This frame is also used to provide information about the status of the communication port. At the beginning of the program, the software notifies the user for selection of communication ports in Figure 3.23.

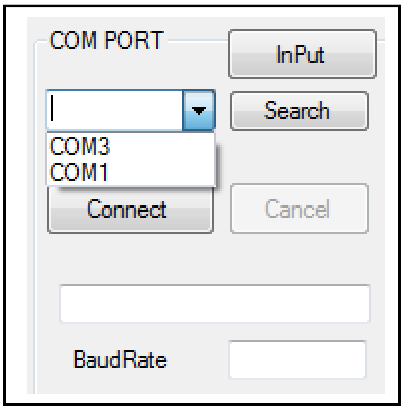



**Figure 3.23:** Communication Port Frame.

If the user proceeds without selecting the appropriate port, it will take the user to select the communication port in Figure 3.24.

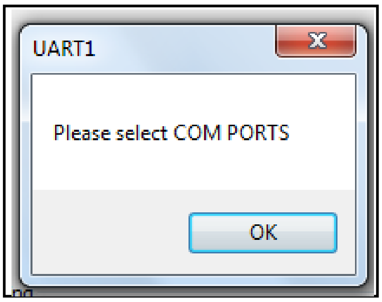



**Figure 3.24:** Warning Message about Communication Port Selection.

If the correct communication port is selected from the options then software detects the gripper would enable the control of gripper in Figure 3.25.

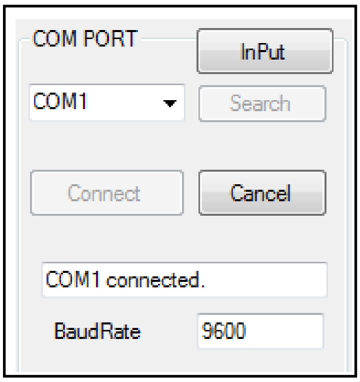



**Figure 3.25:** Gripper Connect on COM1

Status picture box is used to provide information about the gripper. Status gripper connection finger movements are directly reflected in the status bar. Figure 3.26 and Figure 3.27 on the next page shows screenshots of various gripper status.

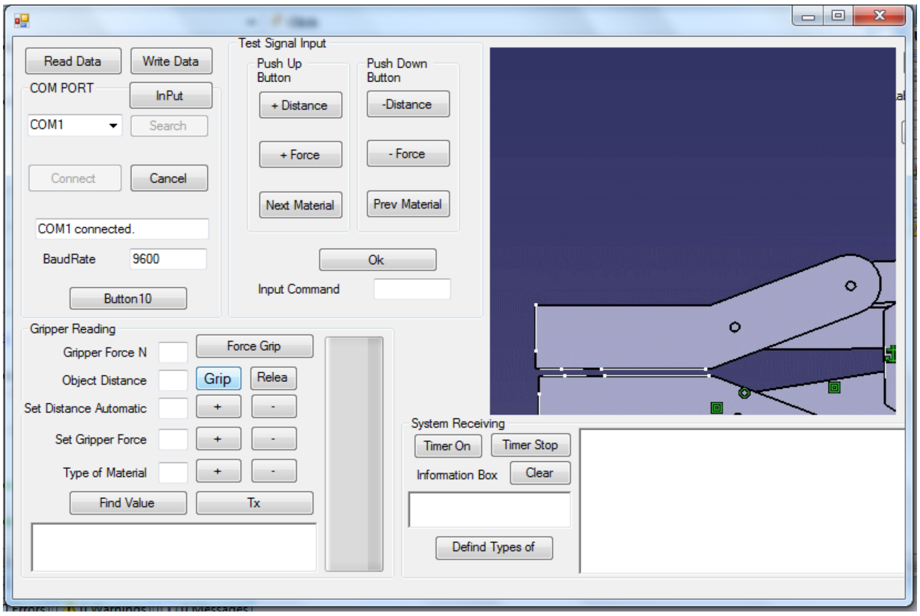



**Figure 3.26:** Gripper Status for grip condition

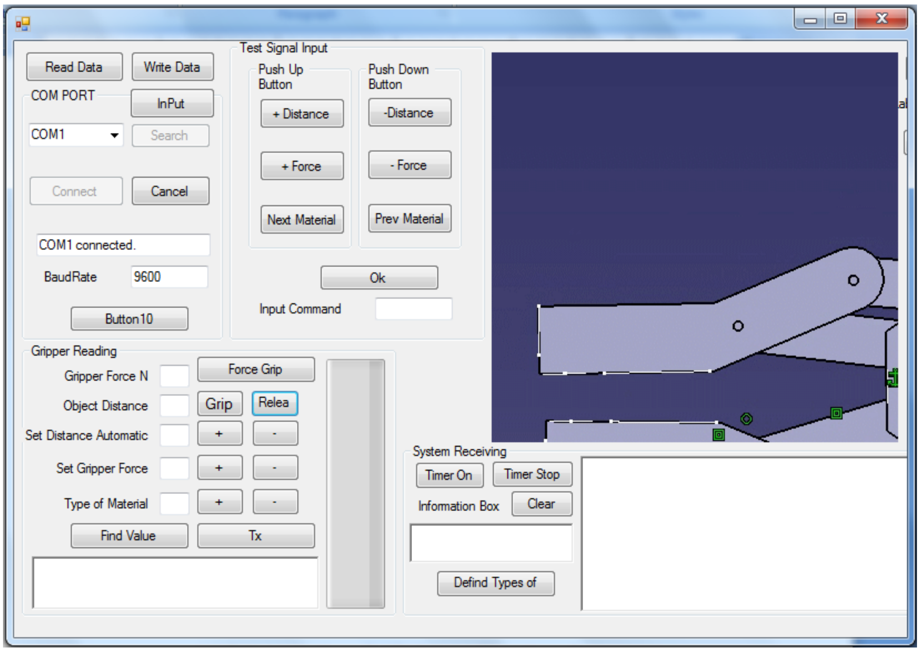



**Figure 3.27:** Gripper Status for release condition

Control framework is used to control the movement of the finger gripper finger. Grip button is used to activate the gripper jaws. Release button in the frame used for deactivation gripper jaws. Finally, the exit button is used for gripper deactivation and termination of programs.

### 3.8 COMMAND FOR THE GRIPPER

Grip command used for activation related to movement of gripping jaws. Release command is used for gripper deactivation.

The framework used for the gripper status information about the gripper. Status gripper connection finger movements are directly reflected in the picture box. Figure 3.27 shows a screenshot various gripper status.

This software uses RS232 serial interface to communicate with the PIC18F4550 microcontroller. It transmits necessary data from the computer to the parallel port and then sent to the PIC18F4550 microcontroller through the USB UART. The microcontroller will analyse the data and sent the output to the DC motor driver L293D. After the driver motor was receiving the data, it will provide the power supply and generates signals for the stepper motor.

The frames of the visual programming application can be described as follows, The communication frame is used for selection of the serial port number used for communication with PIC18F4550 microcontroller. After successful connection with PIC18F4550 microcontroller, visual basic programming enables the access to gripper control frames.

Grip buttons used to activate two fingers. Release button used for deactivation of them. Demo button is used for demonstration. This button activates and deactivates the gripper fingers in a predetermined order.

### 3.9 PROGRAMMING FOR PARALLEL PORT

The parallel port is a port contain a signal line to send and receives the data with other components between hardware and software. It commonly connects to the printer and modem in Figure 3.28 on the next page. At the signal, 1 is represent signal ON in 5V, and 0 is represent signal OFF in 0V operate such as binary system and it is multi directional. It can send 1 bit and receive 1 bit at a time at the same times it show that parallel port only provide 5V in table 3.4. Parallel port use

a. 8 output pins to send the data at a time..
b. 5 input pins to receive the data at a time.
c. 4 output pins as the control port it can use for input and output the signal.
d. The remaining 8 pins are grounded

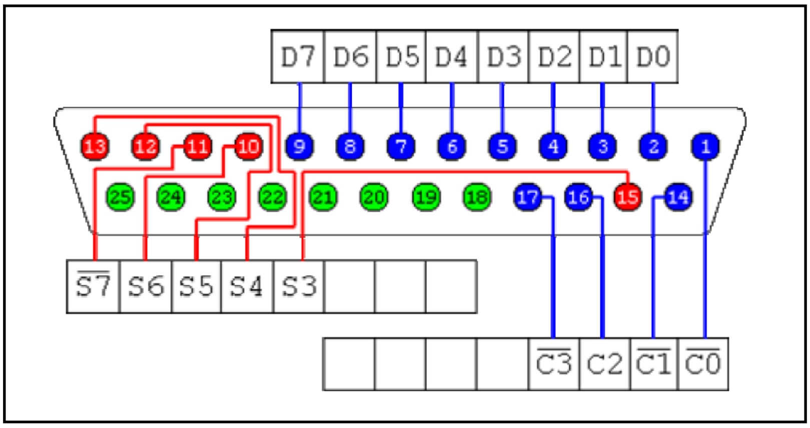



**Figure 3.28:** 25way Female D-Type Connector

Source**:** Parallel Port (Wikipedia)

In this project, only 10 data pins are used:

| | Pin Num | Function | Input or output |
|---|---|---|---|
| 1. | Pin 2: D0 | Push up button to add force for PIC | Output |
| 2. | Pin 3: D1 | Push up button to reduce the force for PIC | Output |
| 3. | Pin 4: D2 | Push up button to add maximum distance in the range for PIC | Output |
| 4. | Pin 5: D3 | Push up button to reduce the maximum distance in the range for PIC | Output |
| 5. | Pin 6: D4 | Push up button to select type of material | Output |

| | Pin Num | Function | Input or output |
|---|---|---|---|
| 6. | Pin 7: D5 | Push up button to select the previous type of material for PIC | Output |
| 7. | Pin 8: D6 | Push up button manually grip for PIC | Output |
| 8. | Pin 9: D7 | Push up button manually release for PIC | Output |
| 5. | Pin 10: S6 | To receive the output from the PIC show grip condition means clockwise | Input |
| 6. | Pin 11: S7 | To receive the output from the PIC show release condition means counterclockwise | Input |

**Table: 3.4:** Connection parallel port with the component.

## 3.10 ANALYSIS STRESSES FROM CATIA FEM

Automated Gripper has been designed using CATIA software, the reason using CATIA is this software can design all the specific part in automated gripper and easier to make users understand and fulfil the client requirement. After design all the specific parts for automated gripper, can simulate the motion of automated gripper in Digital Mockup kinematics (DMU) vin Figure 3.29 on the next page. From the simulation the designer able to know the condition for automated gripper whether is functioning properly. From the Figure 3.29 can observe the product successfully designed. This Automated gripper was designed based on the concept parallel gripping, can refer to Figure 3.5. Because the concept parallel gripping has a large surface contact, that can minimize the deformation of the object.

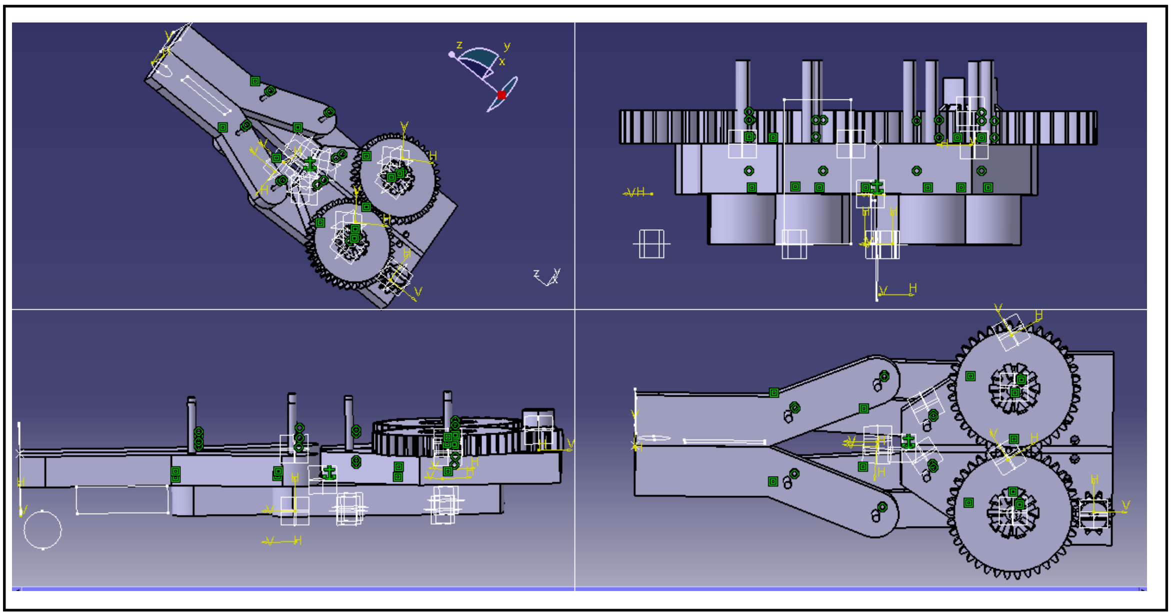


**Figure 3.29:** Structure automated gripper has been designed

Automated gripper is designed to grip a variable object, therefore every time the force to grip the object is different. The structural design and material have to consider, CATIA has a function generative structure analysis which is called Finite Element Method (FEM). In Figure 3.30 on the next page has shown the structure analysis von mises stress for automated gripper has been done. The right hand side has an indicator to show each type of colour represent the von mises stress. The blue part represents the most stiffness part until the red colour is the most brittle or weak.  In Figure 3.31 on the next page is used for analysis displacement for automated gripper. By using this software able to structural design at the same times able to analysis the weakest part for automated gripper.

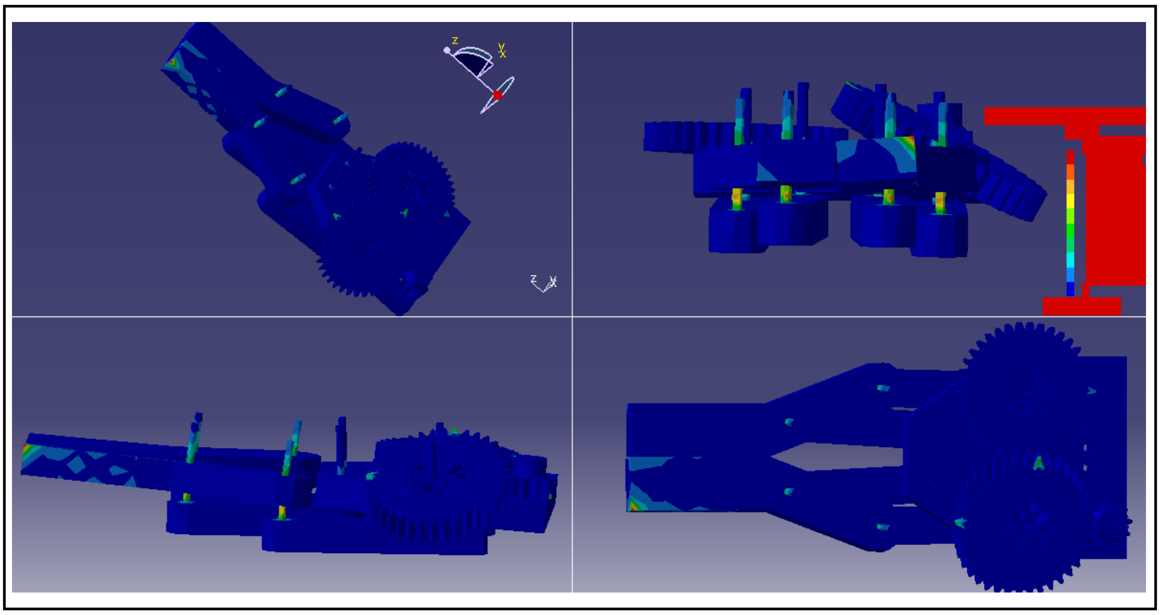


**Figure 3.30:** Structure analysis von mises stress.

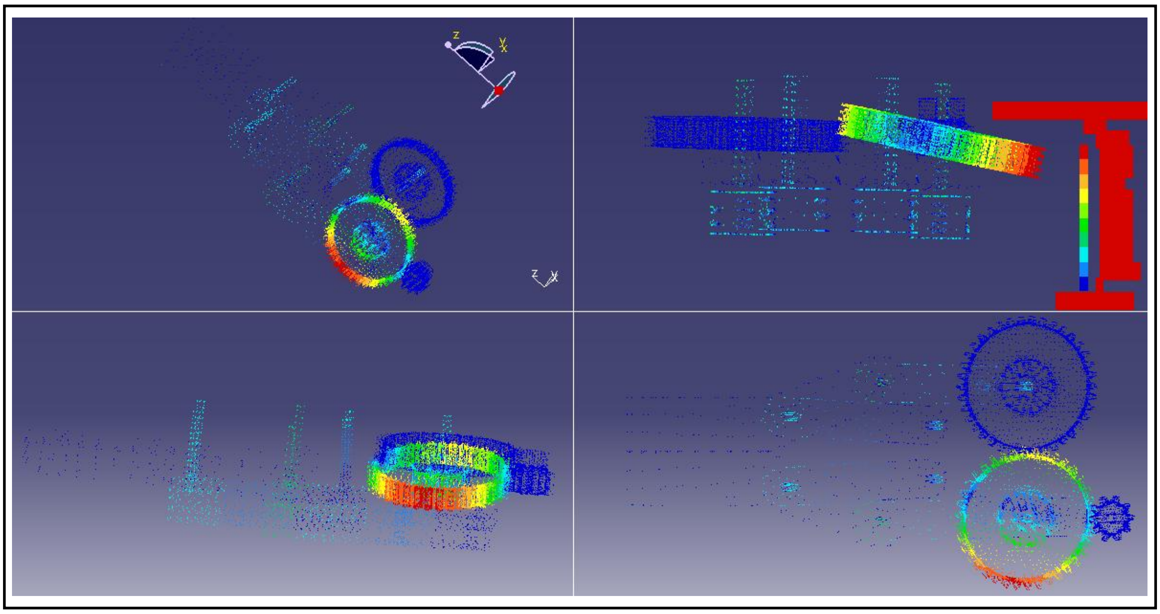


**Figure 3.31:** Structure analysis displacement for automated gripper

In this chapter have described some specification and physical implementation of development and design the automated gripper. The gripper has a sensor and control system can increase the ability and performance, especially in the industries are demanding speedy and efficient work.

# CHAPTER 4

# RESULT AND DISCUSSION

## 4.1 INTRODUCTION

~~In~~ this chapter will show all the results, which has been planned and proceed in the previous chapter. That includes the hardware for automated gripper, connection DC motor with PIC18F4550 microcontroller, programming the CCS compiler microcontroller and Visual Basic Graphical User Interface (GUI). After this all the parts combine, an experiment is conducted to find out the data, Then data collection is done would be sketched in the graph to observe the performance of the automated gripper.

## 4.2 AUTOMATED GRIPPER HARDWARE

An automated gripper hardware can be produced from CATIA drawing, in Figure 4.1 on the next page has shown the automated gripper was using CATIA drawing. After the drawing was drawn, it can assemble with other part of the automated gripper to simulink  the movement for automated gripper.

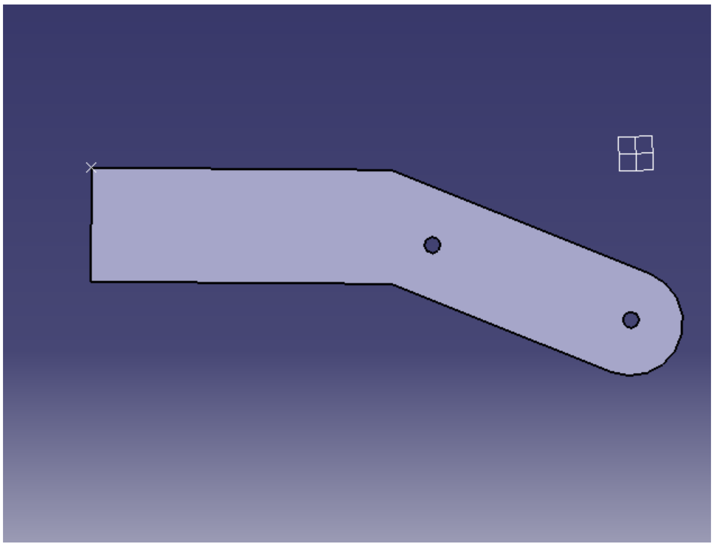


**Figure 4.1 :** The part design for automated gripper

The part design has been designed properly, part design can produce by using an advance maching function in CATIA. In Figure 4.2 show the route has been created to milling the part. The route will show the way across by milling machine to mill the product.

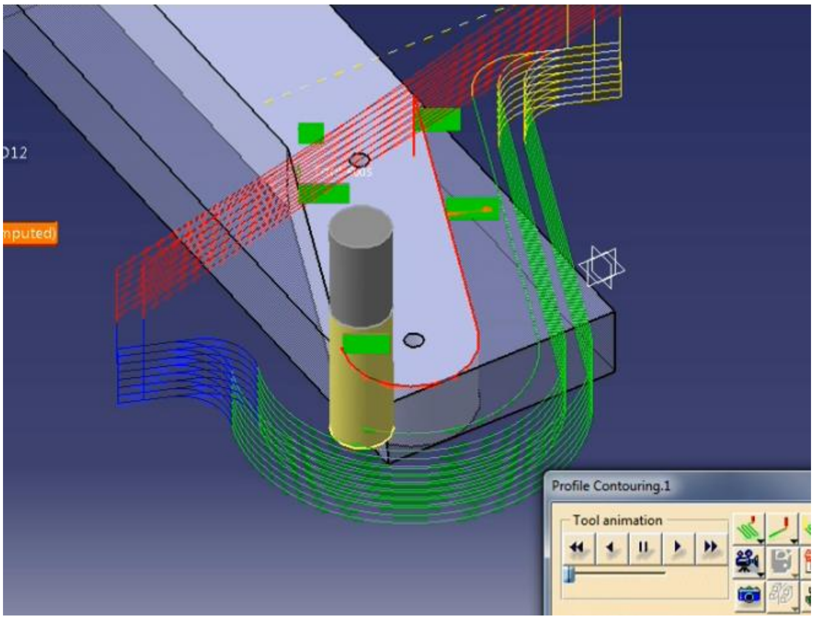



**Figure 4.2 :** Route for machine milling

After the route has been set, the user can generate the G code for milling machine to perform the task. The G code cannot send to the Makino KE55 milling machine directly, that has some changed in the  header was shown in Figure 4.3 on the next page.

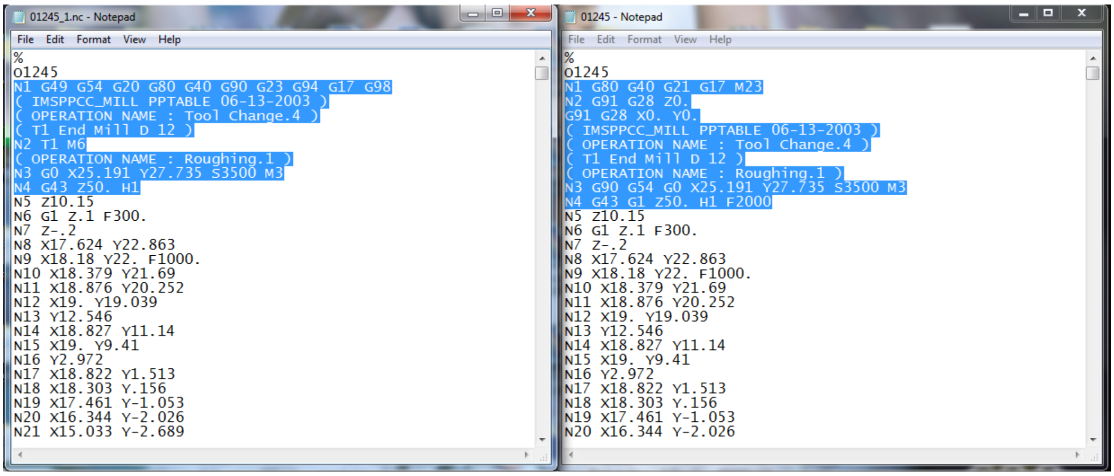



**Figure 4.3 :** The header has to change based on a Makino KE-55 milling machine

After sending the G code to through the computer, the milling machine will mill the workpiece by following the route setting in CATIA. At the end, the aluminium workpiece had been milled would seem like a real product show in Figure 4.4.

**Figure 4.4 :** The real product

Almost all the parts are used the milling to make, but certain part has to make by the rapid prototype machine using Acrylonitrile Butadiene Styrene (ABS) plastic material. Because the milling machine has no capability to make a smaller radius less than 2mm was shown in Figure 4.5 on the next page.

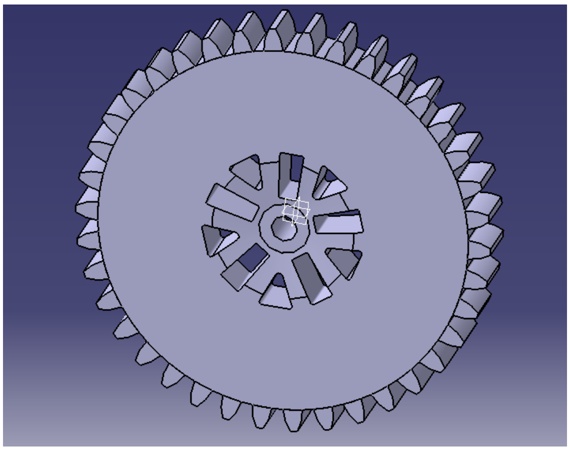


**Figure 4.5 :** The part design for big gear

It's easier to make, the part design has been designed by the user, can directly save in stl file format. This stl format will put in the rapid prototype machine to make the product. In Figure 4.6 show the product has been made by rapid prototype machine by follow the part design in Figure 4.5.

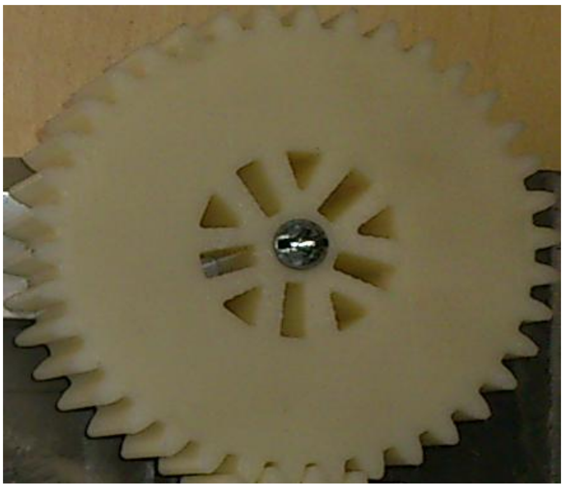


**Figure 4.6 :** The real product big gear make by rapid prototype machine

All the automated gripper part has been made by Makino KE 55 milling machine and rapid prototype machine. The next step combines all the parts in Figure 4.7 on the next page show the assembly design for automated gripper in CATIA.

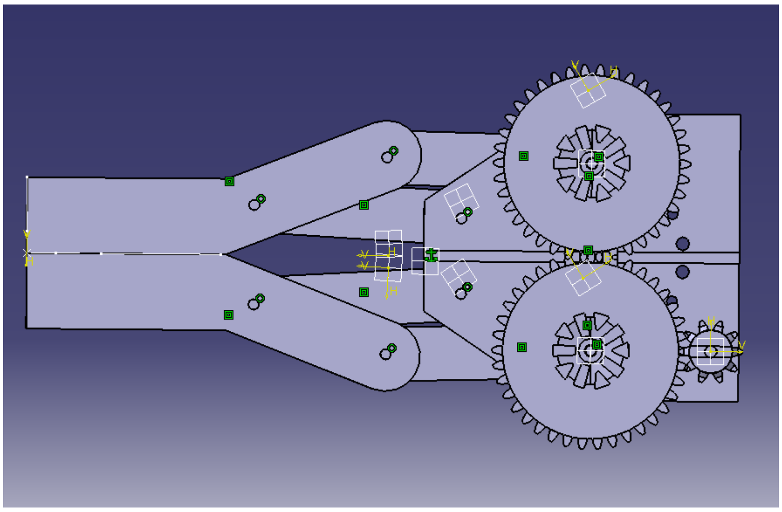


**Figure 4.7 :** The assemble design for automated gripper

After assembly all the real parts, the hardware for automated gripper in Figure 4.8. The hardware automated gripper is exactly same like CATIA design. Its movement is quite sluggish most of the energy lost in friction, because almost all the parts are made of aluminium and it's quite heavy

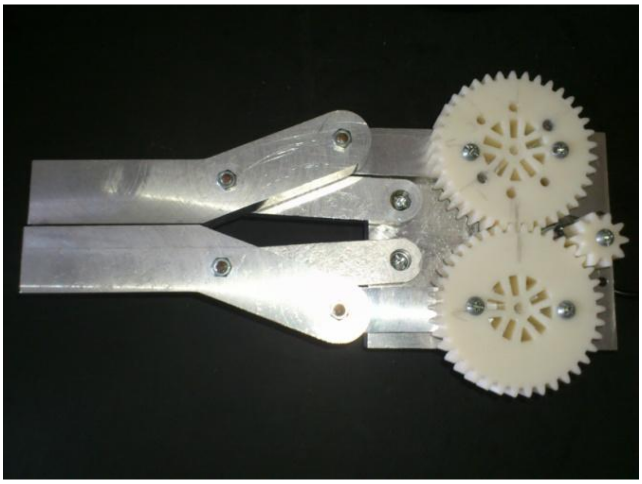


**Figure 4.8 :** The real automated gripper

The energy loss in friction can be overcome by adding the bearing in all the moving parts. In Figure 4.9 after adding the bearing, the friction was reduced and the movement becomes smoother.

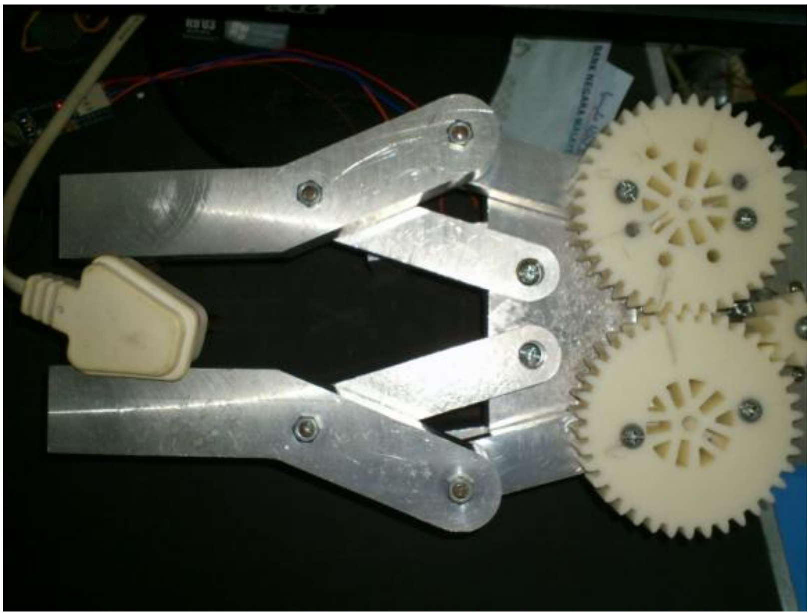


**Figure 4.9 :** The mechanism for automated gripper

## 4.3 ELECTRICAL PART FOR AUTOMATED GRIPPER

The automated gripper main control system is controlled by PIC18F4550 microcontroller. All the circuit is designed at the Protues 7.7 version. This software able to simulink the programming in hex format, which is compiled by CCS compiler. In Figure 4.10 on the next page show the circuit completes with all the electronic device include an LCD screen, com port for UART, DC motor, push button, Sharp infrared distance sensor and virtual terminal for reading input and output from UART. Since the flexi force sensor cannot find at the Protues library. This sensor can replace with the temp sensor, because it also uses the range from 1mV until 1V  for analog output same like a force sensor.

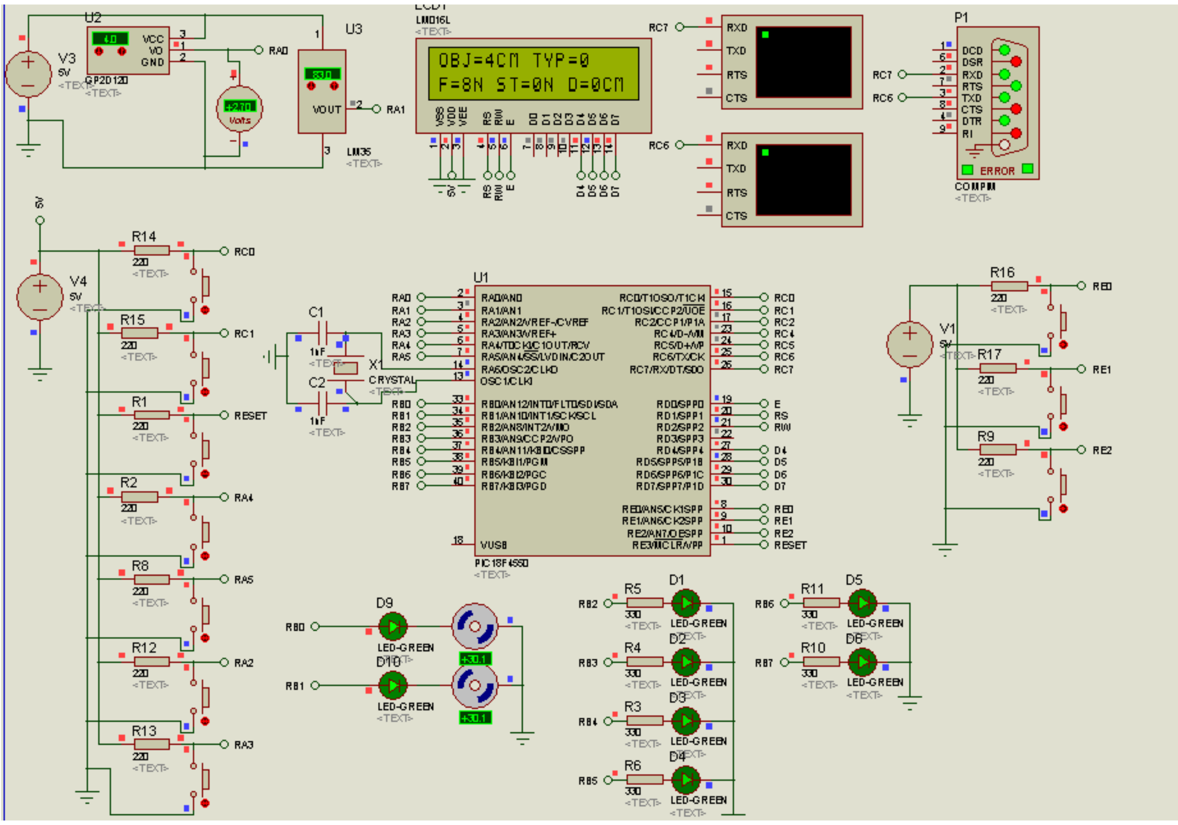



**Figure 4.10 :** The complete circuit for automated gripper in protues

After design the circuit in the protues, can try to insert the programming in hex format by double click the PIC18F4550. The hex file can insert by found out the location where the file it is, then set the clock processor frequency must match with the programming and real crystal frequency in Figure 4.11 on the next page. If all the simulink can function properly, that means a big possibility for real circuit can be worked.

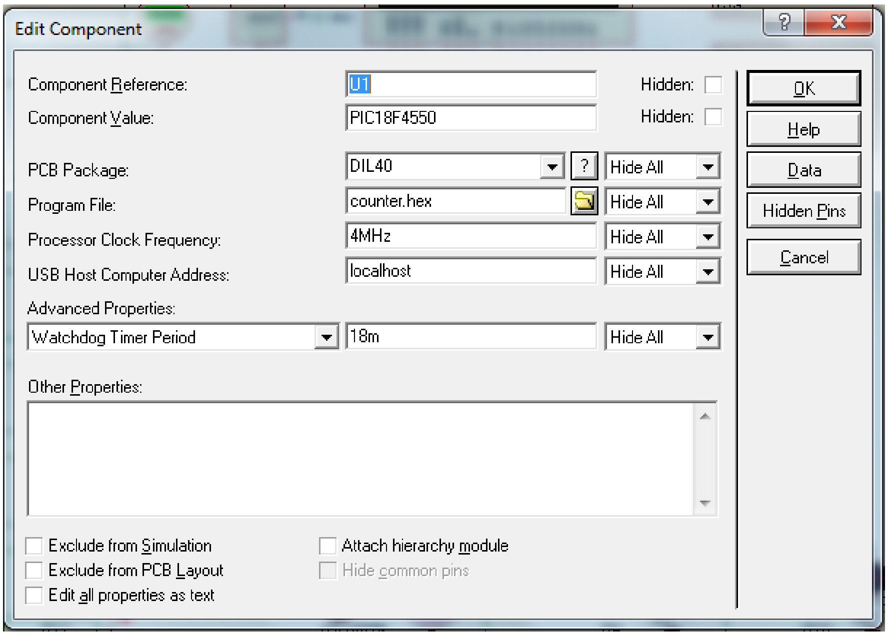



**Figure 4.11 :** The edit component

In Figure 4.12 shows the real circuit for automated gripper which has been done, all the connections are based on the complete circuit in protues can refer to Figure 4.10 and Table 4.1 . In the real circuit it compenent almost same like protues circuit, only add the ICSP which has mentioned in chapter 3. ICSP port is used to program the PIC18F4550 microcontroller.

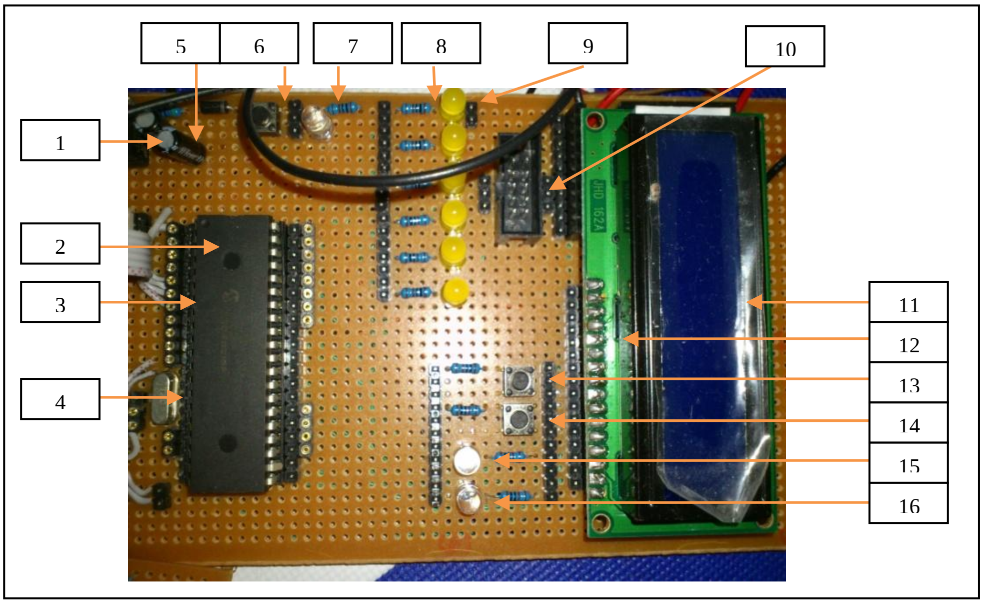



**Figure 4.12 :** The real complete circuit for automated gripper

| Label | Component | Label | Component |
|---|---|---|---|
| 1 | Voltage Regulator | 9 | Push up and down indicator |
| 2 | PIC18F4550 | 10 | ICSP header |
| 3 | Female pin header | 11 | 2x16 LCD screen |
| 4 | 4Mhz Crystal | 12 | LCD female header |
| 5 | 0.1 uF Capacitor | 13 | Button to grip |
| 6 | Reset Button | 14 | Button to release |
| 7 | Reset Indicator | 15 | Indicator for grip |
| 8 | 1khz resistor | 16 | Indicator for release |

**Table 4.1 :** Component for automated gripper

In Figure 4.13 show the L293D motor driver for a DC motor, it has been designed fix with IC pin. The purpose using the IC pin is the IC motor driver can change easily, when the motor driver is breakdown.

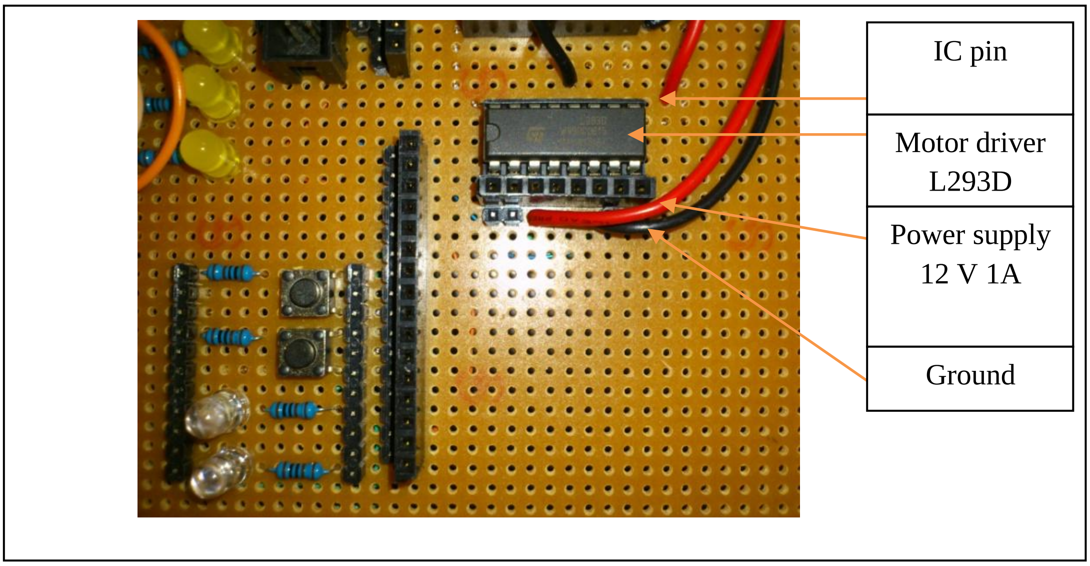



**Figure 4.13 :** The circuit for motor driver

Most serial interface from microcontroller to computer is done through the serial port, but now serial port has been phased out and replace with USB UART. The UART can direct interface with microcontroller and it provides low current 5V from USB port.

UART consist 4 ports which is 5V, ground, TX and RX. RX mean receive the data from PIC, and TX transmits the data to PIC. Pin RX from UART will connect to the TX pin PIC18F4550 at RC6, and Port TX from UART will connect to the RX pin RC7 in Figure 4.14.

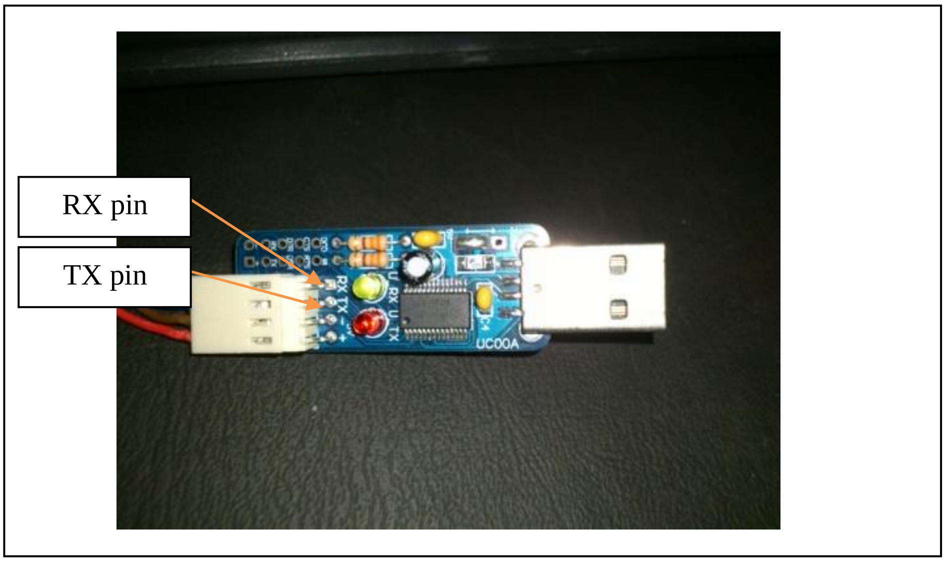



**Figure 4.14 :** UART transmit the data to PIC

In this project box, it consists 5V female connector and 12 V female connector. The 5V female connector is receiving the power supply for PIC18F4550 microcontroller in Figure 4.15 on the next page. Another female connector 12 V is direct supply the power to the motor driver. This two power supply can’t use in one power supply, because the driver motor will produce the glitch which is the disturbance for microcontroller, then will affect the result. The parallel port RS 232 at the project is used to connect push down button and it is control by Visual Basic.


**Figure 4.15 :** The connector on the project box

The flexi force sensor is flexible can be attached any place. To ensure the force is detected with accuracy, the best position to place the flexi force sensor is attached at the gripper jaw in Figure 4.16.

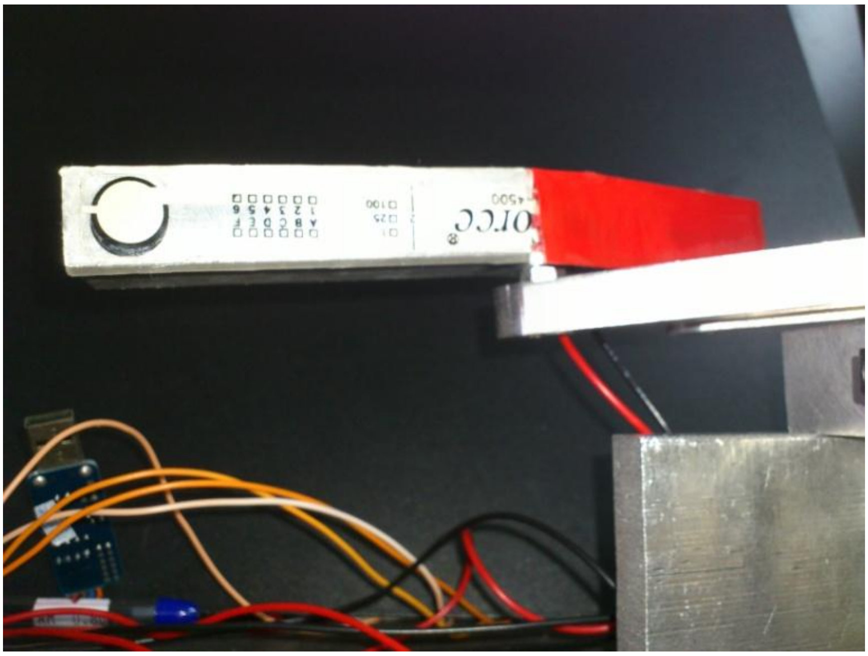


**Figure 4.16 :** The position to attach the flexi force sensor

In Figure 4.17 on the next page show the Sharp infrared distance sensor was placed at the gripper base, the reason distance sensor place at there is the gripper base is the less motion part. Another reason is to give a space for distance sensor to detect the object while gripping, since this sensor only can detect the range from 4-30cm. This sensor also able to detect the slipping when the object slips down.

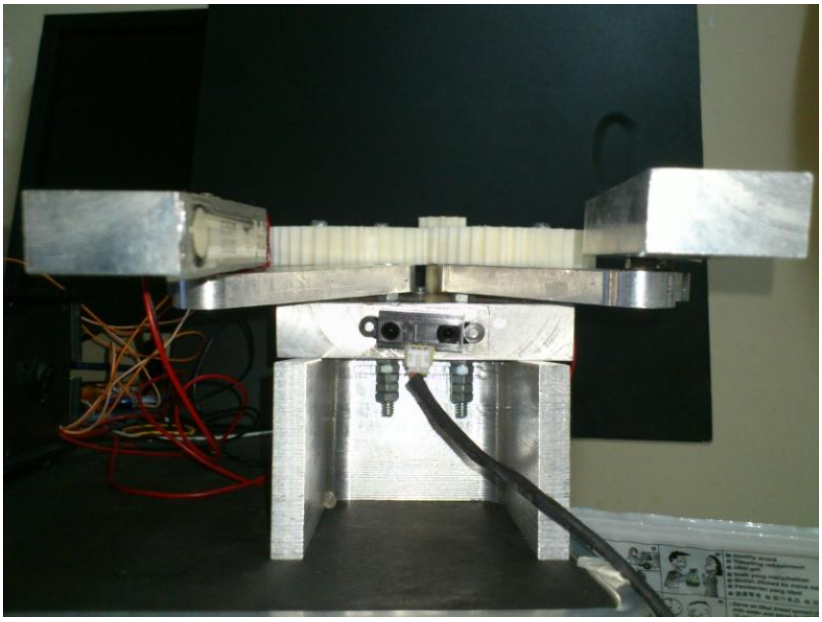


**Figure 4.17 :** The position to place sharp analog infrared distance sensor

The last part of electronics is the button to control the automated gripper was shown in Figure 4.18. It consists of a reset button, manually grip, manually release, set type of material, set the maximum distance and set the desire force to grip button.

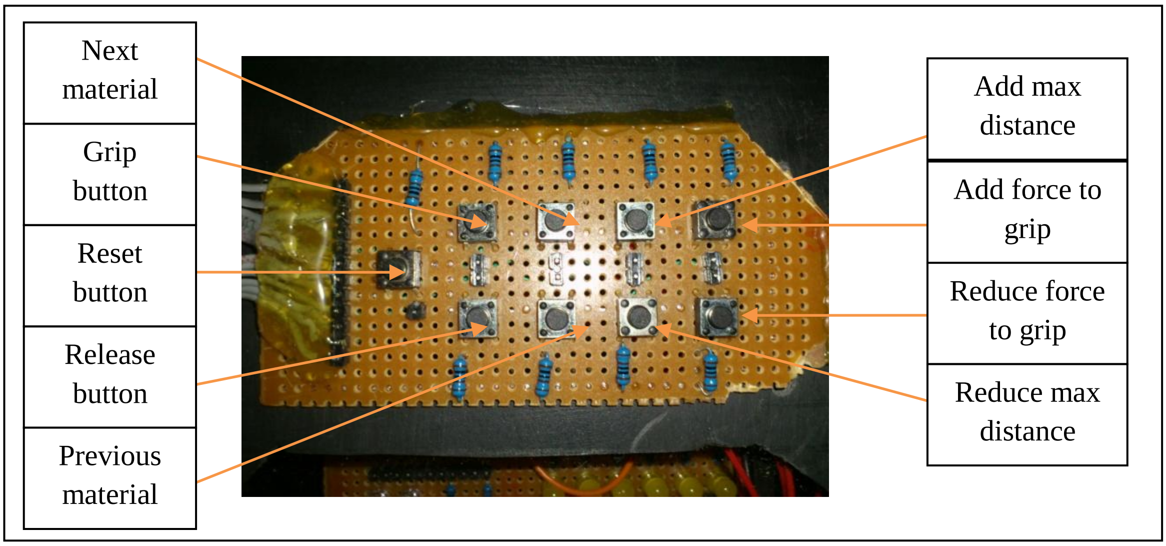



**Figure 4.18 :** The button for automated gripper

## 4.4 PROGRAMMING COMPILE CCS COMPILER

In previous section has mentioned about the electrical component has used at the automated gripper. That includes the sensor, DC motor, LCD screen and etc. This all the electronic component needs a controller to control the automated gripper. The main controller for automated gripper is PIC18F4550 microcontroller, the microcontroller can be programmed by using CCS Compiler. To start compiling the program is creating a new then source file. In Figure 4.19 was shown the layout and source code programming in CCS Compiler for automated gripper.

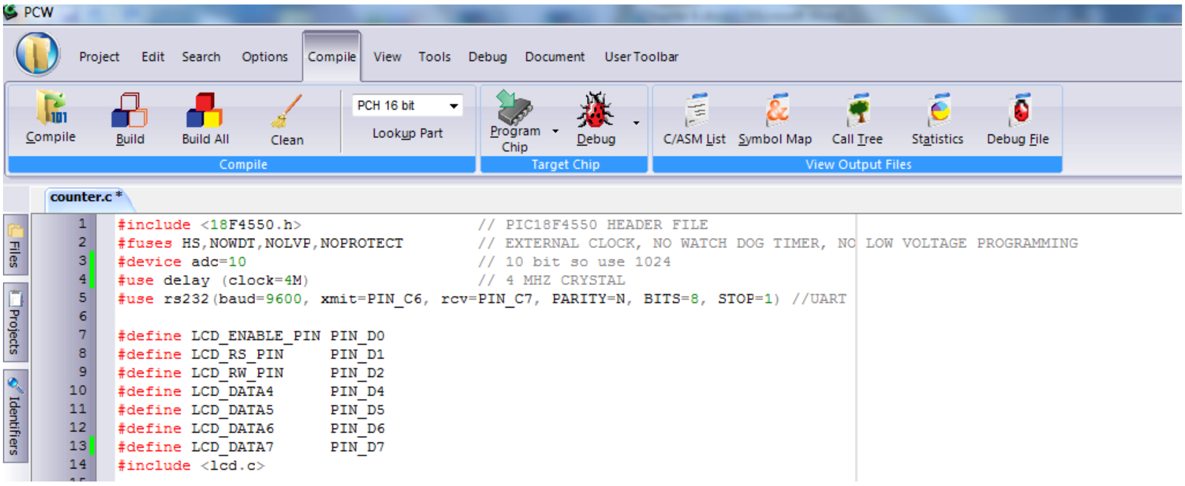



**Figure 4.19 :** The layout and source for CCS Compiler

The source code is written for automated gripper will discuss on here, the 1$^{st}$ header is setting the type of PIC using at the microcontroller. Device adc=10 is the meaning of 10bit use at 1024mv, it is used to convert the analog value to digital. Use delay to set the external clock based on crystal has been used. Use rs232 is used at UART to transmit and receive the data, xmit=PIN_C6 mean receive the data in port C6. Rcy=PIN_C7 mean transmitting the data from port C7. BITS=8 and STOP=1 is show everytimes sent the data in 8bit will stop. The baud rate for standard USB is 9600, so set baud=800. The next paragraph shows the standard coding for LCD screens.

The next stages are defined the name in each pin at microcontroller, that include button, grip, ungrip and limit switch. The purpose to defind the name in each command to make programmer easily to know each pin are belong to which pin in Figure 4.20. Void main () is indicated that the main function receives no data from the operating system. The next defined all local variable using in automated gripper.

```
#define BUTTON1        PIN_A2       // ADD      FORCE
#define BUTTON2        PIN_A3       // REDUCE   FORCE
#define BUTTON3        PIN_A4       // ADD      DISTANCE
#define BUTTON4        PIN_A5       // REDUCE   DISTANCE
#define BUTTON5        PIN_C0       // NEXT     TYPE MATERIAL
#define BUTTON6        PIN_C1       // PREVIOUS TYPE MATERIAL
#define GRIP           PIN_E0       // GRIP     BUTTON
#define UNGRIP         PIN_E1       // RELEASE  BUTTON
#define LMTSWT         PIN_E2       // LIMIT    SWITCH BUTTON

void main()
{

int counter;                        // INTEGER FORCE
int blinking;                       // INTEGER TYPE MATERIAL
int LVL2;                           // INTEGER MAX DISTANCE
int i;                              // INTEGER I FOR LOOP
int value3;                         // INTEGER TO COMPARISON DESIRE FORCE WITH SENSOR FORCE
int32 AI1;                          // VARIABLE INTEGER FROM ANALOG PIN_A0 DISTANCE SENSOR
int32 AI2;                          // VARIABLE INTEGER FROM ANALOG PIN_A1 FORCE SENSOR
int32 LVL;                          // VARIABLE INTEGER FOR FORMULA DISTANCE SENSOR
float total;                        // FLOAT FOR LOOP FOR FORCE SENSOR
double value2;                      // DOUBLE FLOAT FOR FORMULA FORCE SENSOR
```

**Figure 4.20 :** The coding to define the button and local variable

Port initialization is necessary to defend hardware setup, that include the enable the analog function in microcontroller from port A0 until A1. PIC 18F4550 microcontroller has 40 pins that include port A, B, C, D and E. Port A is able to use the analog value, because in port A has implanted the analog to digital (ADC) circuit. For port D is designed to show the data on LCD screen.

Basically this program consists 2 loop program one is for global system for overall program and R232 serial data to connect with USB UART, it is enable_interrups (int_rda) to enable the interrupt in R232 serial data which is used at USB UART. Another is enable_interrups (global) to enable the interrupt in global system which is meant all the program still continue to run even interrupt.

The next stage is the main program for a microcontroller, the function while (TRUE) is mean always loop the program. So it will always always perform the task and print out the data to appear on LCD screen in Figure 4.21.

```
///////////////////////////////////////////////////////////////////////////////
//                               PORT INITIALIZE                            //
///////////////////////////////////////////////////////////////////////////////
setup_adc(ADC_CLOCK_INTERNAL);     // ENABLE FUNCTION ADC
setup_adc_ports(AN0_TO_AN1);       // SET PORT FROM PORT A0 UNTIL A1 AS ANALOG PORT
set_tris_b(0x0F);                  // SET ALL PORT B TO INPUT & OUTPUT IN HEX FORMAT
                                   // ALL THE LED WILL ALWAYS ON
output_b(0xFF);                    // RESET ALL PORT B WHILE RUN PROGRAM
set_tris_c(0xFF);                  // SET ALL PORT C ALWAYS IN INPUT CONDITION
set_tris_e(0xFF);                  // SET ALL PORT E ALWAYS IN INPUT CONDITION
output_e(0xFF);                    // RESET ALL PORT E WHILE RUN PROGRAM
lcd_init();                        // INITIALIZE THE LCD PROGRAM DATA
enable_interrupts(int_rda);        // ALL INTERRUPTS ON FOR R232 UART
enable_interrupts(global);         // ALL SYSTEM INTERRUPS ON FOR GLOBAL SYS
counter =0;                        // RESET COUNTER FOR FORCE
blinking =0;                       // RESET COUNTER FOR TYPE MATERIAL
LVL2 =0;                           // RESET COUNTER FOR MAX DISTANCE
///////////////////////////////////////////////////////////////////////////////
//                                 MAIN PROGRAM                             //
///////////////////////////////////////////////////////////////////////////////
while(TRUE)                        // ALWAYS LOOP THE PROGRAM
{
printf("\fOBJ= %u     TYP= %u     ",LVL,(int)blinking);// PRINT UART DATA
printf("\nF= %d     ST= %u     D= %u     ",value3,(int)counter,(int)LVL2);
delay_ms(100);                     // DUTY TIME TO PERFORM PREVIOUS PROGRAM
```

**Figure 4.21 :** Port initialization and main program for automated gripper

The microcontroller for PIC18F4550 able to convert the analog value to a digital value in Figure 4.22, both of the sensor has been used at automated gripper is analog sensor. For Sharp infrared distance sensor, it reads the value from channel 0. Then convert the data then print out on the LCD screen in Figure 4.23. Another analog sensor is force sensor it reads the analog value from channel 1 then for loop at 100 times and average it. The reason to do it is stabilized the analog force sensor data.

```
// SHARP INFRA RED DISTANCE SENSOR
{
set_adc_channel(0);                 // POINTS ADC AT CHANNEL 0
delay_ms(100);                      // DELAY TO GET ANALOG VALUE
AI1=read_adc();                     // READ ANALOG INPUT 1

LVL=(AI1*500000)/1024;              // SCALLING DISTANCE VALUE
LVL=(LVL-19000)/2099;               // FORMULA FROM SN-GP2Y0A21 (10cm-80cm)
LVL=1000/LVL;                       // APPLY IN SN-GP2D120XJ00F (4-30cm)
LVL=LVL/2.692307692;                // (80cm-30cm)/(10cm-4cm)=2.692307692
LVL=LVL+2;
printf(lcd_putc,"\fOBJ=%uCM TYP=%u",LVL,(int)blinking);// DISPLAY TEMP VALUE AT LCD DISPLAY
printf(lcd_putc, "\nF=%dN ST=%uN D=%uCM",value3,(int)counter,(int)LVL2);
}
```

```
//FORCE SENSOR
{
set_adc_channel(1);                 // POINTS ADC AT CHANNEL 1
delay_ms(100);                      // DELAY TO GET ANALOG VALUE
AI2=read_adc();                     // READ ANALOG INPUT 2
value2=0;                           // RESET INTEGER VALUE 2;
total=0;                            // RESET FLOAT TOTAL VALUE;
i=0;                                // RESET I INTEGER FOR LOOP;

for (i=0; i<100; i++);                      // FOR LOOP 100 TIMES;
  {
    value2 = AI2;                  // READ THE VALUE FROM ANALOG PIN FOR 100 TIMES
    total += value2 * (5.0/1024)*1000;// 5V DEVIDE ADC 10 BIT AND MULTIPLE 1000mV
  }
value2 =(total/100);                // TO GET AVERAGE VALUE FROM 100 LOOP
value2= (value2/10);                // TO COVERT DATA TO KG
value2= (value2*9.81);              // TO CONVERT KGTO NEWTON
value3=value2;                      // TO DEFINE VALUE 3 TO AVOID INTERRUPT DATA
}
```

**Figure 4.22 :** Formula for Sharp infrared distance sensor and flexi force sensor


**Figure 4.23 :** Data appear on LCD screen

The program in automated gripper includes the counter to adjust the value such as desire force to grip, the maximum distance to operate and selection type of material in Figure 4.24. All was used the counter function to count up and count down the value. Each time the button is pressed, the LED will indicate which value has been counted. Besides the function While (! Input (Button1)) at here is to show the each time the button has a press, only will trigger 1 time at microcontroller.

```
if (!input(BUTTON1))                      // WHEN PRESS THE BUTTON1
{
while(!input(BUTTON1))                    // TRIGGER FOR 1 TIME
{}
blinking=0;                               // TYPE MATERIAL IS RESET
counter++;                                // ADD INTEGER FORCE
   {
   output_low(PIN_B2);                    // INDICATE AT LED
   delay_ms(200);
   output_high(PIN_B2);
   }
}

else if(!input(BUTTON2))                  //WHEN PRESS THE BUTTON 2
{
while(!input(BUTTON2))                    //TRIGGER FOR 1 TIME
{}
blinking=0;                               // TYEP MATERIAL IS RESET
counter--;                                // REDUCE INTEGER FORCE
   {
   output_low(PIN_B3);                    // INDICATE AT LED
   delay_ms(200);
   output_high(PIN_B3);
   }
}
```

**Figure 4.24 :** Function counter to count up and down the value.

Besides the compare function also use at automated gripper, there have 2 comparison which are desired force has been set by user with reading force sensor. Another one is a distance sensor compare with reading distance sensor. The program will analysis to sent out the output to grip in rotation clockwise or release in a counter clockwise as shown in Figure 4.25 on the next page.

```
if (counter==value3)          //WHILE COUNTER IN FORCE IS SAME TO SENSOR READING FORCE
{                             //2 COMMAND WILL SENT AT THE SAME TIME TO HOLD THE OBJ
   output_high(PIN_B0);       //PIN B0 FOR ROTATION CLOCKWISE
   output_high(PIN_B1);       //PIN B1 FOR ROTATION COUNTER CLOCKWISE
}

else if(blinking==1)           //SELECTION TYPE 1 MATERIAL IS 6 NEWTON TO GRIP
{
   counter=6;                  //DESIRE FORCE SET TO GRIP 200N LOAD
}

else if(blinking==2)           //SELECTION TYPE 2 MATERIAL IS 4 NEWTON TO GRIP
{
   counter=4;                  //DESIRE FORCE SET TO GRIP DRINK CAN

   if (counter<value3)          //WHILE THE COUNTER IN FORCE IS LESS THAN SENSOR
   {                            //-READING FORCE WILL PERFORM BELOW COMMAND
      if (LVL<=LVL2)            //WHILE THE MAXIMUM DISTANCE IS LESS OR SAME WITH
      {                         //-READING DISTANCE SENSOR, THE GRIPPER WILL OPERATE
      output_low(PIN_B0);
      output_high(PIN_B1);      //PIN B1 FOR ROTATION COUNTERCLOCKWISE OPERATE
      }
      else                      // ELSE GRIPPER NO MOVEMENT
      {
      output_high(PIN_B0);
      output_high(PIN_B1);
      }
   }

   if (counter>value3)          //WHILE THE COUNTER IN FORCE IS MORE THAN SENSOR
   {                            //-READING FORCE WILL PERFORM BELOW COMMAND
      if (LVL<=LVL2)            //WHILE THE MAXIMUM DISTANCE IS MORE OR SAME WITH
      {                         //-READING DISTANCE SENSOR, THE GRIPPER WILL OPERATE
      output_high(PIN_B0);      //PIN B0 FOR ROTATION CLOCKWISE OPERATE
      output_low(PIN_B1);
      }
      else                      // ELSE GRIPPER NO MOVEMENT
      {
     output_high(PIN_B0);
      output_high(PIN_B1);
      }
   }
```

**Figure 4.25 :** Function compares at automated gripper

The last stage for automated gripper is the button to grip and release. This button can operate the automated gripper in the manual mode in Figure 4.26 on the next page. When gripping button is pressed, the motor will rotate counterclockwise. For release button has additional limit switch to avoid over the release at automated gripper. So when the release button is pressed, it will release until the touch the limit switch. Then the automated gripper will automatic stop.

```
if (!input(GRIP))            //WHILE PRESS BUTTON GRIP
{
   output_low(PIN_B1);
   output_high(PIN_B0);      //PIN B1 FOR ROTATION COUNTERCLOCKWISE OPERATE
}

if (!input(UNGRIP))          //WHILE PRESS BUTTON RELEASE SENT OUTPUT B0
{
   if (!input(LMTSWT))       //UNTIL LIMIT SWITCH HAS TOUCH
   {
      output_high(PIN_B0);
      output_high(PIN_B1);
   }
   else
   {
   output_low(PIN_B0);       //PIN B0 FOR ROTATION CLOCKWISE OPERATE
   }
}
```

**Figure 4.26 :** Manually function button to grip and release

## 4.5 VISUAL BASIC GRAPHIC USER INTERFACE (GUI)

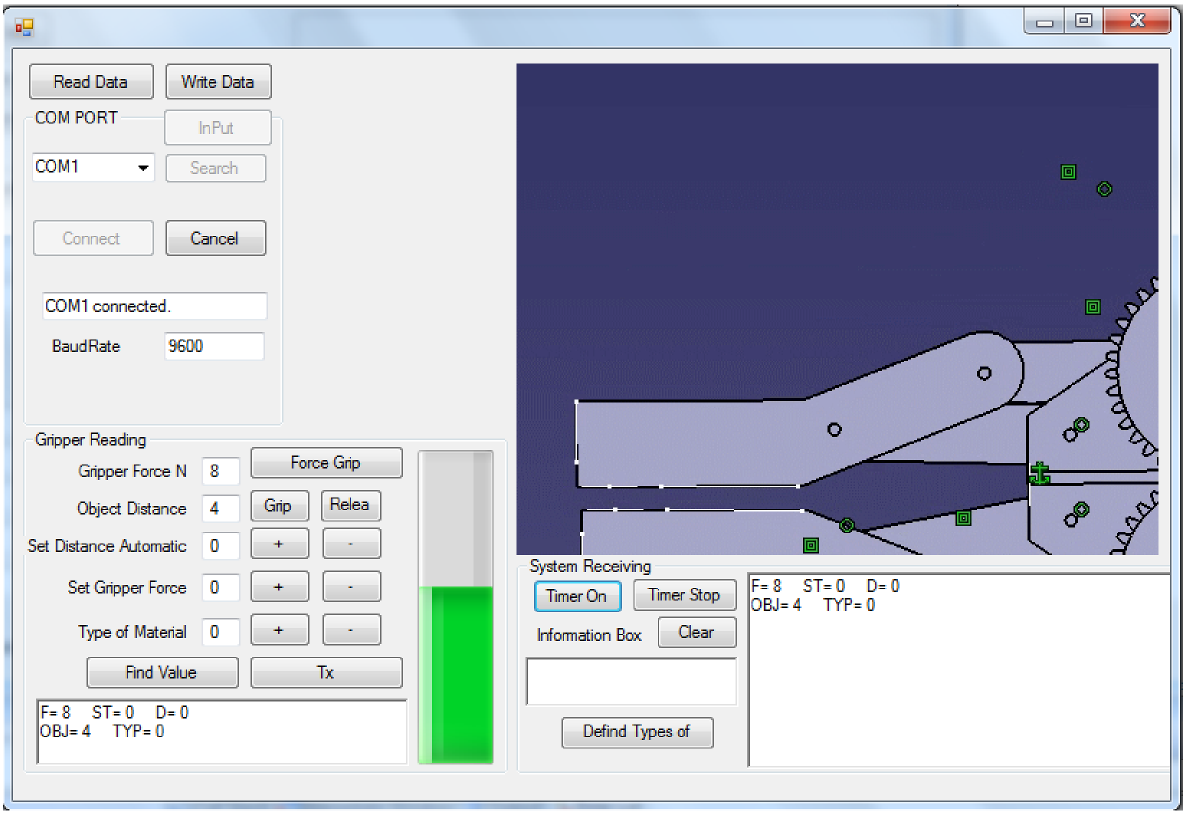



**Figure 4.27 :** The interface for automated gripper

In order to make the user interface of the Automated Gripper to be as user friendly as possible. There has a simple feature in the interface so that user does not need to key in the data in the configuration data. In Figure 4.27 on the previous page shows the GUI when user click UART.exe.

To create the visual basic form it must consist a header, the header from line 1 until line 4 is used in to read and write the data to the com port has been selected. The function inpout32.dll is used to control the parallel port was shown in Figure 4.28.

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.IO.Ports

Public Class Form1
    Public Declare Function Inp Lib "inpout32.dll" Alias "Inp32" _
(ByVal PortAddress As Integer) _
As Integer
    Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
    (ByVal PortAddress As Integer, _
    ByVal Value As Integer)

    Dim rxbuff As String    'Buffer for receievd data(x fr SerialPort_Test Vb)
    Dim indata2 As String

    Dim myPort As Array  'COM Ports detected on the system will be stored here & Com ports enumerated into here
    Delegate Sub SetTextCallback(ByVal [text] As String) 'Added to prevent threading errors during receiveing of data

    'define com port
    Dim WithEvents SerialPort As New IO.Ports.SerialPort
    Dim transmitt As Integer

    'SerialPort1 auto searching when panel loading
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        For i As Integer = 0 To My.Computer.Ports.SerialPortNames.Count - 1
            ComboBox1.Items.Add(My.Computer.Ports.SerialPortNames(i))
        Next
        Button9.Enabled = False
    End Sub
```

**Figure 4.28 :** Source code Visual Basic Header

In Figure 4.29 on the next page com port show the selected port which has been connected, parity bit and the number of data bits, number of stop bits, handshake and com port status. User only needs to click read data button and select which com to connect with GUI.

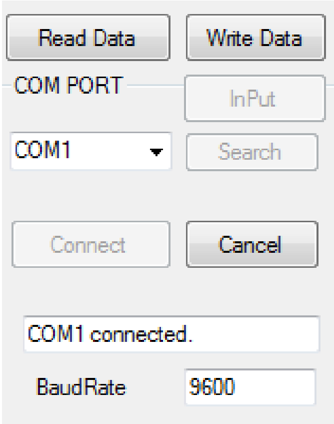



**Figure 4.29 :** The status while connects with UART and microcontroller

All of these parameters have been set up in source code Figure 4.30. based on the standard UART setting. The GUI will show the com port status and the baud rate use to connect the UART to PIC18F4550 microcontroller.

```
Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button8.Click
    Button7.Enabled = False
    Button8.Enabled = False
    Button9.Enabled = True
    If SerialPort1.IsOpen Then
        SerialPort1.Close()
    End If
    Try
        'Set remaining port attributes
        With SerialPort1
            'Set SerialPort1 portname to first available port
            .PortName = ComboBox1.Text()
            .BaudRate = 9600
            .Parity = IO.Ports.Parity.None
            .DataBits = 8
            .StopBits = IO.Ports.StopBits.One
            .Handshake = Handshake.None '(x)
        End With

        TextBox2.Text = SerialPort1.BaudRate
        SerialPort1.Open()
        TextBox1.Text = ComboBox1.Text & " connected."
        transmitt = 1
    Catch ex As Exception
        MsgBox("Please select COM PORTS")
    End Try
End Sub
```

**Figure 4.30 :** The source code to select the com port

In Figure 4.31 is a system receiving use to receive the data from the microcontroller, the interface will show data based on data sent out from a microcontroller through the UART to visual basic GUI interface.

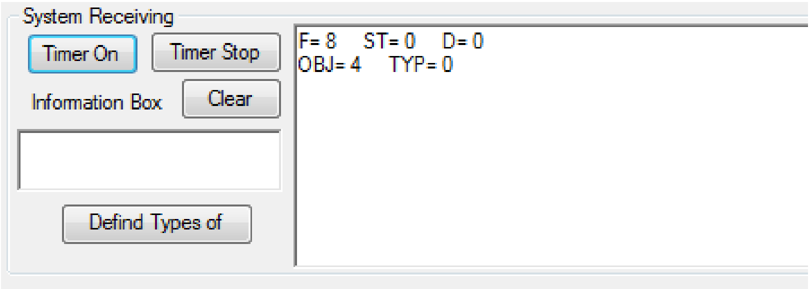



**Figure 4.31 :** System receives the information from UART

A system receiving its source code was shown in Figure 4.32. The command rtb. Received. InvokeRequired is a function used to request the data for to read, when receiving the data from the UART.

```
Private Sub ReceivedText(ByVal [text] As String)

    'compares the ID of the creating Thread to the ID of the calling Thread
    If Me.rtbReceived.InvokeRequired Then
        Dim x As New SetTextCallback(AddressOf ReceivedText)
        Me.Invoke(x, New Object() {(text)})

    Else
        Me.rtbReceived.Text &= [text]

    End If
End Sub
```

**Figure 4.32 :** The source code for system receiving

In Figure 4.33 on the next page show data have to rearrange to each text box, the reason rearranges the data is make the data able to read by another program such as a vertical bar on the right side. At the same time the user also can read the data to understand in a short period.

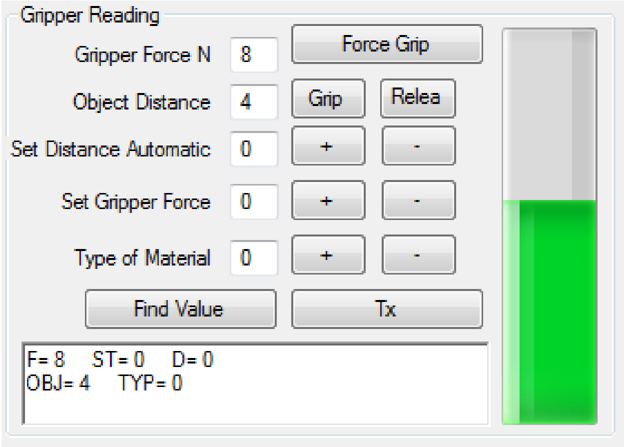



**Figure 4.33 :** The information reading for automated gripper

The source code to rearrange the data can be found in Figure 4.34.at there has shown the way to rearrange the data, this data can read in string format by type Microsoft. VisualBasic. Mid ( the text box to read, start position, end position).

```
Private Sub Findvalue_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Findvalue.Click
    Dim text1 As String
    text1 = rxbuff
    TextBoxForce.Text = Microsoft.VisualBasic.Mid(text1, 3, 3)
    TextBoxSetForce.Text = Microsoft.VisualBasic.Mid(text1, 13, 14)
    TextBoxSetDistance.Text = Microsoft.VisualBasic.Mid(text1, 22, 23)
    TextBoxobject.Text = Microsoft.VisualBasic.Mid(text1, 34, 35)
    TextBoxTypes.Text = Microsoft.VisualBasic.Right(text1, 7)
End Sub
```

**Figure 4.34 :** The source code to rearrange the data

When the grip or release button was clicked, the GUI will show the condition of grippers in Figure 4.35 on the next page. At the same times will send the output through the parallel port to the microcontroller.

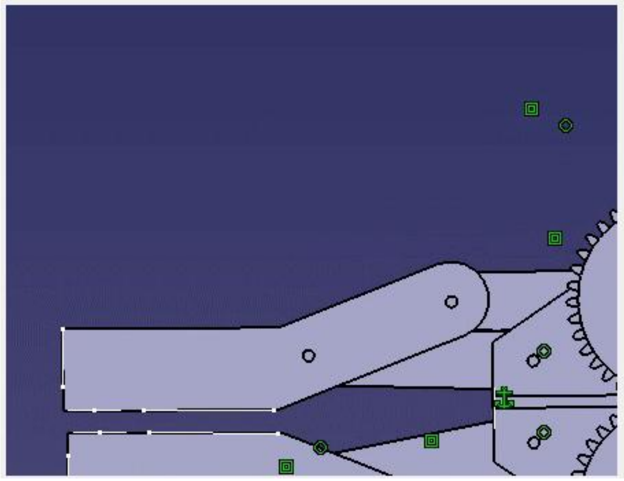


**Figure 4.35 :** The condition for automated gripper

In Figure 4.36 show the source code to show the image gripper in GIF format, the show the image at GUI must insert the location. The button to grip and release can refer to Figure 4.36, each button has its own source code. The command Out (888,255) is showing the all ports is sent 5V. The command Inp (889,16) is used to receive the input

```
Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button16.Click
    Dim fileLocation As String
    fileLocation = System.Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
    fileLocation += "\Automated Gripper\Automated Gripper\WindowsApplication1\WindowsApplication1\Resources\Grip.gif"

    PictureBox1.ImageLocation = fileLocation

    Out(888, 254)
    System.Threading.Thread.Sleep(500)
    Out(888, 255)
    System.Threading.Thread.Sleep(500)
End Sub
```

**Figure 4.36 :** The source code for image gripper and the output command

The information box in Figure 4.37 on the next page is used to show the data based what type of material has been selected in the microcontroller. That can show the information with clear for the user, it this way can prevent select wrong material to grip.

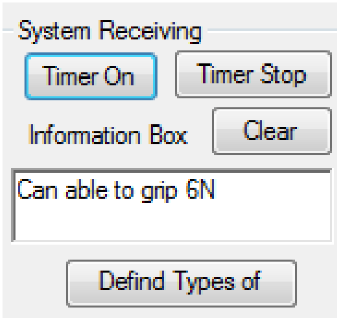



**Figure 4.37 :** The information box

The source in Figure 4.38 has shown the way to appear the data in the information box. The function using at here is case, this case function able to read the data from type material textbox. If the condition is in the case, it will show the information in rich text box.

```
Private Sub Informationtype_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Informationtype.Click
    Dim types As Integer
    types = TextBoxTypes.Text
    Select Case types
        Case "1"
            RichTextBox2.Text = "Can able to grip 6N"
        Case "2"
            RichTextBox2.Text = "200g Load able to grip in 4N"
        Case "3"
            RichTextBox2.Text = "Egg able to grip in 3N"
    End Select
End Sub
```

**Figure 4.38 :** The source code for key information

## 4.6 AN EXPERIMENT FOR AUTOMATED GRIPPER DISTANCE SENSOR

| Time (s) | Distance (cm) |
|---|---|
| 1 | 26 |
| 2 | 18 |
| 3 | 13 |
| 4 | 8 |
| 5 | 5 |
| 6 | 5 |

**Table 4.2 :** Relationship of Distance Sensor and Time

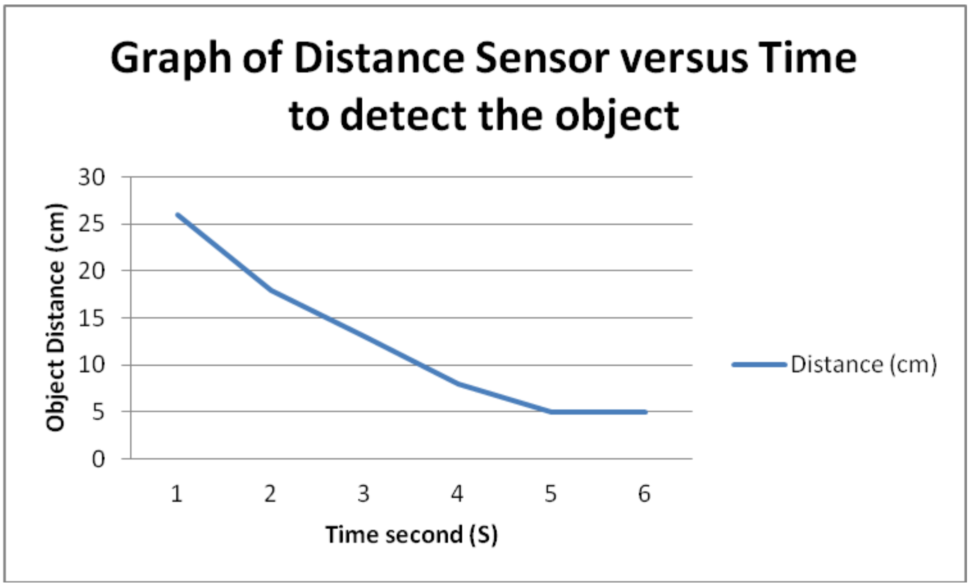



**Figure 4.39 :** The Graph for Distance Sensor versus Time

The control system microcontroller has been implemented in automated gripper, the program using in control system automated include comparison and counter function. Three main electronic components are the distance detection force sensor and microcontroller. The distance sensor was attached at the gripper base to enhance the stability detection object.

The result was shown in Figure 4.39 is the measurement to detect the distance object. Although still not optimal, but the distance sensor already can detect the object accurate and detect the appearance of objects in front of the sensor. That will increase the capability for grasping a moving object. Once the object has been grasped, it will grip firmly without slip.

## 4.7 AN EXPERIMENT FOR AUTOMATED GRIPPER FORCE SENSOR

| Time (s) | Force Sensor (Newton) |
|---|---|
| 1 | 0 |
| 2 | 1 |
| 3 | 2 |
| 4 | 2 |
| 5 | 3 |
| 6 | 4 |
| 7 | 5 |
| 8 | 6 |
| 9 | 5 |
| 10 | 6 |

**Table 4.3 :** Relationship of Force Sensor and Time

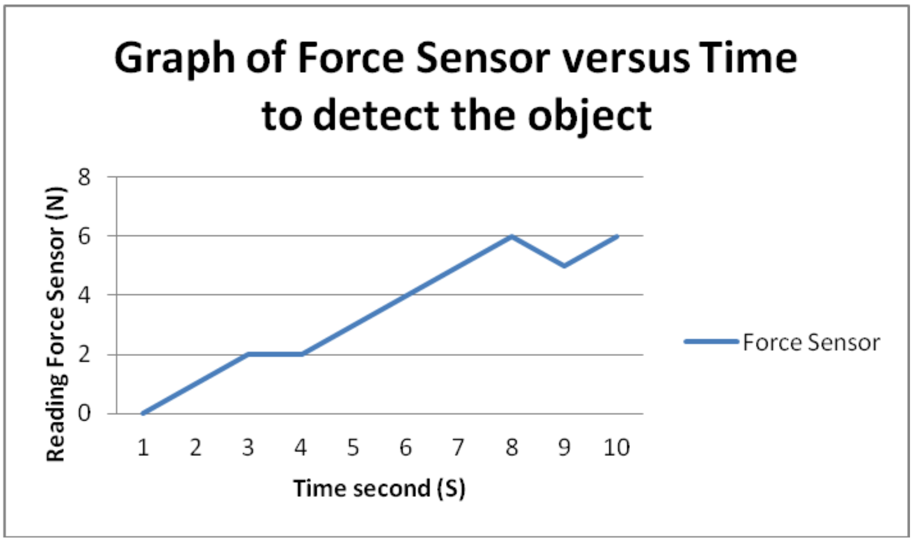



**Figure 4.40:** The Graph measurement to detect the applied force on the object

This experiment is used to detect the force being applied to the object and the result was shown in Figure 4.40. The force sensor was attached at gripper jaw will increase the accuracy to detect the force. In Figure 4.41 show the object is used to analysis gripping force.

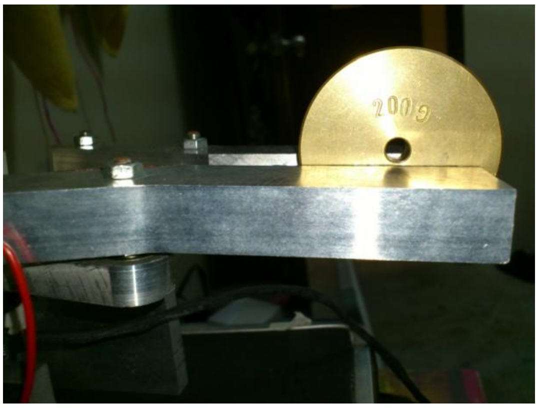


Figure 4.41: The object use to analysis the gripping force.

The result was shown in Figure 4.40 is not stable, that may affect by the vibration of the automated gripper. But the gripper still can function properly because the range for unstable is small so it would not affect the result to grip.

Figure 4.42: The can applied over gripping force

In Figure 4.42 show the can are applied over gripping force, this case will happen when the desire force set by the user is overload. Thus this can will deformation, when the gripper tries to grip the object.


Figure 4.43: The can applied suitable gripping force

When the user sets the suitable desire force, the gripping force was applied to the can would not deform in Figure 4.43. The reason is when the force sensor detects overload, the gripper will release to reduce the gripping force by this way can avoid the deformation.

Design and development automated gripper control for robotic arm was successed, it can be used to grip different type of object based on the force has been set by the user. The Graphical User Interface has been designed also can clearly show all the information, this makes user easy to control the automated gripper.

# CHAPTER 5

# CONCLUSION

## 5.1 CONCLUSION

As a conclusion, the aim and objective of this project has been achieved. A control system for an automated gripper has been developed with PIC18F4550 microcontroller. At the same time, the control system can be linked with user friendly and simple Graphical User Interface (GUI) using Visual Basic 2008. This eases the user to control the automated gripper to grip different type of object and control well in gripping force.

In addition, this automated gripper also consists sharp infrared distance sensor and flexi force sensor. A sharp infrared distance sensor is used to detect the distance object to increase the possibility chance to success grip the object. Another flexi force sensor is used to enhance detection gripping force to avoid the deformation object. The algorithm programming was applied in a control system is function comparison and counter, thus the sensor will follow the desire force has been set by user perform the task.

The electronic part was designed by using the Protues simulation software, that make the automated gripper is more reliable. That believe will greatly reduce the accident happened and unstable condition when the automated gripper perform the task. This automated gripper was designed by using CATIA and analysis the structure by Finite Element Method (FEM). Then almost all the part it was made in aluminium by using the CADCAM Makino milling machine. That will increase the precision and reliability of automated gripper work well in any condition, believe that automated gripper can contribute in the future.

## 5.2 PROBLEMS

Although the controller can function well as expected, but the performance is slightly sluggish, it may take a few second to react properly especially grip the object. This problem can overcome to replace more high speed and high torque DC motor.

Besides that, the UART connection between the visual basic GUI and microcontroller are unstable. Sometimes it has a problem to select the COM port, this problem was happening because of fundamental visual basic programmer is not enough of experience and knowledge. If the problem has solved, the parallel port R232 is not necessary to use control the button to count up and count down, because the data can transmit without interrupt.

The detection for flexi force sensor is unstable, that may affect by disturbance from the analog signal. Need to explore more in flexi force sensor, this kind of force sensor mostly used in Arduino platform, thus hard to find the coding for a microcontroller.

## 5.3 RECOMMENDATION

The performance of the system is quite satisfying. For future works, has some recommendations all has been listed based on the problems and suggestion can improve the performance.

i. To increase the DC motor speed and enhance the accurate for gripping force, mathematical modelling such as PID controller can be implemented in control system to reduce the time to grip and control well in gripping force.

ii. Implement the image processing to easily find out the gripping point automatically, this function can enhance the reliability to grip more difficult shape and avoid the deformation.

iii. Implement additional electronics circuit to improve the stability of flexi force sensor output, it is important when grip the brittle object.

iv. Arduino instead of microcontroller as a controller system for automated gripper, because this Arduino platform provided a more convenient way in programming. The user only needs to key in the series sensor, without compile a complicate programming. Thus this makes programming easier and more user friendly.

## REFERENCE

J.D.TEDFORD, 1991. Design of a Robot Gripper with Force Feedback Control. University of Auckland, Auckland, New Zealand.

Gab Soon Kim, 2007. Development of a Three Axis Gripper Force Sensor and The Intelligent Gripper Using It. Gyeongsang National University.

Ashish Dutta and Goro Obinata, 2001. Impedance Control of a Robotic Gripper for Cooperation with Humans. Akita Universit.

Chiara Lanni and Marco Ceccarelli. An Optimization Problem Algorithm for Kinematic Design of Mechanisms for Two-Finger Grippers. Laboratory of Robotics and Mechatronics, DiMSAT- University of Cassino.

Redwan Alqasemi, 2007. Design and Construction of a Robotic Gripper for Activities of Daily Living for People with Disabilities. University of South Florida.

Tudor Deaconescu,and Andrea Deaconescu, 2011. Pneumatic Muscle Actuated Gripper. International Multi Conference of Engineers and Computer Scientists.

Robert D. Howe, 1994. Tactile Sensing and Control of Robotic Manipulation. Division of Applied Sciences.

Dain Yong Lin and Shyh Chour Huang, 2009. Design and Control of a Topologu Optimal Compliant Microgripper. National Kaohsiung.

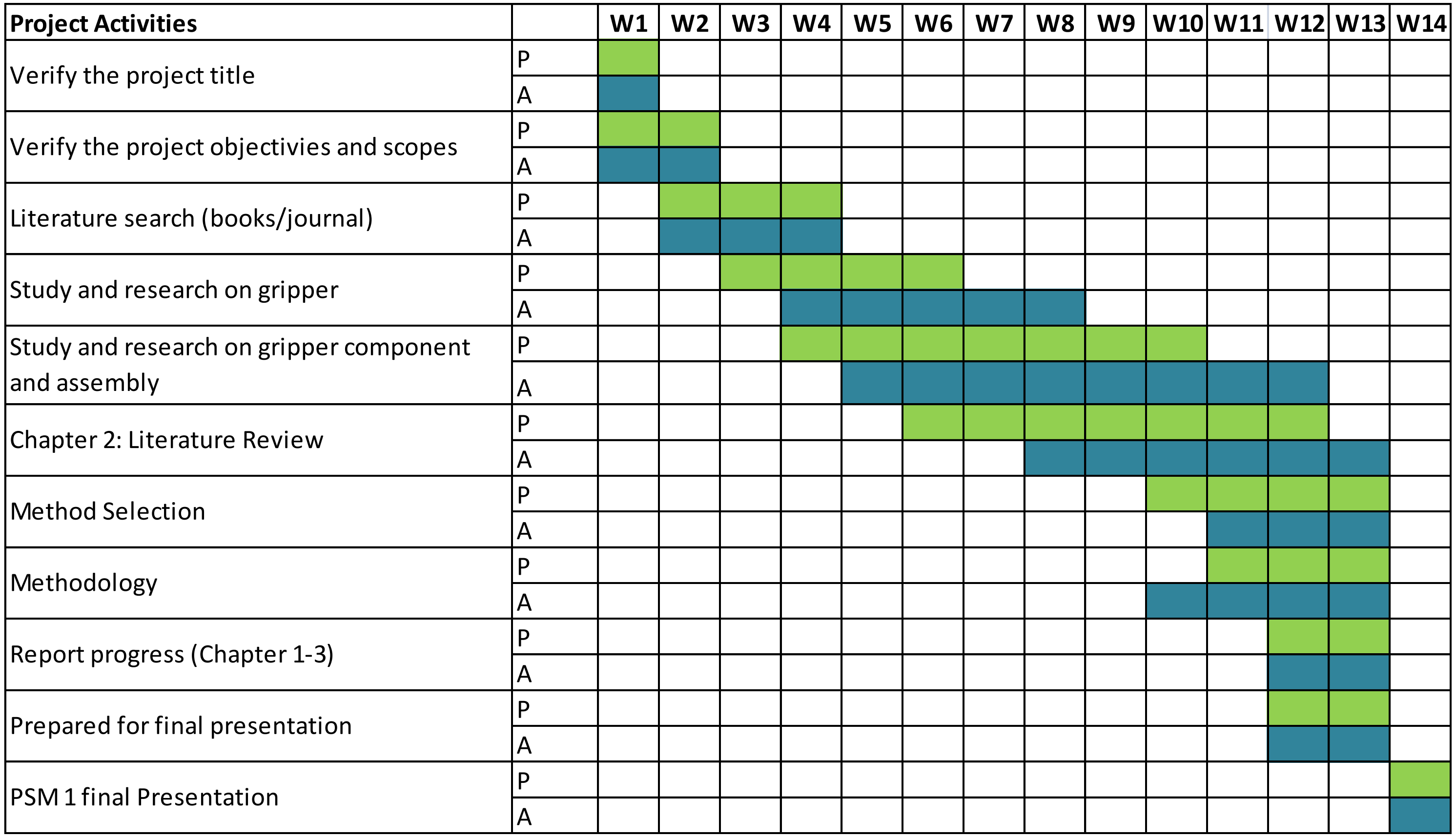
Th. Doersam, Simple Controllers for Multifingered Gripper. University of Karlsruhe. Germany.

M.F. Barsky and D.K. Lindner, 1989. Robot Gripper Control System Using PVDF Piezoelectric Sensors. Virginia Polytechnic Institute.

Richard M. Crowder. An anthropomorphic robotic and effector.University of Southampton.

# APPENDIX A1

## Gantt chart for Final Year Project 1



# APPENDIX A2

## Gantt chart for Final Year Project 2

| Project Activities | | W1 | W2 | W3 | W4 | W5 | W6 | W7 | W8 | W9 | W10 | W11 | W12 | W13 | W14 | W15 | W16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Design the gripper structure | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Select types of sensor and material | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| CATIA Finite Element FEM simulation | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Learning PIC programming CCS Compiler | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Machining for Automated Gripper part | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Rapidprotype for gear through CATIA | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| CATIA CADCAM Machining | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Protues Simulation electrical circuit | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Complete the real circuit | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Assemble each part of Gripper | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Assemble electrical part with Gripper | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Running the programming | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Data Collection/Fabricate the hardware | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Result analysis and modification | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Chapter 4: Result and Discussion | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Chapter 5 conclusion and recommendation | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| Prepared for final presentation | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |
| PSM 2 final Presentation | P | | | | | | | | | | | | | | | | |
| | A | | | | | | | | | | | | | | | | |

# APPENDIX B

## Programming in Microcontroller PIC18F4550 for Automated Gripper

```
#include <18F4550.h>                   // PIC18F4550 HEADER FILE
#fuses HS,NOWDT,NOLVP,NOPROTECT        // EXTERNAL CLOCK, NO WATCH DOG TIMER, NO LOW VOLTAGE PROGRAMMING
#device adc=10                         // 10 bit so use 1024
#use delay (clock=4M)                  // 4 MHZ CRYSTAL
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7, PARITY=N, BITS=8, STOP=1) //UART

#define LCD_ENABLE_PIN PIN_D0
#define LCD_RS_PIN     PIN_D1
#define LCD_RW_PIN     PIN_D2
#define LCD_DATA4      PIN_D4
#define LCD_DATA5      PIN_D5
#define LCD_DATA6      PIN_D6
#define LCD_DATA7      PIN_D7
#include <lcd.c>

#define BUTTON1        PIN_A2      // ADD      FORCE
#define BUTTON2        PIN_A3      // REDUCE   FORCE
#define BUTTON3        PIN_A4      // ADD      DISTANCE
#define BUTTON4        PIN_A5      // REDUCE   DISTANCE
#define BUTTON5        PIN_C0      // NEXT     TYPE MATERIAL
#define BUTTON6        PIN_C1      // PREVIOUS TYPE MATERIAL
#define GRIP           PIN_E0      // GRIP     BUTTON
#define UNGRIP         PIN_E1      // RELEASE  BUTTON
#define LMTSWT         PIN_E2      // LIMIT    SWITCH BUTTON

void main()
{

int counter;                       // INTEGER FORCE
int blinking;                      // INTEGER TYPE MATERIAL
int LVL2;                          // INTEGER MAX DISTANCE
int i;                             // INTEGER I FOR LOOP
int value3;                        // INTEGER TO COMPARISON DESIRE FORCE WITH SENSOR FORCE
int32 AI1;                         // VARIABLE INTEGER FROM ANALOG PIN_A0 DISTANCE SENSOR
int32 AI2;                         // VARIABLE INTEGER FROM ANALOG PIN_A1 FORCE SENSOR
int32 LVL;                         // VARIABLE INTEGER FOR FORMULA DISTANCE SENSOR
float total;                       // FLOAT FOR LOOP FOR FORCE SENSOR
double value2;                     // DOUBLE FLOAT FOR FORMULA FORCE SENSOR
////////////////////////////////////////////////////////////////////////////////
//                               PORT INITIALIZE                             //
////////////////////////////////////////////////////////////////////////////////
setup_adc(ADC_CLOCK_INTERNAL);     // ENABLE FUNCTION ADC
setup_adc_ports(AN0_TO_AN1);       // SET PORT FROM PORT A0 UNTIL A1 AS ANALOG PORT
set_tris_b(0x0F);                  // SET ALL PORT B TO INPUT & OUTPUT IN HEX FORMAT
                                   // ALL THE LED WILL ALWAYS ON
output_b(0xFF);                    // RESET ALL PORT B WHILE RUN PROGRAM
set_tris_c(0xFF);                  // SET ALL PORT C ALWAYS IN INPUT CONDITION
set_tris_e(0xFF);                  // SET ALL PORT E ALWAYS IN INPUT CONDITION
output_e(0xFF);                    // RESET ALL PORT E WHILE RUN PROGRAM
lcd_init();                        // INITIALIZE THE LCD PROGRAM DATA
enable_interrupts(int_rda);        // ALL INTERRUPTS ON FOR R232 UART
enable_interrupts(global);         // ALL SYSTEM INTERRUPS ON FOR GLOBAL SYS
counter =0;                        // RESET COUNTER FOR FORCE
blinking =0;                       // RESET COUNTER FOR TYPE MATERIAL
LVL2 =0;                           // RESET COUNTER FOR MAX DISTANCE
////////////////////////////////////////////////////////////////////////////////
//                                MAIN PROGRAM                               //
////////////////////////////////////////////////////////////////////////////////
while(TRUE)                        // ALWAYS LOOP THE PROGRAM
{
printf("\fOBJ= %u      TYP= %u      ",LVL,(int)blinking);// PRINT UART DATA
printf("\nF= %d      ST= %u      D= %u       ",value3,(int)counter,(int)LVL2);
delay_ms(100);                     // DUTY TIME TO PERFORM PREVIOUS PROGRAM
```

```
// SHARP INFRA RED DISTANCE SENSOR
{
set_adc_channel(0);                     // POINTS ADC AT CHANNEL 0
delay_ms(100);                          // DELAY TO GET ANALOG VALUE
AI1=read_adc();                         // READ ANALOG INPUT 1

LVL=(AI1*500000)/1024;                  // SCALLING DISTANCE VALUE
LVL=(LVL-19000)/2099;                   // FORMULA FROM SN-GP2Y0A21 (10cm-80cm)
LVL=1000/LVL;                           // APPLY IN SN-GP2D120XJ00F (4-30cm)
LVL=LVL/2.692307692;                    // (80cm-30cm)/(10cm-4cm)=2.692307692
LVL=LVL+2;
printf(lcd_putc,"\fOBJ=%uCM TYP=%u",LVL,(int)blinking);// DISPLAY TEMP VALUE AT LCD DISPLAY
printf(lcd_putc, "\nF=%dN ST=%uN D=%uCM",value3,(int)counter,(int)LVL2);
}

//FORCE SENSOR
{
set_adc_channel(1);                     // POINTS ADC AT CHANNEL 1
delay_ms(100);                          // DELAY TO GET ANALOG VALUE
AI2=read_adc();                         // READ ANALOG INPUT 2
value2=0;                               // RESET INTEGER VALUE 2;
total=0;                                // RESET FLOAT TOTAL VALUE;
i=0;                                    // RESET I INTEGER FOR LOOP;

for (i=0; i<100; i++);                          // FOR LOOP 100 TIMES;
  {
    value2 = AI2;                    // READ THE VALUE FROM ANALOG PIN FOR 100 TIMES
    total += value2 * (5.0/1024)*1000;// 5V DEVIDE ADC 10 BIT AND MULTIPLE 1000mV
  }
value2 =(total/100);                    // TO GET AVERAGE VALUE FROM 100 LOOP
value2= (value2/10);                    // TO COVERT DATA TO KG
value2= (value2*9.81);                  // TO CONVERT KG TO NEWTON
value3=value2;                          // TO DEFINE VALUE 3 TO AVOID INTERRUPT DATA
}
if (!input(BUTTON1))                    // WHEN PRESS THE BUTTON1
{
while(!input(BUTTON1))                  // TRIGGER FOR 1 TIME
{}
blinking=0;                             // TYPE MATERIAL IS RESET
counter++;                              // ADD INTEGER FORCE
   {
   output_low(PIN_B2);                  // INDICATE AT LED
   delay_ms(200);
   output_high(PIN_B2);
   }
}

else if(!input(BUTTON2))                //WHEN PRESS THE BUTTON 2
{
while(!input(BUTTON2))                  //TRIGGER FOR 1 TIME
{}
blinking=0;                             // TYEP MATERIAL IS RESET
counter--;                              // REDUCE INTEGER FORCE
   {
   output_low(PIN_B3);                  // INDICATE AT LED
   delay_ms(200);
   output_high(PIN_B3);
   }
}
else if(!input(BUTTON3))          // when press button 2
{
while(!input(BUTTON3))
{}
counter=0;
blinking++;                           // dec counter
   {
   output_low(PIN_B4);
   delay_ms(200);
   output_high(PIN_B4);
   }
}

else if(!input(BUTTON4))          // when press button 2
{
while(!input(BUTTON4))
{}
counter=0;
blinking--;                           // dec counter
   {
   output_low(PIN_B5);
   delay_ms(200);
   output_high(PIN_B5);
   }
}
```

```
else if(!input(BUTTON5))          // when press button 2
{
while(!input(BUTTON5))
{}
LVL2++;                          // dec counter
   {
   output_low(PIN_B6);
   delay_ms(200);
   output_high(PIN_B6);
   }
}

else if(!input(BUTTON6))          // when press button 2
{
while(!input(BUTTON6))
{}
LVL2--;                          // dec counter
   {
   output_low(PIN_B7);
   delay_ms(200);
   output_high(PIN_B7);
   }
}
else if(blinking==1)          //SELECTION TYPE 1 MATERIAL IS 6 NEWTON TO GRIP
{
   counter=6;                 //DESIRE FORCE SET TO GRIP 200N LOAD
}

else if(blinking==2)          //SELECTION TYPE 2 MATERIAL IS 4 NEWTON TO GRIP
{
   counter=4;                 //DESIRE FORCE SET TO GRIP DRINK CAN
}
if (counter<value3)           //WHILE THE COUNTER IN FORCE IS LESS THAN SENSOR
{                             //-READING FORCE WILL PERFORM BELOW COMMAND
   if (LVL<=LVL2)             //WHILE THE MAXIMUM DISTANCE IS LESS OR SAME WITH
   {                          //-READING DISTANCE SENSOR, THE GRIPPER WILL OPERATE
   output_low(PIN_B0);
   output_high(PIN_B1);       //PIN B1 FOR ROTATION COUNTERCLOCKWISE OPERATE
   }
   else                       // ELSE GRIPPER NO MOVEMENT
   {
   output_high(PIN_B0);
   output_high(PIN_B1);
   }
}

if (counter>value3)           //WHILE THE COUNTER IN FORCE IS MORE THAN SENSOR
{                             //-READING FORCE WILL PERFORM BELOW COMMAND
   if (LVL<=LVL2)             //WHILE THE MAXIMUM DISTANCE IS MORE OR SAME WITH
   {                          //-READING DISTANCE SENSOR, THE GRIPPER WILL OPERATE
   output_high(PIN_B0);       //PIN B0 FOR ROTATION CLOCKWISE OPERATE
   output_low(PIN_B1);
   }
   else                       // ELSE GRIPPER NO MOVEMENT
   {
  output_high(PIN_B0);
   output_high(PIN_B1);
   }
}
if (!input(GRIP))               //WHILE PRESS BUTTON GRIP
{
   output_low(PIN_B1);
   output_high(PIN_B0);         //PIN B1 FOR ROTATION COUNTERCLOCKWISE OPERATE
}

if (!input(UNGRIP))             //WHILE PRESS BUTTON RELEASE SENT OUTPUT B0
{
   if (!input(LMTSWT))          //UNTIL LIMIT SWITCH HAS TOUCH
   {
      output_high(PIN_B0);
      output_high(PIN_B1);
   }
   else
   {
   output_low(PIN_B0);          //PIN B0 FOR ROTATION CLOCKWISE OPERATE
   }
}

else
{
}

}
}
```

# APPENDIX C

## Programming Visual Basic Graphical User Interface for Automated Gripper

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.IO.Ports

Public Class Form1
    Public Declare Function Inp Lib "inpout32.dll" Alias "Inp32" _
(ByVal PortAddress As Integer) _
As Integer
    Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" _
    (ByVal PortAddress As Integer, _
    ByVal Value As Integer)

    Dim rxbuff As String    'Buffer for receievd data(x fr SerialPort_Test Vb)
    Dim indata2 As String

    Dim myPort As Array  'COM Ports detected on the system will be stored here & Com
ports enumerated into here
    Delegate Sub SetTextCallback(ByVal [text] As String) 'Added to prevent threading
errors during receiveing of data

    'define com port
    Dim WithEvents SerialPort As New IO.Ports.SerialPort
    Dim transmitt As Integer

    'SerialPort1 auto searching when panel loading
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        For i As Integer = 0 To My.Computer.Ports.SerialPortNames.Count - 1
            ComboBox1.Items.Add(My.Computer.Ports.SerialPortNames(i))
        Next
        Button9.Enabled = False
    End Sub

    Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button10.Click
        Try
            SerialPort1.Close()
            transmitt = 0
```

```
        Catch ex As Exception
        End Try
    End Sub

    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button9.Click
        Button7.Enabled = True
        SerialPort1.Close()           'Close our Serial Port

        Try
            SerialPort1.Close()
            TextBox1.Text = SerialPort1.PortName & " disconnected."
            Button8.Enabled = True
            Button9.Enabled = False
            transmitt = 0

        Catch ex As Exception
            MsgBox("Try again")
        End Try
    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button8.Click
        Button7.Enabled = False
        Button8.Enabled = False
        Button9.Enabled = True
        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
        Try
            'Set remaining port attributes
            With SerialPort1
                'Set SerialPort1 portname to first available port
                .PortName = ComboBox1.Text()
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
                .Handshake = Handshake.None '(x)
            End With

            TextBox2.Text = SerialPort1.BaudRate
            SerialPort1.Open()
            TextBox1.Text = ComboBox1.Text & " connected."
            transmitt = 1
        Catch ex As Exception
            MsgBox("Please select COM PORTS")
        End Try
    End Sub
```

```
    Private Sub SerialPort1_DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
        If Button17.Enabled = True Then
            If SerialPort1.IsOpen Then
                SerialPort1.Close()
            End If
            With SerialPort1
                .Open()
                .Write(TextBox3.Text)
                .Close()
            End With
        Else
            'Automatically called every time a data is received at the serialPort
            ReceivedText(SerialPort1.ReadLine())

            ''This sub gets called automatically when the com port recieves some data
            rxbuff = (SerialPort1.ReadLine)
        End If
    End Sub

    Private Sub ReceivedText(ByVal [text] As String)

        'compares the ID of the creating Thread to the ID of the calling Thread
        If Me.rtbReceived.InvokeRequired Then
            Dim x As New SetTextCallback(AddressOf ReceivedText)
            Me.Invoke(x, New Object() {(text)})

        Else
            Me.rtbReceived.Text &= [text]

        End If
    End Sub

    Private Sub btnSend_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSend.Click

        ''The text contained in the txtText will be sent to the serial port as ascii
        ''plus the carriage return (Enter Key) the carriage return can be ommitted if the
other end does not need it
        ''*Write this data to port

        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
        With SerialPort1
            .Open()
            .WriteLine(TextBox5.Text & vbCr)
            .Close()
```

```
        End With
    End Sub

    Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button15.Click
        Dim returnValue As String
        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
        SerialPort1.Open()
        SerialPort1.ReadLine()
        SerialPort1.Close()
        returnValue = SerialPort1.ReadLine()
        Label6.Text = returnValue
    End Sub

    Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button14.Click
        Dim message As Char
        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
        SerialPort1.Open()
        SerialPort1.ReadLine()
        SerialPort1.Close()

        message = SerialPort1.ReadLine
        Label5.Text = message

    End Sub

    Private Sub Tx_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Tx.Click

        'Else display the recieved data in the RichTextBox
        RichTextBox1.Text = rxbuff
    End Sub
    Private Sub Clear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Clear.Click
        rtbReceived.Clear()
    End Sub
    Private Sub Timeron_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timeron.Click
        Timer1.Start()
    End Sub
    Private Sub Timerstop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timerstop.Click
        Timer1.Stop()
    End Sub
```

```
    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        sub2repeat()
    End Sub
    Private Sub sub2repeat()
        Clear.PerformClick()
        Tx.PerformClick()
        Findvalue.PerformClick()
        Button18.PerformClick()
        Progress.PerformClick()
        Informationtype.PerformClick()
    End Sub

    Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button11.Click
        Dim Text3 As String
        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
        With SerialPort1
            .Open()
            .Write(TextBox3.Text)
            .Close()
        End With
        Text3 = TextBox3.Text
        Label2.Text = Text3
    End Sub

    Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button12.Click
        Dim c As Char
        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
        With SerialPort1
            .Open()
            .ReadChar()
            .Close()
        End With
        Label3.Text = c
    End Sub

    Private Sub Findvalue_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Findvalue.Click
        Dim text1 As String
        text1 = rxbuff
        TextBoxForce.Text = Microsoft.VisualBasic.Mid(text1, 3, 3)
        TextBoxSetForce.Text = Microsoft.VisualBasic.Mid(text1, 13, 14)
        TextBoxSetDistance.Text = Microsoft.VisualBasic.Mid(text1, 22, 23)
```

```
        TextBoxobject.Text = Microsoft.VisualBasic.Mid(text1, 34, 35)
        TextBoxTypes.Text = Microsoft.VisualBasic.Right(text1, 7)
    End Sub

    Private Sub Progress_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Progress.Click
        MTech010VerticalProgessBar1.Value = TextBoxForce.Text
    End Sub

    Private Sub Button17_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button17.Click
        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
        With SerialPort1
            .Open()
            .Write("P")
            .Close()
        End With
    End Sub

    Private Sub Button31_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button31.Click
        Button17.Enabled = True
        SerialPort1.Close()          'Close our Serial Port
        Try
            SerialPort1.Close()
            transmitt = 0

        Catch ex As Exception
            MsgBox("Try again")
        End Try
    End Sub

    Private Sub Button32_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button32.Click
        Button17.Enabled = False
        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
    End Sub

    Private Sub Distanceadd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Distanceadd.Click
        Out(888, 191)
        System.Threading.Thread.Sleep(500)
        Out(888, 255)
        System.Threading.Thread.Sleep(500)
    End Sub
```

```
    Private Sub Distancereduce_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Distancereduce.Click
        Out(888, 127)
        System.Threading.Thread.Sleep(500)
        Out(888, 255)
        System.Threading.Thread.Sleep(500)
    End Sub

    Private Sub Forceadd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Forceadd.Click
        Out(888, 251)
        System.Threading.Thread.Sleep(500)
        Out(888, 255)
        System.Threading.Thread.Sleep(500)
    End Sub

    Private Sub Forcereduce_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Forcereduce.Click
        Out(888, 247)
        System.Threading.Thread.Sleep(500)
        Out(888, 255)
        System.Threading.Thread.Sleep(500)
    End Sub

    Private Sub Materialadd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Materialadd.Click
        Out(888, 239)
        System.Threading.Thread.Sleep(500)
        Out(888, 255)
        System.Threading.Thread.Sleep(500)
    End Sub

    Private Sub Materialreduce_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Materialreduce.Click
        Out(888, 223)
        System.Threading.Thread.Sleep(500)
        Out(888, 255)
        System.Threading.Thread.Sleep(500)
    End Sub

    Private Sub Informationtype_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Informationtype.Click
        Dim types As Integer
        types = TextBoxTypes.Text
        Select Case types
            Case "1"
                RichTextBox2.Text = "Can able to grip 6N"
            Case "2"
```

```
            RichTextBox2.Text = "200g Load able to grip in 4N"
          Case "3"
            RichTextBox2.Text = "Egg able to grip in 3N"
        End Select
    End Sub

    Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button16.Click
        Dim fileLocation As String
        fileLocation =
System.Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
        fileLocation += "\Automated Gripper\Automated
Gripper\WindowsApplication1\WindowsApplication1\Resources\Grip.gif"

        PictureBox1.ImageLocation = fileLocation

        Out(888, 254)
        System.Threading.Thread.Sleep(500)
        Out(888, 255)
        System.Threading.Thread.Sleep(500)
    End Sub

    Private Sub Button19_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button19.Click
        Dim fileLocation As String
        fileLocation =
System.Environment.GetFolderPath(Environment.SpecialFolder.Desktop)
        fileLocation += "\Automated Gripper\Automated
Gripper\WindowsApplication1\WindowsApplication1\Resources\Ungrip.gif"
        PictureBox1.ImageLocation = fileLocation

        Out(888, 253)
        System.Threading.Thread.Sleep(500)
        Out(888, 255)
        System.Threading.Thread.Sleep(500)
    End Sub
End Class
```