

University of Southern Queensland
Faculty of Engineering and Surveying

Wall Climbing Robot

A dissertation submitted by

Kang Boon Geh

in fulfillment of the requirements of

Courses ENG4111 and 4112 Research Project

towards the degree of

Bachelor of Electrical and Electronic Engineering

University of Southern Queensland

Faculty of Engineering and Surveying

**ENG4111 Research Project Part 1 &
ENG4112 Research Project Part 2**

Limitations of Use

The council of the University of Southern Queensland, its Faculty of Engineering and Surveying, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the council of the University of Southern Queensland, its Faculty of Engineering and Surveying or the staff of the University of Southern Queensland.

This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled “ Research Project” is to contribute to the overall education within the student’s chosen degree program. This document, the associated hardware, software, drawings, and other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

Professor Frank Bullen

Dean

Faculty of Engineering and Surveying

CANDIDATES CERTIFICATION

I certify that the ideas, design and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other courses or institution, except where specifically stated.

Kang Boon Geh
Student Number: 0050003192

Signature
31.10.2007

Date

ABSTRACT

This project is to build a wall climbing robot. This robot is a suction type of wall climbing robot; using PIC to do the interfacing, control the movement of the robot, ventilation fan and the sensors. The position sensors are used to sense any things blocking its path while it is climbing up, therefore it can be rotated left, right, up and down. The on-board pump has been replaced by computer fan which will lower the cost as well as the weight of the original product. The cost of building this robot must be cheaper than the commercial robots.

BACKGROUND

In all suction adhesion robot that are used for wall climbing robot, they used onboard pump that created a pressure drop inside the vacuum cups that are pressed against the wall or ceiling. As the vacuum cup are pressed against the wall, the on board pump will start to pump the air in the vacuum cup out to the surrounding. After so time, the pressure inside the vacuum cup is lower than that outside the vacuum cup thus it is able to “stick” onto the wall.

This type of wall climbing robots require a smooth and flat vertical surface, therefore if there is a crack in the wall or the vertical surface is uneven, it will not work. Another problem of this type of robot is that it is much slower than other type of walk climbing robot. As it need time to build up the require pressure to hold itself on to the wall.

The suction adhesion robots are normally used on valley bridges and the temperature could be around 0 to 50 degree Celsius. As for the weight of this type of robot should not be more than 35 kg, thus the body of this robot should be make of non metallic material. The center of gravity of the robot must be kept as close as to the vertical surface as possible, as the force required to hold the whole weight of the robot will be reduced.

Although this kind of robot are normally used to do the cleaning for high buildings and under the high way bridges. This will in turn replaced human beings thus reducing the accident rate. This will farther increase the human being rate of living.

ACKNOWLEDGEMENT

I would like to thank my supervisor, Mr. John Billingsley, for giving me this chance to do this interesting project. Giving me helpful advice and suggestion. The staffs of Microchip and RF Solutions Ltd for their valuable advise on their products. My family members for their understanding of my situation. Cause the whole project are being done fully at home. And those Internet experts on the website who have helped answering my questions.

LIST OF FIGURES

Title	Page
Figure 2.1: Full wave stepping position	5
Figure 2.2: Half wave stepping position	5
Figure 2.3: Robot arm movement using half wave stepping	6
Figure 2.4: Home position	7
Figure 2.5: Left hand side position	8
Figure 2.6: 180° position	8
Figure 2.7: Right hand side position	9
Figure 2.8: H-bridge circuit	10
Figure 2.9: H-bridge circuit turning clockwise direction	11
Figure 2.10: H-bridge circuit turning anti-clockwise direction	12
Figure 3.1: Project manager structure	13
Figure 3.2: PORT A structure	14
Figure 3.3: TRISA structure	14
Figure 3.4: PIC16F690 microcontroller	15
Figure 3.5: Flow diagram of the whole program	24
Figure 4.1: isometric view (position of EE-SX671)	26
Figure 4.2: bottom view (position of EE-SX671)	26
Figure 4.3: Front view (position of EE-SX671)	27
Figure 4.4: Photo of the EE-SX671 sensors and circuit	27
Figure 4.5: isometric view (position of PM2-LH10)	28
Figure 4.6: Top view (position of PM2-LH10)	28
Figure 4.7: Front view (position of PM2-LH10)	29
Figure 4.8: Photo of actual PM2-LH10 sensors and the structure	29
Figure 4.9: Circuit testing for EE-SX671	30
Figure 4.10: The timing chart for EE-671 sensors	31
Figure 4.11: Testing circuit for PM2-LH10	32
Figure 5.1: Air suction test	37
Figure 5.2: suction control assembly	38
Figure 5.3: Addition force acting on robot	39
Figure 6.1: internal of vacuum cleaner	41
Figure 6.2: The action of the vacuum pad (before compression)	42
Figure 6.3: The action of the vacuum pad (after compression)	42
Figure 6.4: Actual drawing of pipe used	44
Figure 6.5: The dimension for the actual pipe	45
Figure 6.6: Robot partial assembly	46
Figure 6.7: The flow rate test (collect from pipe)	48
Figure 6.8: The flow rate test (collect directly from tap)	49
Figure 6.9: The flow rate test (used smoke for test)	50
Figure 7.1: Circuit to control motor	52
Figure 7.2: Circuit to control fan	53
Figure 7.3: Circuit to control detecting sensors	54
Figure 7.4: Circuit to control PM2-LH10 sensors	54

Figure 7.5: Fan circuit for project	55
Figure 7.6: Detecting sensor circuit for project	56
Figure 7.7: PM2-LH10 sensor circuit for project	56
Figure 7.8: Small vacuum pad drawing	57
Figure 7.9: Dimension for small vacuum pad	58
Figure 7.10: Big vacuum pad drawing	59
Figure 7.11: Dimension for big vacuum pad	59
Figure 7.12: Structure drawing	60
Figure 7.13: Dimension for structure	61
Figure 7.14: Sensor bracket drawing	61
Figure 7.15: Dimension for Sensor bracket	62
Figure 7.16: Support bracket drawing	63
Figure 7.17: Dimension for support bracket	64
Figure 7.18: Support beam drawing	64
Figure 7.19: Dimension for Support beam	65
Figure 7.20: fan bracket drawing	66
Figure 7.21: Dimension for fan bracket	67
Figure 7.22: fan support drawing	67
Figure 7.23: Dimension for fan support	68
Figure 7.24: Overall drawing	69
Figure 7.25: Error make for fan bracket	70
Figure 8.1a: Correct position	72
Figure 8.1b: Incorrect position	72
Figure 8.2: Welding chart	74

LIST OF TABLES

Title	Page
Table 4.1: Detecting distance result	32
Table 5.1: Total weight and total force	38
Table 6.1: The flow rate result	49
Table A: Half-wave (clockwise direction)	97
Table B: Half-wave (anti-clockwise direction)	97

TABLES OF CONTENTS

Contents	Page
CANDIDATES CERTIFICATION	i
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	iv
LIST OF TABLES	vi
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 STEPPER MOTOR	4
2.1 How stepper motor works	4
2.2 Full wave stepping and half wave stepping	4
2.3 Choosing the angle of each steps	6
2.4 Choosing the correct speed for one complete revolution	7
2.5 Stepper motor arm movement	7
2.5.1 What is a H- bridge Circuit?	9
2.5.2 Why need to use H-bridge?	9
2.5.3 How does it work?	9
CHAPTER 3 PIC programming	13
3.1 Rules for choosing the PORTS	14
3.1.1 Deciding which PORTS to be input/output	14
3.2 Software programming	16
3.3 Flow diagram for the whole project	20
CHAPTER 4 SENSORS	25
4.1 Theory of infrared sensors	25
4.1.1 Photomicrosensor	25
4.1.2 Convergent Reflective Micro Photoelectric Sensor	28
4.2 Testing of sensors	30
CHAPTER 5 Computer Fan	34
5.1 Different type of computer fan	34
5.1.1 PSU fan	34
5.1.2 Graphics card fan	34
5.2 Selecting the correct air flow rate	34

5.3 Whole weight of the robot	38
CHAPTER 6 Vacuum Cups/Pads	41
6.1 Theory of how vacuum pads works	42
6.1.1 Calculation of the pressure drop	43
6.1.2 Deciding on how many vacuum pads used	46
6.2 Different types of vacuum cups/pads	47
6.3 Troubleshooting	48
CHAPTER 7 Electrical/Mechanical Designs	51
7.1 Types of software used	51
7.2 Circuitry diagrams	51
7.2.1 Diagram for motors	51
7.2.2 Diagram for computer fan	52
7.2.3 Diagram for sensor	53
7.2.4 Overall diagram for my project	54
7.3 Mechanical design	57
7.3.1 Design for small vacuum pad	57
7.3.2 Design for big vacuum pad	58
7.3.3 Design for structure	60
7.3.4 Design for sensor bracket	61
7.3.5 Design for support bracket	62
7.3.6 Design for support beam	64
7.3.7 Design for fan bracket	65
7.3.8 Design for fan support	67
7.4 Overall final design	68
7.5 Troubleshooting	69
CHAPTER 8 Future Development /Improvement	71
8.1 Improvement in the software programming	71
8.1.1 Adding a delay timer at every position	71
8.2 Improvement in the hardware design	72
8.2.1 Improve the methods of joining	72
8.2.2 Improve the quality of the vacuum pads	73
8.2.3 Selecting of different type of sensors	73
8.3 New things added	73
8.3.1 Different type of welding	74
CHAPTER 9 CONCLUSION	76
REFERENCE	77

APPENDIX A	78
Project specification	78
APPENDIX B	80
Bill of Material	80
Source code	80
APPENDIX C	83
PIC16F690 data	83
Computer fan data	91
Fan specification	91
Vacuum data	92
EE-SX671 data sheet	95
PM2-LH10 data sheet	96
APPENDIX D	97
PORT B structure	97
TRISB structure	98
PORT C structure	98
TRISC structure	98

CHAPTER 1

INTRODUCTION

This chapter will list out what are the objectives of this project. A short introduction of individual chapters was discussed giving the reader an idea about each chapter.

OBJECTIVES

1. Research on components required by the robot.
2. Design the bi-direction circuit for the stepper motor.
3. Write the PIC program for this robot.
4. Determine the correct sensor used in this robot.
5. Sizing the correct fan used to created the suction force for this robot.
6. Research on the vacuum suction theory.
7. Selecting of tube/pipe for this project.
8. Combining all the things listed above to build the final product.

Below are the list of things that will be discuss for this report.

Chapter 2 Stepper motor

This chapter will discuss on the full wave stepping and half wave stepping used in the stepper motor. How to calculate each angle steps and the speed required for the motor to rotate one complete revolution. A H-bridge circuit diagram will be used to show how the rotation of the stepper motor works , whether it is a clockwise direction or anti clockwise direction. How this circuit has been modify and put it into the final products. Lastly discussion on how the Pulse Width Modulation will be used in this robot.

Chapter 3 PIC programming

This chapter will be explaining on the microcontroller IC chip, PIC16F690, works as well as the limitation on its programming language. Steps of starting a new program will also be discuss here and the rules to follow when jumping from one BANK to another. How to decide whether the whole PORT to be input/output or whether if partially input and output ports. Special address will be introduce and how to make full used for each of them will be discuss, certain things need to be introduce before they are able to assign to the PORTs.

These are the PORTS to be discuss that are shown below. Whether they are 8 bits long or just be able to used half of the bits in each PORTs. PORT A, B and C will be discussed the most as they are mostly used. Others like TRISA, B, C etc, will also be introduced in this chapter.

Chapter 4 Sensor

This chapter will be discussing about different type of sensors and which type of sensor is used for what purpose in this project. As there are two different type of purpose the sensor are used for in this project. One of them are used to detect if the vacuum cup are in place while the other is to detect if there are anything blocking the way of the robot.

Chapter 5 Computer fan

This chapter will be explaining on the different type of fans used and what are the things to consider when choosing the fan such as the air flow rate and the pressure required for this project. Other things such as ways to minimize the noise rate but increase the air flow rate at the same time. The most important thing for this project is to reduce the thing to build up the required pressure therefore the faster then fan rotates the more air is being suck out of the restricted volume. This chapter will also explained if the sizing of the fan is chosen correctly and what will happen if the fan is not chosen correctly.

Chapter 6 Vacuum cups/pads

This chapter will be going through the theory of the vacuum cups as well as the theory of the vacuum cleaner. Using both this theory I will it apply to the final product for this project. The formula to calculate how much pressure will be build up. Different type of material used for vacuum cup will have different result in different temperature. Discussion on the vacuum pad I am using it for this project and to check if they are suitable for this project. In this chapter, I will also discuss on the problems that I have face when handing this project. Troubleshooting of this problems will also be discussed here.

Chapter 7 Electrical/Mechanical Design

This chapter will be discussing about what type of software used to draw the circuit for this project. Here also discuss about the theory of each different circuit and ways for modifying them for the final products. There will be a short brief report on each of the circuit used to control the motor, fan etc. In this chapter, we discuss another software will be used to design the hardware components. Different types of materials, weight and cost of the material will also be discussed here. This will conclude the final design for the robot and at the end of this chapter, there will steps that I have make to upgrade the design as well as the problems of the design.

Chapter 8 Future development/Improvement

This chapter will be going through some of the improvements that can be made to the existing design. Future development of the project can be carry on as this includes addition of equipment or using better material for this existing design. Other problems face when constructing this project will be discuss here as well as the solution to overcome them.

Chapter 9 Conclusion

This will discuss on what the objectives that have are being completed and what is not. A final conclusion for the whole project will be stated here.

CHAPTER 2

STEPPER MOTOR

2.1 How does stepper motor works

A stepper motor can be view as a DC motor with the number of poles on both rotor and stator. Additional, soft magnetic material with many teeth on the rotor stator cheaply multiplies the number of poles. The number of steps can be control by software by using a certain formula; they are control by the number of poles on the stator as well. Basically there are two different kinds of stepping known as full wave stepping and half wave stepping. These will be discussed in detail in the next section (**section 2.2**). Once the kind of stepping have being decided, the next thing to discuss is about the angle of each step the motor rotates. The detail will be discussed in **section 2.3**.

Next thing is to know that it is possible to drive unipolar stepper motor with bipolar drivers. The way to do so is to connect the output pins of the driver to 4 transistors. The transistor must be grounded at the emitter and the driver pin must be connected to the base. Collector is connected to the coil wire of the motor.

In order for the motor to move in bi-directional, which mean the motor can turn clockwise and anticlockwise. A circuit has to be designed to do that. It will have to use a H-bridge circuit to do that.

Lastly we need to consider the time for the motor to reach one complete revolution. This will determine the agilities of the robot and how fast it is going to move. And different clock pulses were applied to the chip to test whether the motor speed will react accordingly. Lastly is the explanation on what is Pulse Width Modulation (PWM). All of these will be stated in details in **section 2.4**.

2.2 Full wave and half wave stepping

As discuss earlier in this report, there are two different kind of stepping. First we will discuss about the full wave stepping and then the half wave stepping.

For full wave stepping, there are only 4 positions, as each position is 90° different of the pervious point. Therefore there is only 4 steps altogether for a full wave stepping, we can also control whether it rotates in a clockwise direction or an anti-clockwise direction. This work the same way in the half wave stepping too. The difference between a half wave stepping and full wave stepping will be shown in the diagram below in this section. Below show the full wave diagram:

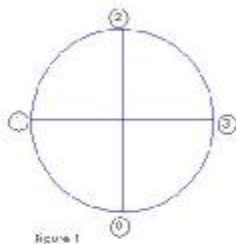


Figure 1
Full wave stepping position

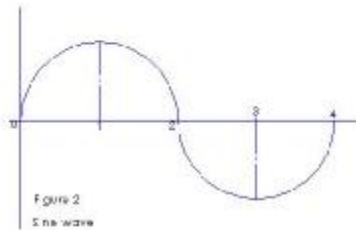


Figure 2
Sine wave

Figure 2.1: Full wave stepping position

For half wave stepping, there are 8 positions, as each position is 45° different of the pervious point. Therefore there are only 8 steps altogether for a half wave stepping. Below show the half wave diagram:

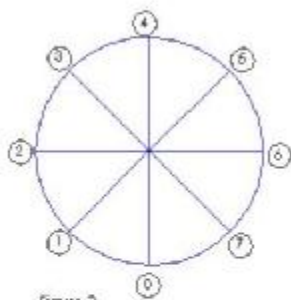


Figure 3
Half wave stepping position

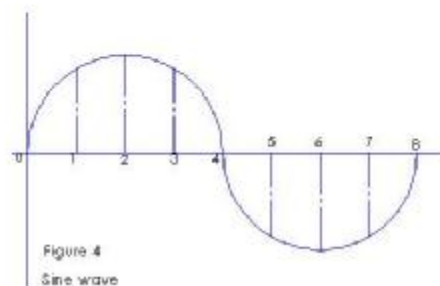


Figure 4
Sine wave

Figure 2.2: Half wave stepping position

Thus from the figure shown above I can use this theory for my software programming the table for this will be discussed in the next section and they are shown in the appendix.

2.3 Choosing the angle of each steps

When come to choosing the angle of every steps of the stepper motor, here is the formula to be used that are stated below:

$$\text{Angle for each step} = 360^\circ / 2 \times N$$

N= the number of the pole on the stator.

For this project, it need not be to accurate therefore the angle for each steps need not be too small. Since the stepping sequence is half wave stepping and it requires 8 steps to complete one revolution, below are the calculation I have done by using the formula above.

$$\begin{aligned} \text{Angle of each step} &= 360^\circ / 2 \times 4 \\ &= 360^\circ / 8 \\ &= 45^\circ \end{aligned}$$

Below are the diagram shows how the stepping sequence in one complete revolution and how it is related to the sine wave signals.

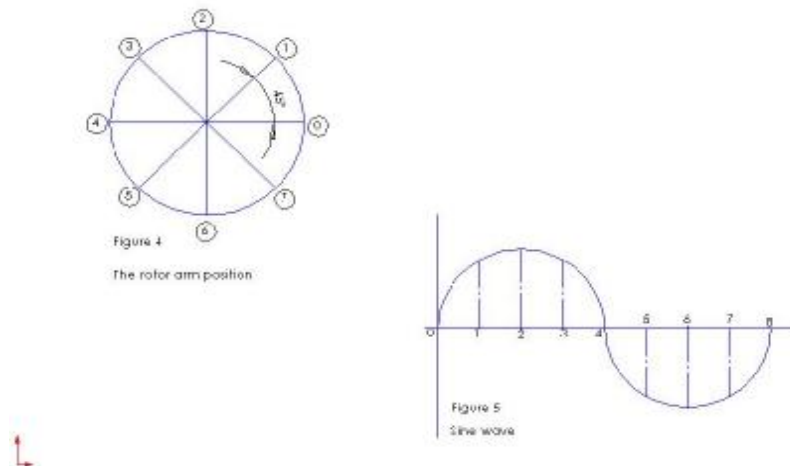


Figure 2.3: Robot arm movement using half wave stepping

By using the command from the data taken from appendix D, the command to rotate the arm to the left by 90° will be 1000 in binary and in a clockwise direction. Converting that to hexadecimal, it will be 08h. The command to rotate the arm to a 180° up in the clockwise direction is 0001 in binary. Converting it into hexadecimal value, it will be

01h. Lastly the command to rotate the arm to the right by 90° will be 0010 in binary and in an anti-clockwise direction. Converting that to hexadecimal, it will be 02h.

2.4 Choosing the correct speed for the motor to make one complete revolution

Stepper motor has lower torque capacity as compare to other DC motor thus it has limited speed. Therefore to determine how fast the motor going to move in one complete revolution depends on how fast the switch on of each pole in the stator. But firstly we need to know how many type of speed control it has.

As stated earlier in this report, they are two different type of speed control for the motor. They are known as the open loop speed controller as well as the closed loop controller also known as the feedback speed controller/ Feedback speed controller will be better but more complicated. An open loop speed controlled will be must easier but it will have missing pulsing especially for stepper motor.

2.5 Stepper motor arm movement

Below are the position of each position for this project, there are show in the home position follow by the left hand side rotation then follow by the 180° position and lastly the right hand side position. As long as the movement position is as stated below, then the rotation of the motor arm is complete. The figures are shown here:

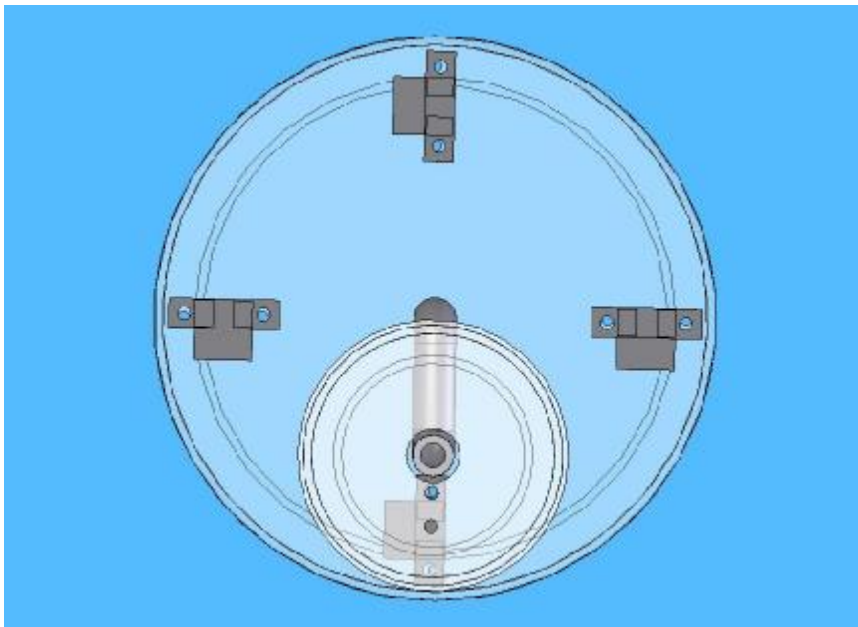


Figure 2.4: Home position

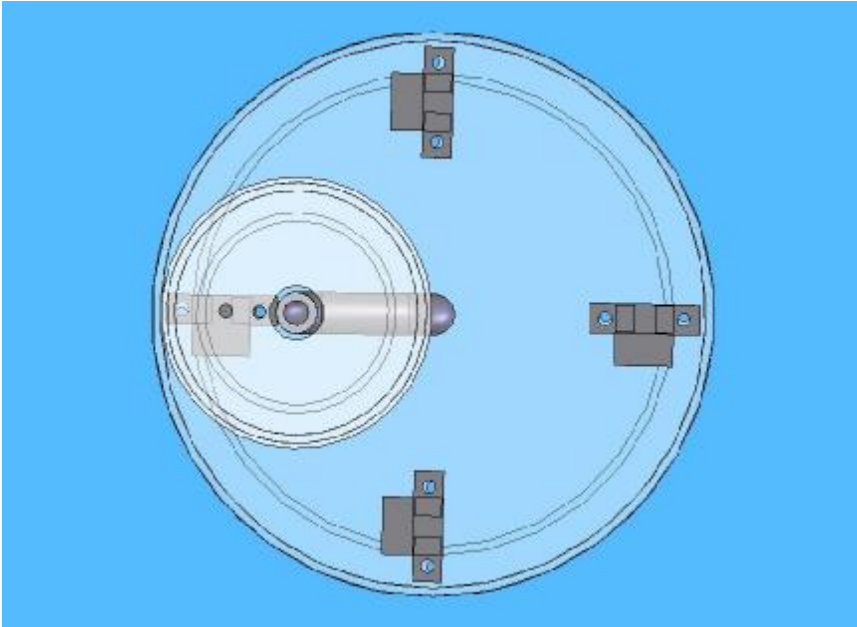


Figure 2.5: Left hand side position

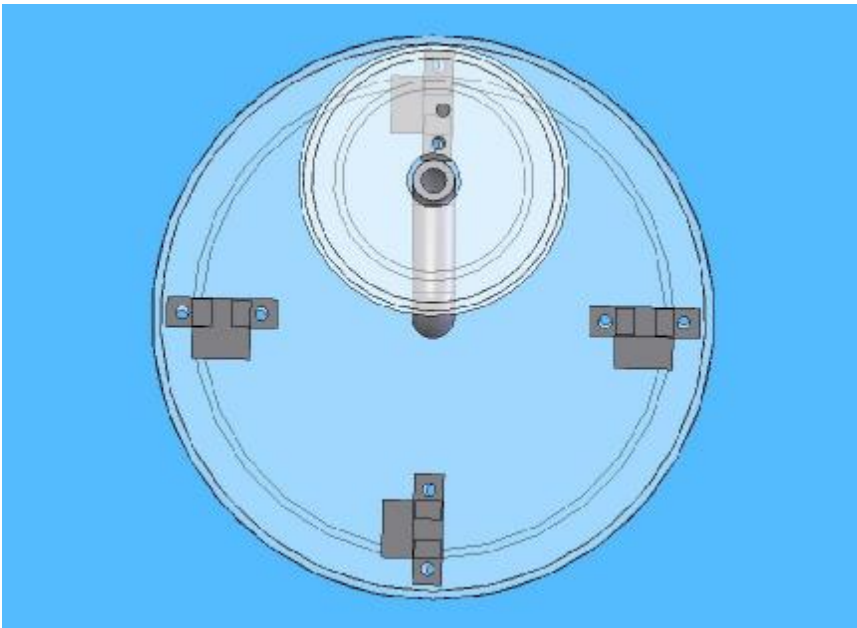


Figure 2.6: 180° position

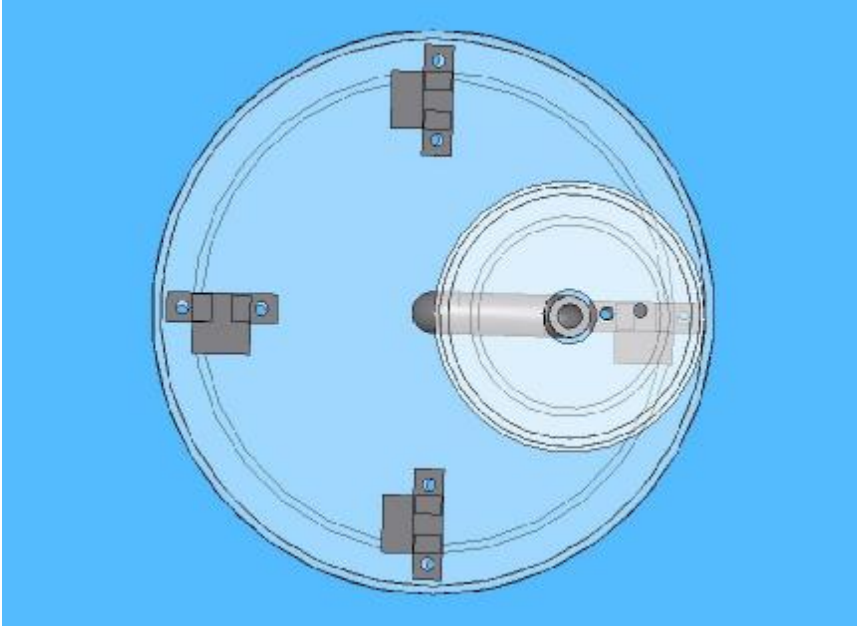


Figure 2.7: Right hand side position

Left hand position is for the robot to avoid any object blocking the right path it is moving. As for the right hand position is the opposite action of the left hand position of the robot. If there are no objects blocking its path, it will move on with the original path that is to the centre top position. The photomicrosensors are there to sense if the motor arm has reach the require position.

2.5.1 What is a H-bridge circuit?

The H-bridge derived its name from the common way the circuit is drawn. This is the only solid-state way to operate the motor in both directions.

2.5.2 Why need to use H-bridge?

To operate the motor in both directions and using a low voltage to control high input voltage to the motor.

2.5.3 How does it work?

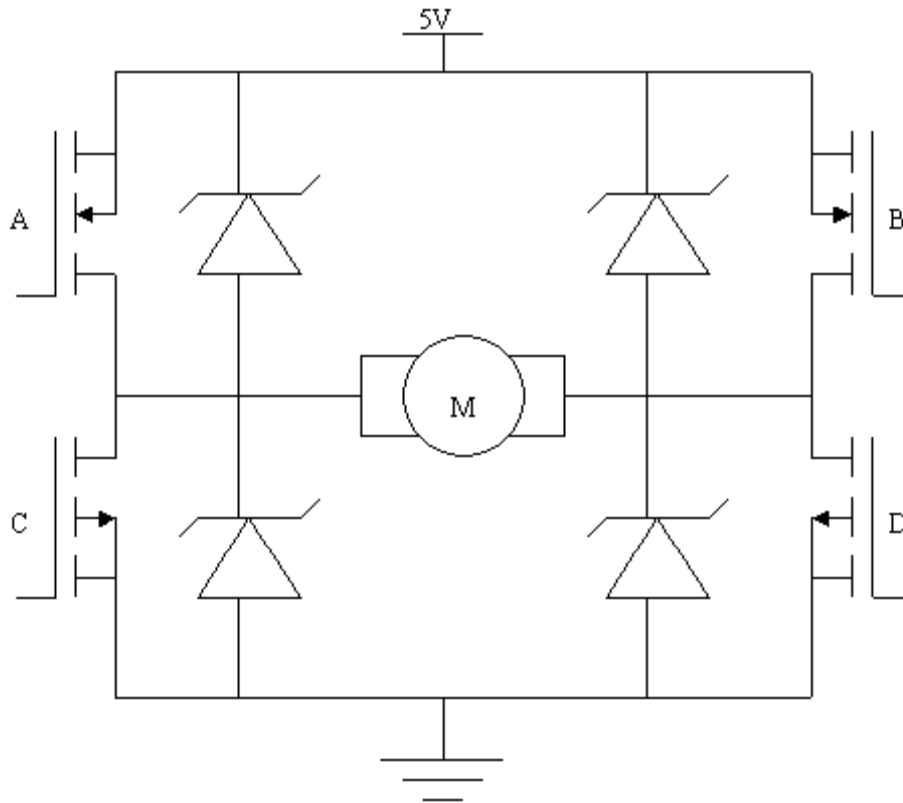


Figure 2.8: H- bridge circuit

This circuit is run using 4 MOSFETs. Each time only two MOSFETs will be turned on to run the motor. Zener diode is added to prevent the back voltage that is generated by the motor coil when the power is switched on and off. This fly back voltage can be many times higher than the supply voltage. If without using the zener diodes, the MOSFETs may get burn out. Due to sudden reduce of current.

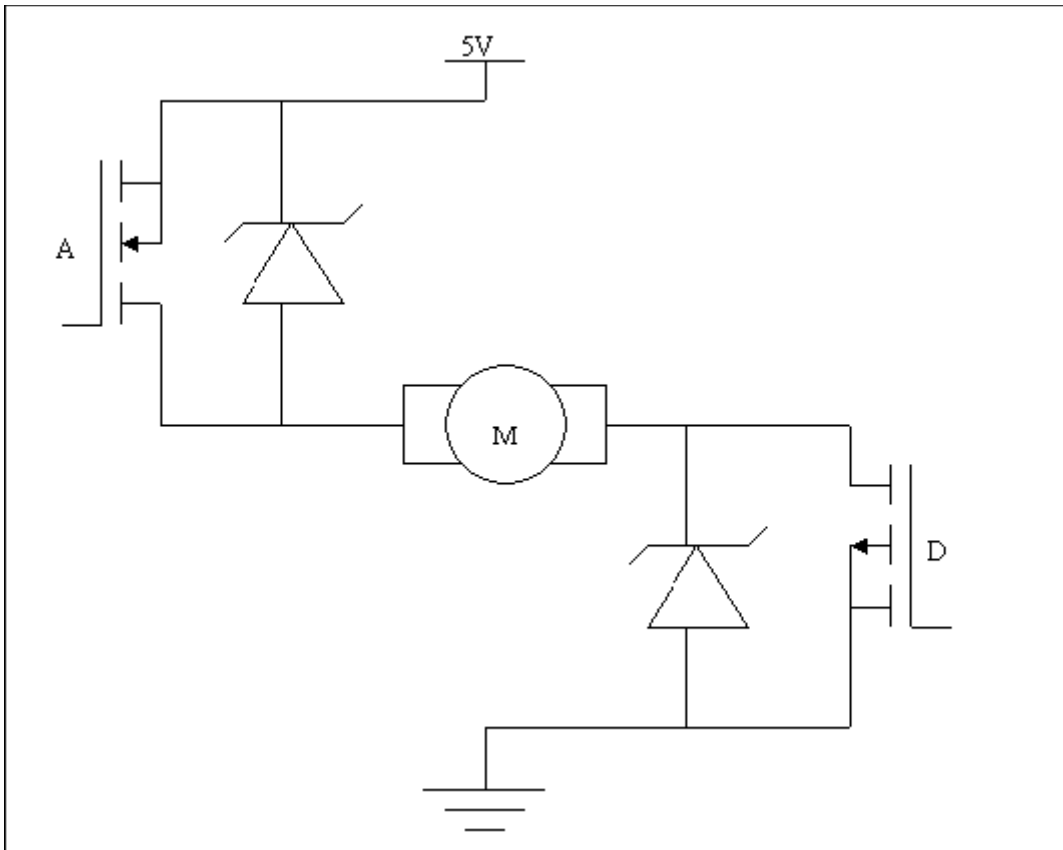


Figure 2.9: H-bridge circuit turning in clockwise direction

To run the motor in clockwise direction, turn on input 'A' and 'D'. Current will flow through MOSFET 'A', to the motor, and then to MOSFET 'D' and then ground. Forming a close loop.

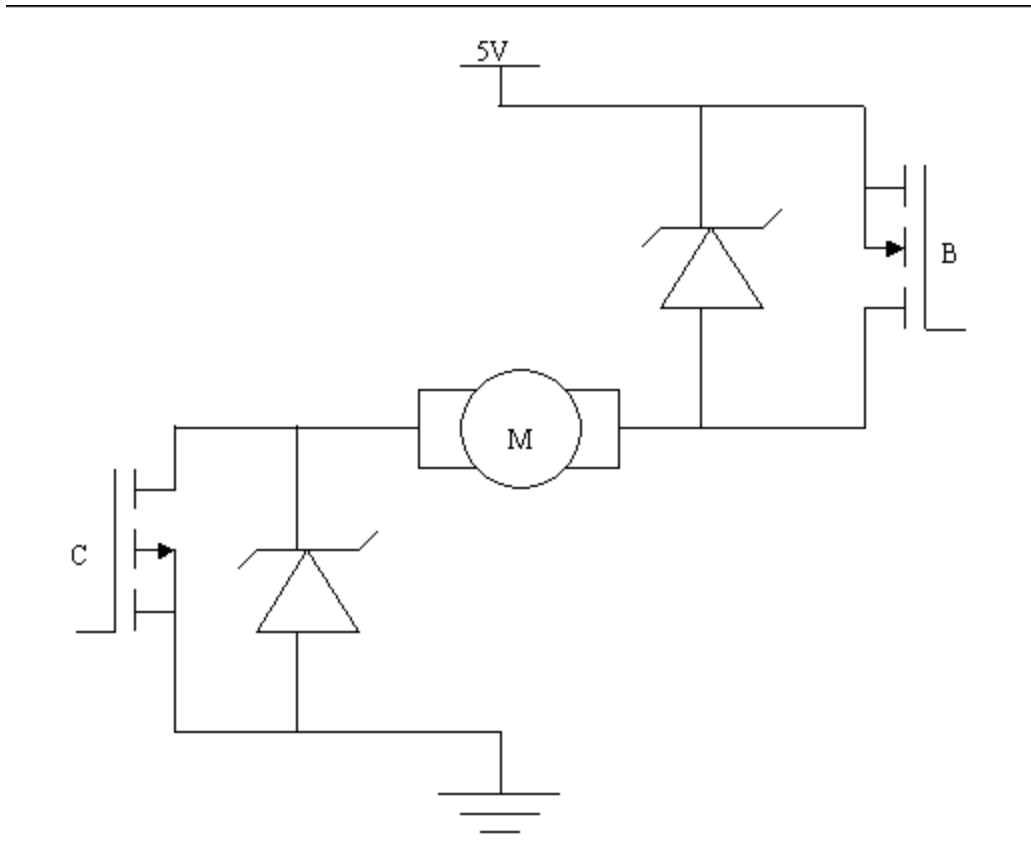


Figure 2.10: H-bridge circuit turning in anti-clockwise direction

To run the motor in anti-clockwise direction, turn on input 'B' and 'C'. Current will flow through MOSFET 'B', to the motor, then to MOSFET 'C' and then ground. Forming a close loop.

Pulse Width Modulation (PWM) is using pulses to control the switching "on" and "off" of a power supply. The percentage of the duty cycle is the "on" time of the power supply. The higher the percentage, the longer will be the switched "on" time of the power supply.

CHAPTER 3

PIC Programming

The software programming that has been chosen for this project is known as the PIC programming. It belongs to the MLAB groups of microcontroller. It is unlike the microcontroller from other brands. There are certain rules to follow when choosing the PORTs. Although the final data are stored inside the PORTs address, they are stored by using the store command like that in the Motorola microcontroller chip, it has to be stored in another address that is indirectly controlling the PORTs to be inputs or outputs. This detail will be discussed in the next section.

In PIC programming, there are few commands as compared to other microcontrollers, therefore there are rules to follow when one is to use them. All these will be discussed in details which I have separated them for easy understanding use.

The below diagram will show how the MPLAB IDE project manager operates:

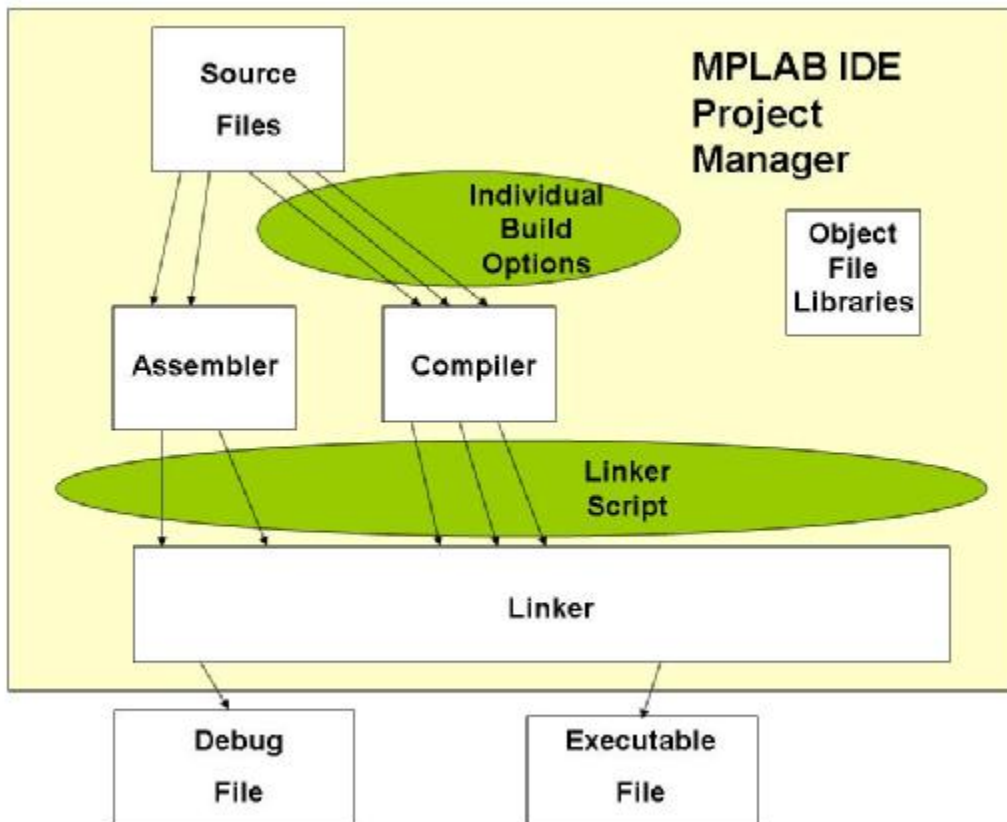


Figure 3.1: Project manager structure

The project manager organizes the files to be edited and associated files so that they can be sent to the language tools for assembly or compilation. The linker's task is to place the object code fragments from the assembler and compiler as well as from the object libraries into proper memory areas of the embedded controller. This is to ensure the

modules function is linked together with one another. This is how the project build system looks like/works.

3.1 Rules for choosing the PORTs

In PIC 16F690 microcontroller, there are eighteen general purposes of input/output pins but not every bits in the PORTs are usable. For example PORT A is a 6 bit wide bidirectional port. This means that only bit 0 to bit 5 are usable as for the last two bits (bit 6 and bit 7) are under “don’t care” system. This means that they do not have any effect on the PORT whether they have a ‘1’ or ‘0’ data in them. Below show the structure of PORT A:

PORT A address at 05h/105h

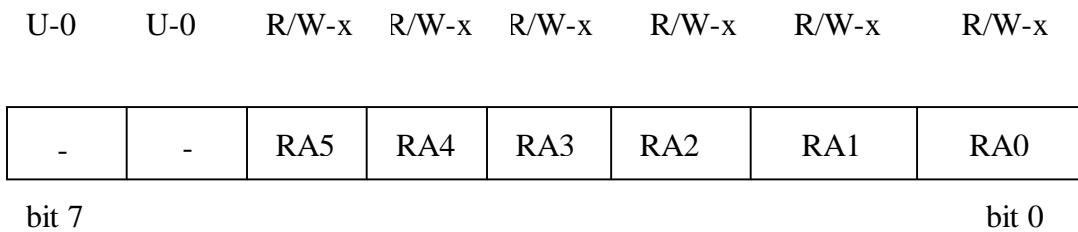


Figure 3.2: PORT A structure

However to decide on whether they are input or output pin, depend on the data if they are ‘1’ or ‘0’. However these data are store under another address which control the PORTs, they are known as the TRIS address. Now we are discussing on PORT A therefore the data store in the TRIS address is known as TRISA address. The data in the TRISA address is a ‘1’ then PORT A will be an input. If the data store in TRISA address is ‘0’ then it is an output pin. Within this PORT A alone we can have different number of input pins as well as output pins, all this depend on which bit of TRISA address is a ‘1’ or ‘0’. Below show the status of the TRISA:

TRISA address at 85h/185h

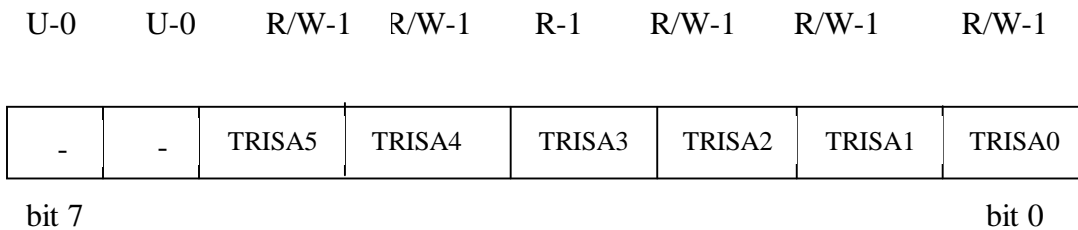


Figure 3.3: TRISA structure

Bit 7-6 Unimplemented(read as ‘0’)

Bit 5-0 Tri-State Control bit

1 = PORT A pin configured as an input (tri-stated)

0 = PORT A pin configured as an output (tri-stated)

The same principle apply to all the PORTs such as PORT B, PORT C etc. The structure for PORT B, TRISB, PORT C and TRISC will be shown in the appendix at the end of this report. Unlike PORT C which all the bits in this ports will be used, PORTB is a 4 bit wide, bidirectional port and PORT C is a 8 bit wide bidirectional port. As PORT B is a 4 bits wide port and from the structure of PORT B, only the upper most 4 bits are selected to control to determine if they are inputs or output pins. The lower 4 bit are in the “don’t care” system.

3.1.1 Deciding which PORTs to be input /output

For this project here at the things I need, they are listed below:

- 2 computer fans
- 2 detecting sensors
- 4 position sensors
- 1 stepper motor

Firstly here is a diagram of how the PIC16F690 microcontroller looks like, here also show the pin location on the ic chip itself:

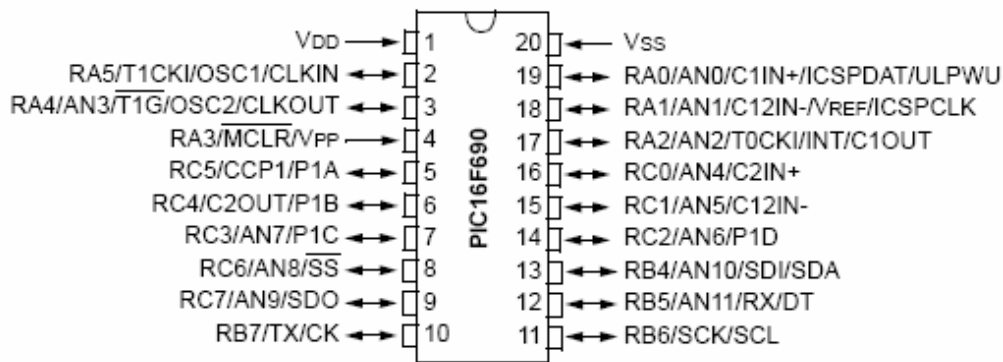


Figure 3.4: PIC16F690 microcontroller

All the sensors are to detect if there is any thing obstructing the path either left or right of the robot as well as if the vacuum cup is in position. As they are used to control certain condition of the project, therefore they are all inputs. As PORT A has 6 bits, that means I am able to use them for the sensors. Below are the commands to set PORT A to be inputs:

```

bsf      03h,5      ; Go to Bank 1
movlw   2fh        ; Put 11 1111 into W
movwf   TRISA      ; move 11 1111 onto PORTA, setting all pin to input
bcf     03h,5      ;Go back to Bank 0

```

As for the computer fans, they are used to turn on and off the vacuum suction, each fan required 2 bits from one of the PORTs. They are a total of 4 bits, and this is control by the software condition to turn on or off, therefore I am using PORT B, as it can be seen in the appendix in the end of the report, there are 4 different bits. For this case, I am setting

all the 4 bits of the PORT B to be outputs. Here is the command used to set PORT B to be outputs:

```

bsf      03h,5      ; Go to Bank 1
movlw   00h        ; Put 0000 0000 into W
movwfb  TRISB      ; move 0000 0000 onto PORTB, setting all upper bits to be
                  ; outputs
bcf      03h,5      ; Go back to Bank 0

```

For the motor, I required a few command for it and since all the other PORTs bits have been used up completely, PORT C is the one left. Here are the few things that are required for the stepper motor to do, one of them is to rotor the motor arm in the anti clockwise by 90° and another to rotate in the anti clockwise or clockwise by 180°. The are another command is to rotate it in a clockwise by 90°. As for the stepper motor, there are a certain things that need to configure such as full wave or half wave stepping, all this required bits for PORT C to control. The command given below are to set PORT C to be outputs:

```

bsf      03h,5      ; Go to Bank 1
movlw   00h        ; Put 0000 0000 into W
movwfb  TRISC      ; move 0000 0000 into PORTC, setting all pins to be
                  ; outputs
bcf      03h,5      ; Go back to Bank 0

```

3.2 Software Programming

Here is the set of the software programming for the whole project. They are listed below:

```

                CBLOCK          0X05          ;start of internal program
COUNTER0, COUNTER1          ;define delay counters
                ENDC

#define step          PORTA,4
#define direct       PORTA,5
#define fan1         PORTA,0
#define fan2         PORTA,1
#define sens1        PORTB,5
#define sens2        PORTC,2
#define Psens1       PORTC,6
#define Psens2       PORTC,5
#define Psens3       PORTC,4
#define Psens4       PORTC,3

_ResetVector set          0X00          ; reset vector
                org          _ResetVector ;Pic start code execution from here
                goto         START

```

```

nop
nop
START
    bsf        03h,5      ; Go to Bank 1
    movlw     00h        ; Put 00 0000 into W
    movwf    TRISA      ; move 00 0000 onto PORTA, setting all pin
                        ; to input
    bcf        03h,5      ;Go back to Bank 0
    bsf        03h,5      ; Go to Bank 1
    movlw     09h        ; Put 0001 0000 into W
    movwf    TRISB      ; move 0001 0000 onto PORTB, setting all
                        ; upper bits to be outputs except bit 5
    bcf        03h,5      ; Go back to Bank 0

    bsf        03h,5      ; Go to Bank 1
    movlw     00h        ; Put 0000 0000 into W
    movwf    TRISC      ; move 0000 0000 into PORTC, setting all
                        ; pins to be outputs
    bcf        03h,5      ; Go back to Bank 0

MAIN    clrwdt          ;clear watchdog timer in case of latchup

    movlw     01h        ; set sensor1 in home position, vacuum cup
                        ; should be at home position
CHKFAN  movwf    Psens1   ; move the contents of W onto PORTC,6
        goto    CFAN1    ; jump to CFAN1
        btfss   fan1     ; check if fan 1 is on
CHKFAN1 goto    CHKFAN   ; else go back to CHKFAN
        goto    CFAN2O   ; go to CFAN2O
        btfsc   fan2     ; check if fan 2 is off
CHKSEN  goto    CHKFAN1  ; go to CHKFAN1
SENOFF  goto    SENSOR   ; go to SENSOR
        btfsc   PORTC    ; both sensor off
        goto    RSENSOR  ; go to RSENSOR
CHKAGAIN movlw    01h    ; move the motor to the top(180°)clockwise
                        ; direction
        movwf   Direct   ; store the content of W onto PORTA,5
        call    DELAY    ; go to DELAY
        btfss   Psen3    ; check if top sensor on, this is to on the top
                        ; position sensor
        goto    CHKAGAIN ; go to CHKAGAIN
CHKFAN2 goto    CFAN2N   ; go to CFAN2N
        btfss   fan2     ; check if fan 2 is on
CHKFAN3 goto    CHKFAN2  ; else go back to CHKFAN2
        goto    CFAN1O   ; go to CFAN1O
        btfsc   fan1     ; check if fan 1 is off

```

```

goto      CHKFAN3 ; go to CHKFAN3
goto      CHKHOME ;go to CHKHOME
goto      MAIN    ; go back to the initial program

CFAN1     movlw    10h      ; turn on fan 1, this value is to turn on fan 1
movwf     fan1        ; store the content of W to PORTA,0, turn
                    ; on fan
return    ; return to main program

CFAN1O    movlw    00h      ; turn off fan 1, this value is to turn off fan 1
movwf     fan1        ; store the content of W to PORTB, turn on
                    ; fan
return    ; return to main program

CFAN2O    movlw    00h      ; turn off fan 2, this value is to turn off fan 2
movwf     fan2        ; store the content of W to PORTA,1, turn
                    ; on fan
return    ; return to main program

CFAN2N    movlw    20h      ; turn on fan 2, this value is to turn on fan 2
movwf     fan2        ; store the content of W to PORTA,1, turn
                    ; on fan
return    ; return to main program

MOTOR     movlw    02h      ; turn on motor, anti-clockwise direction,
                    ; half step and stop at 90°
movwf     Direct     ; store content of W onto PORTC
call     DELAY       ;go to DELAY
btfs     Psen4       ; check if top sensor on, this is to on the top
                    ; position sensor
goto     MOTOR       ; go to MOTOR
return    ; return to main program

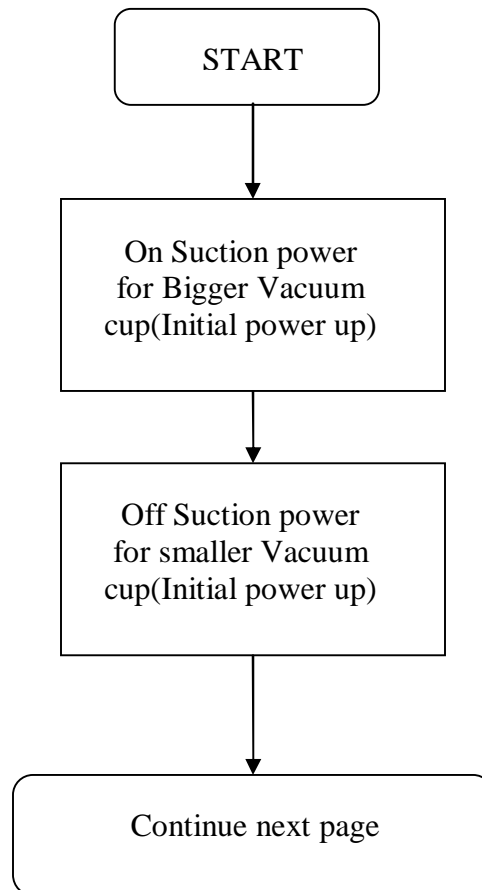
MOTOR1    movlw    08h      ; turn on motor, clockwise direction, half
                    ; step and stop at 90°
movwf     Direct     ; store content of W onto PORTC
call     DELAY       ;go to DELAY
btfs     Psen2       ; check if top sensor on, this is to on the top
                    ; position sensor
goto     MOTOR1     ; go to MOTOR1
return    ; return to main program

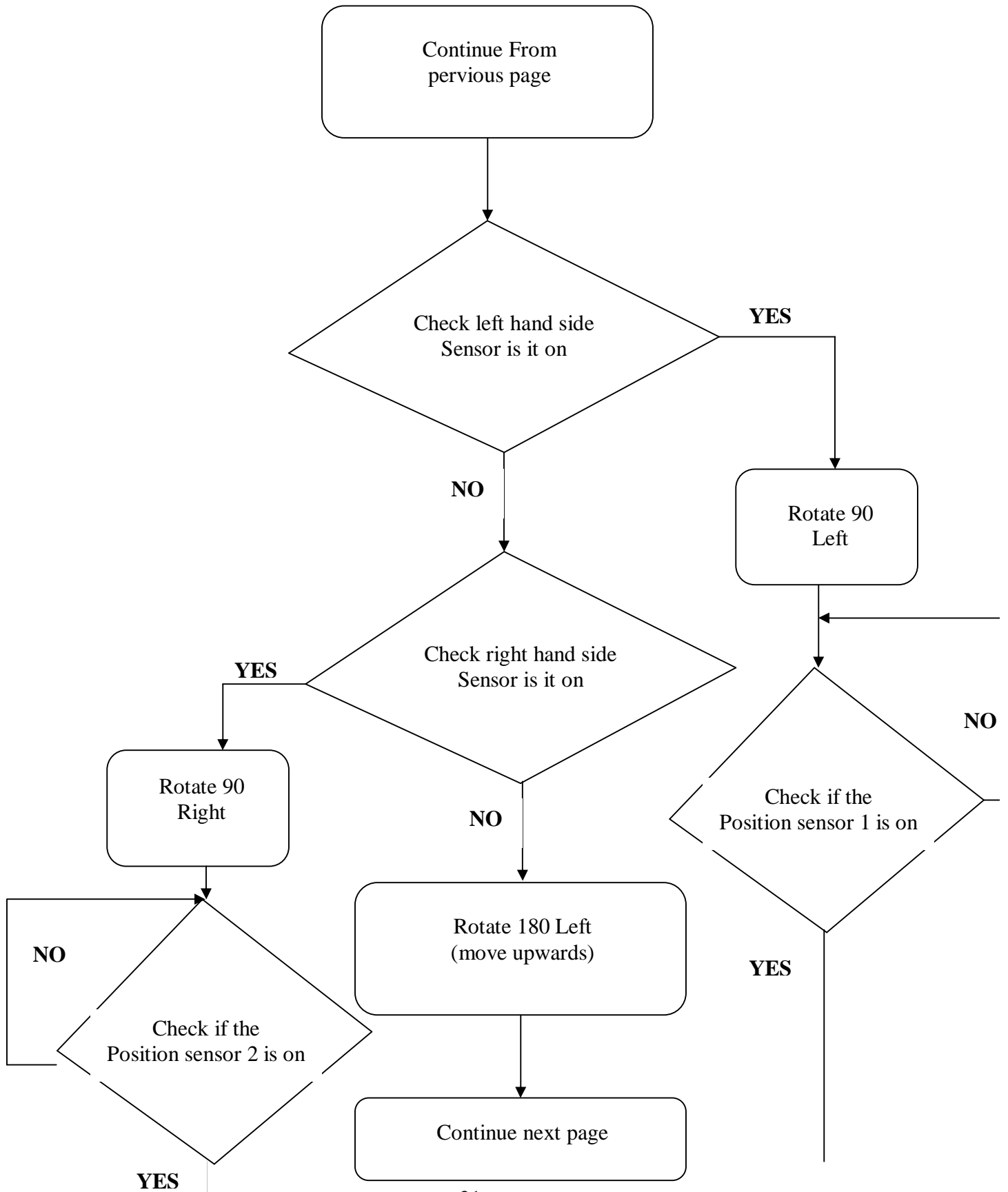
MOTOR2    movlw    01h      ; move the motor to the top(180°)clockwise
                    ; direction
movwf     Direct     ; store the content of W onto PORTA,5
call     DELAY       ; go to DELAY
btfs     Psen1       ; check if top sensor on, this is to on the top

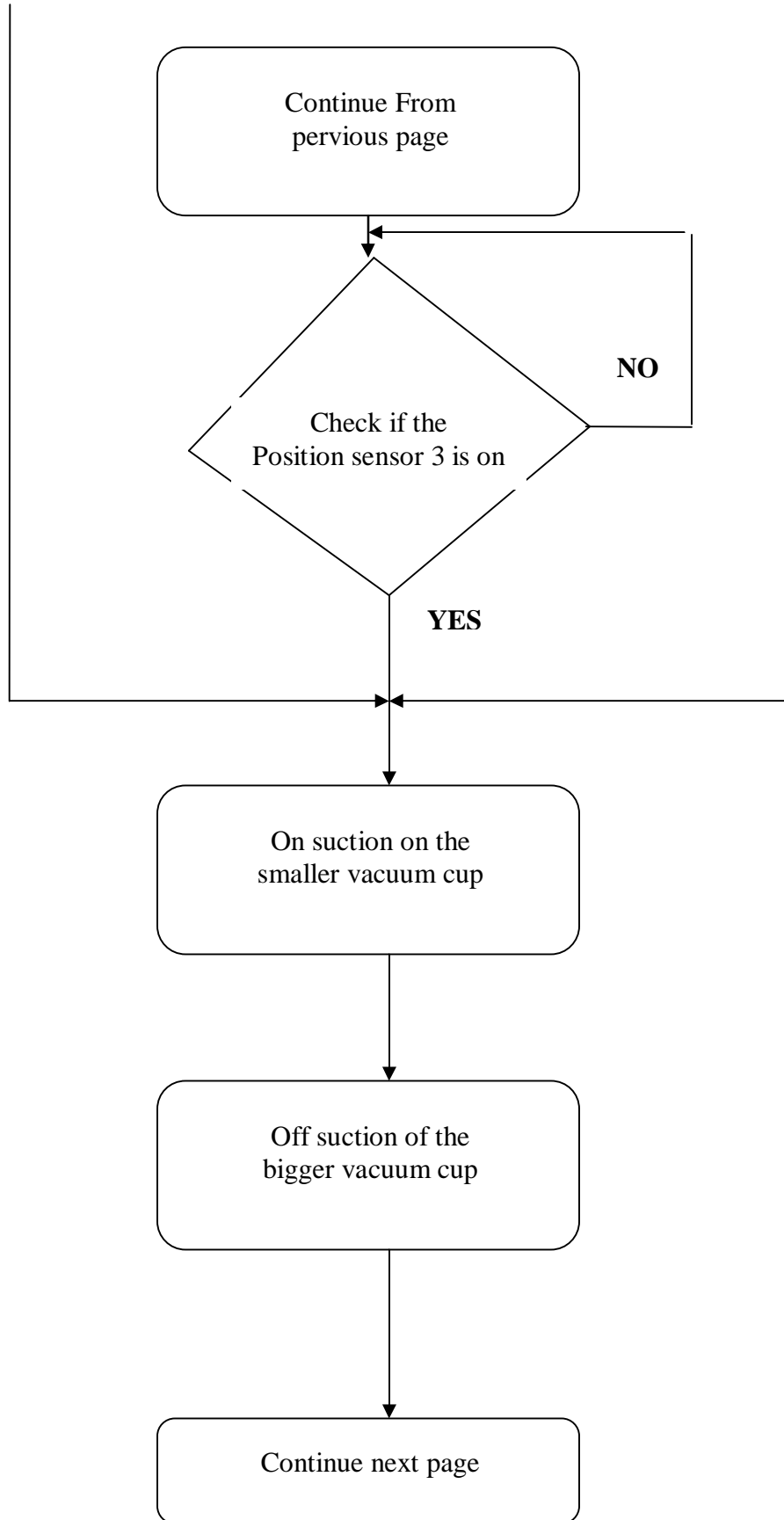
```

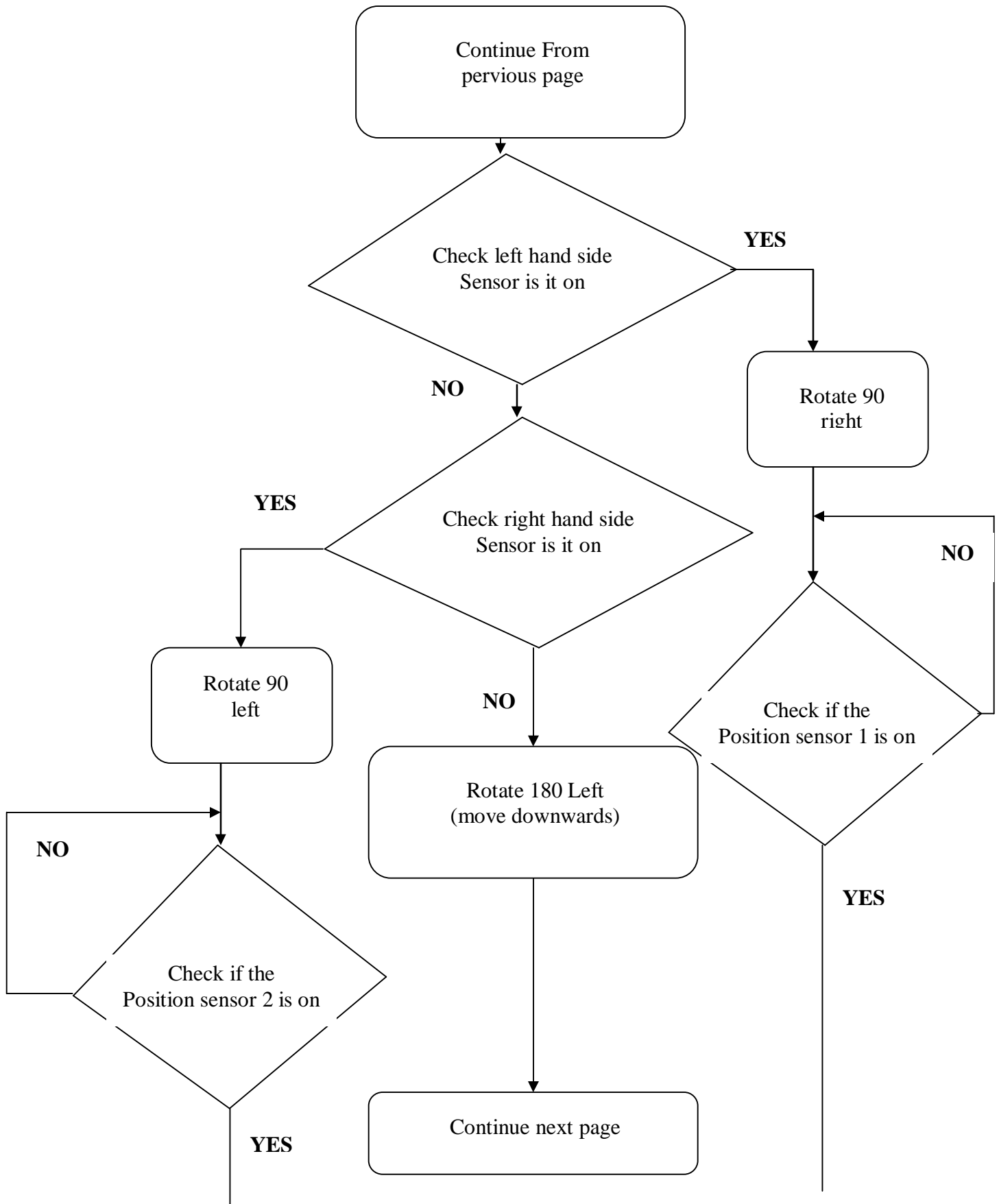

3.3 Flow diagram

The whole idea of controlling the movement of the robot is to move the vacuum cup one after the other. It can either rotate right or left or a 180° up, control by either side of the detecting sensors, that are mount onto the robot. When the left side detecting sensor sense there is an object obstaing the way, then the motor will rotate in the opposite direction to avoid the object. It has the same theory when it happen on the right side of the sensor. However when the both side of the sensors did not detect anything, it will rotate to the top side of the outer vacuum cup. Below is the flow diagram of the overall program.









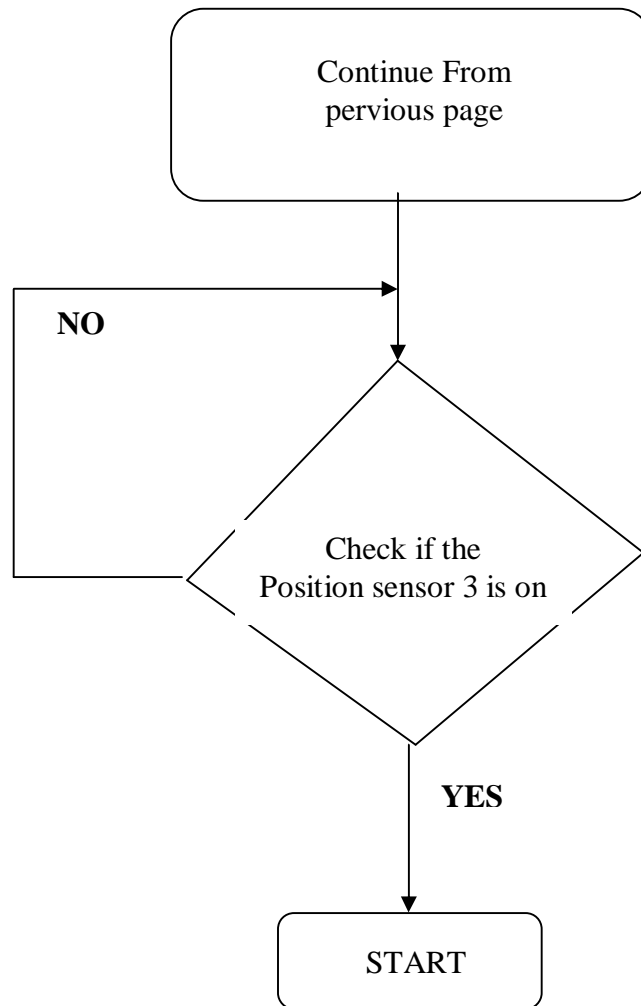


Figure 3.5: Flow diagram of the whole program

CHAPTER 4

SENSOR

There are different types of sensors in this world and for this project; I am using two different types of sensors for two different purposes. One type of sensor is used to detect if there are any object obstructing the path of the robot. Another type is used to sense if the vacuum cup have reach it destination position. These two different types of sensor will be discussed in the following section later in this report.

4.1 Theory of infrared sensor

Infrared sensors are electronic devices which detect object by measuring the infrared light radiating from an object, this light are invisible to human eyes but they can be pick up by this sensors. They are also known as the passive Infrared sensors (PIR sensors). How this sensors works? They sit in a position accepting infrared energy through the sensor face that are made from a pyroelectric material. As long as the path of radiated light is block by an object, the sensor are able to detect it, therefore this sensor for this project is used to detect if there is anything obstructing the path of the robot. The sensor I am using is the convergent reflective photoelectric sensor. As the Infrared energy is able to reach the sensor through the plastic which is transparent to infrared radiation. As the vacuum pad is make of clear plastic, this type of infrared sensor is not able to detect the motor arm has reached its designated position or not. Thus the other type of sensor will be used for this purpose. The other type of sensor I will be using is known as the photomicrosensors. They will be discussed later in this chapter.

4.1.1 Photomicrosensor

I am using the photomircosensor for detecting the vacuum pad is in the correct position. The model I am using is the EE-SX671 as the pin connection are shown in the appendix C of this report. As it is at a right angle, I am able to connect it the inner surface of the big vacuum pad with the pins still inside of the big vacuum pad. In this way I am able to solder the pin to the wires and will route this cable out of the big vacuum pad and connect to the pcb board. The figures below show the diagram of all the photomicrosensor in the right position. There are three views, isometric view, bottom view and the front view respectively.

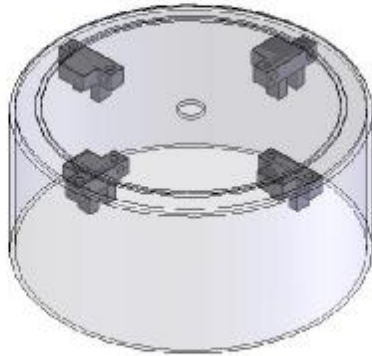


Figure 4.1: isometric view (position of EE-SX671)

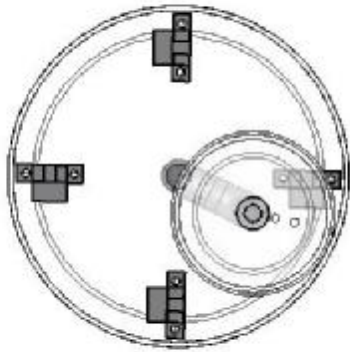


Figure 4.2: bottom view (position of EE-SX671)

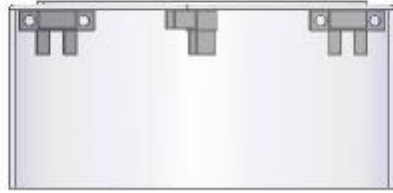


Figure 4.3: Front view (position of EE-SX671)

After looking at the drawing, here is the physical detecting sensors that has been attached to the big vacuum pad as well as I have solder the circuit that has been shown in chapter 7. Here is the photo of the item stated below:



Figure 4.4: Photo of the EE-SX671 sensors and circuit

4.1.2 Convergent Reflective Micro Photoelectric Sensor

I am using the Convergent Reflective Micro Photoelectric sensor for detecting if there is anything blocking the robot way. There will be two of them, each on the side of the structure. The model I am using for my test is PM2-LH10, the pin connection are shown in the appendix C of this report. In this way I am able to solder the pin to the wires and will route this cable from the structure base to the top of the robot. The figures below show the diagram of all the Photoelectric sensor in the correct position. There are three views, isometric view, top view and the front view respectively.

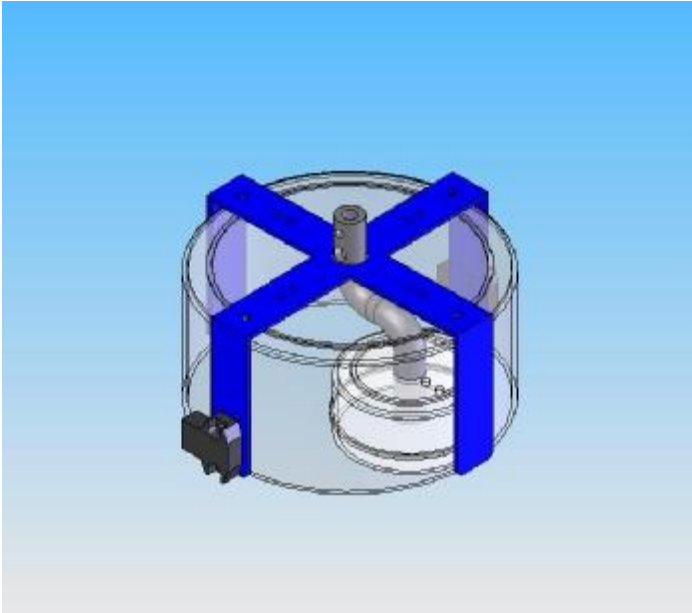


Figure 4.5: isometric view (position of PM2-LH10)

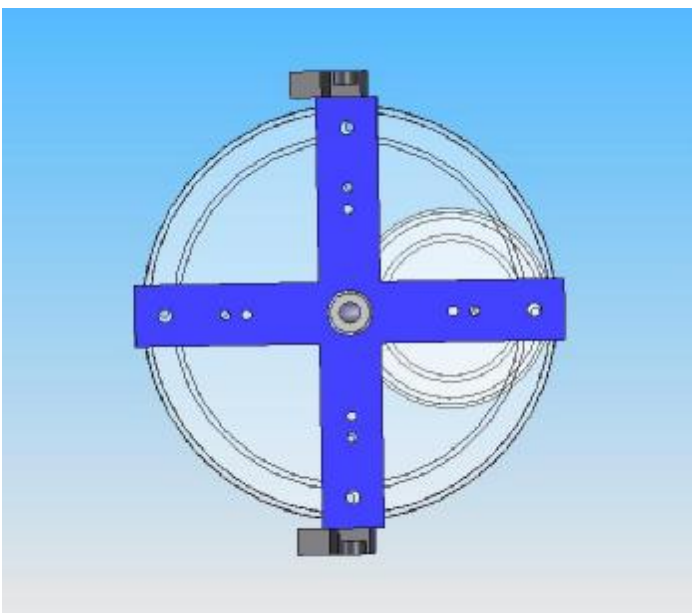


Figure 4.6: Top view (position of PM2-LH10)

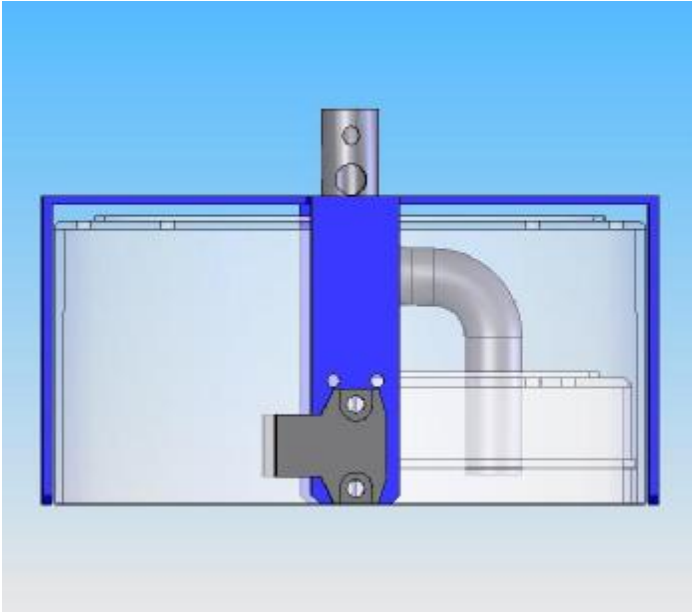


Figure 4.7: front view (position of PM2-LH10)

Below is the photo of the PM2-LH10 attached to the base structure. From here we can compare the actual diagram and the drawing attached above:



Figure 4.8: Photo of the actual PM2-LH10 sensors and the structure

4.2 Testing of the sensors

First I will show how I tested the detecting sensors and below are the figure of how I connect the sensors and test how they work. After testing that they work, then I connect them to me circuit diagram which will be show in the later part of the report. They are show in chapter 7 of this report.

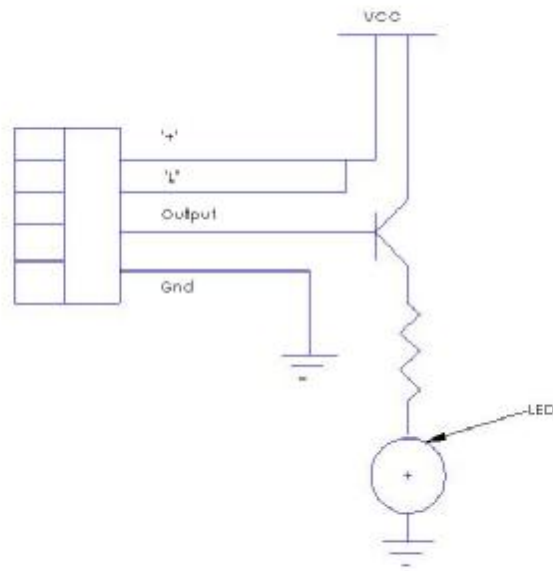


Figure 4.9: Circuit testing for EE-SX671

The sensor I using above is EE-SX671 the above connection is connected to test if I have got the connection correct as well as to check if the sensors are in working condition. After testing the circuit above, I am able to check if the sensors (EE-SX671) is in working condition. I connect in a way that as long as there is something blocking the light between the receiver and transmitter, the LED will light up. The timing chart I have showed below:

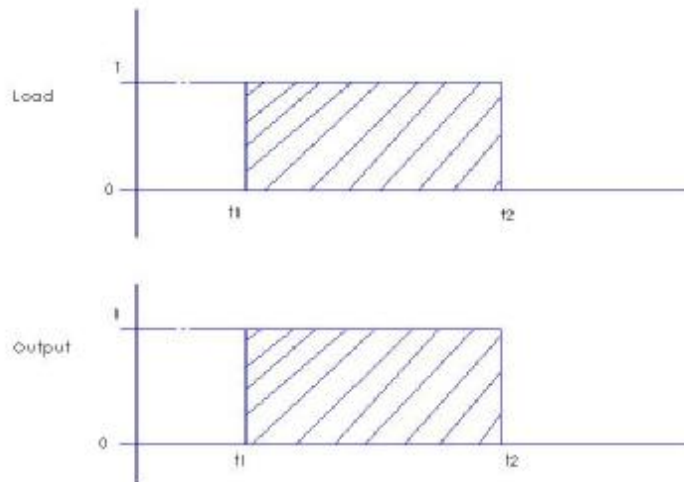


Figure 4.10: The timing chart for EE-SX671 sensors

From the graph above, the shaped area means that the load is blocking the path of the receiver and the transmitter. As the output below, the shaped area means that the output is high. For this case, it means the LED lights up and from here I have come up with a circuit to connect it to the main circuit board. This will be shown in chapter 7 under section 7.2.3.

After finish testing the detecting sensors, I have to do the same for the sensor that are used to detect if there is any things blocking the robot way. As I have select the type of sensor for this purpose to be convergent reflective photoelectric sensor. I have come up with a few tests to check if the sensors works. Below is the diagram I used it for checking if the sensor are in working condition:

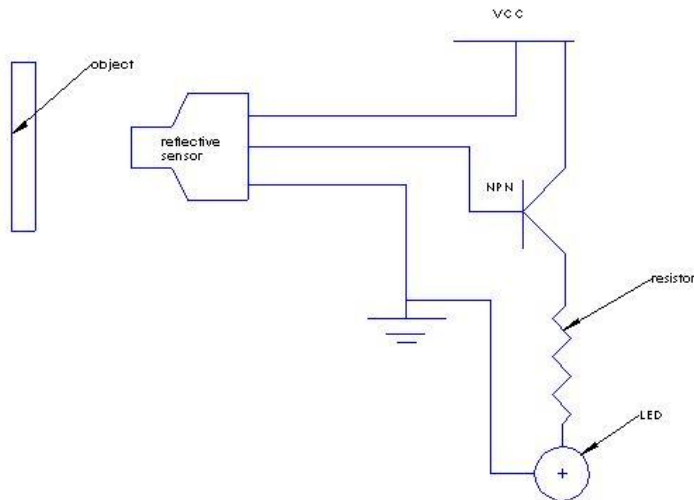


Figure 4.11: Testing circuit for PM2-LH10

The test circuit above show that if the object is blocking the path of the sensor, the LED will light up. However the sensing distance for this sensor is unknown therefore by using this circuit, I can also check what maximum sensing distance is. Below is the table that I have done to test the sensing distance. The results are shown in it too.

DISTANCE(mm)	NO DETECTION	DETECTION
200	X	
100	X	
50	X	
20	X	
10	X	
6	X	
5		✓
2		✓
1		✓

Table 4.1 : Detecting distance result

From the table above, the maximum detecting distance for this sensor is 5mm. Thus the sensor used for my project is not suitable but there are a few ranges for this type of reflective sensors.

CHAPTER 5

COMPUTER FAN

The computer fan are found inside the computer that are used to cool down the computer, usually has one exhaust fan and intake fan, they either suck out the hot air cause by the component out to the surrounding or suck in cooler air from the surrounding into the computer to cool down the component. This way they can prevent the computer from over heating. They are attached to components and are usually used in combination with a heatsink to increase the surface area available for heat conduction, thereby improving the efficiency of cooling. Using this logic, I used it as the suction device for the project.

5.1 Different type of fan

There are a few different kind of computer fan that I will discuss in details in the later section of this chapter. They are listed below:

- PSU fan
- Graphics card fan

From the fans listed above, I will choose the most suitable fan for my suction device.

5.1.1 PSU fan

PSU fans often has a double role, not only keeping the PSU itself from overheating, but also removing warm air from inside the case. Most of them expel the air from the rear, they have two fans, one of them is located at the bottom for removing case air and a smaller one on the back for expelling the warm air at a faster rate.

5.1.2 Graphics card fan

These fans are used to cool the GPU and/or memory on graphics cards. The graphic cards can overheat up to over 140W, more heat than even the CPU so efficient cooling is especially important. The advantages of a graphics card fan are that they are quieter and they can protect a video card from overheating. In order for the fan to cool down such high temperature it required to produce a higher air flow. This means that more air is been suck out of the computer or more air is been suck into the computer to cool the card down.

5.2 Selecting the correct air flow rate

After going through the research, and doing some shopping in my local centre, I find out the suitable computer fan for my project. I have also did some calculation which I have found on the internet and below are my calculation when selecting the correct fan for this project. I have also came up with a circuit design to control my fan and it is shown in chapter 7.

In order to find the correct air flow, I need to know a few things such as pressure in both the vacuum pad as well as other things such as the density rho. The formulas are shown below:

$$\rho = P/RT$$

P = pressure

R = gas constant

T = 27°C

By using the atmospheric pressure in the surrounding, I am able to find the rest of the values, the value for the atmospheric pressure is shown below:

Then we have to find the density of the atmospheric and it is given as :

$$\text{Density of surrounding air} = 1.225 \text{ kg/m}^3$$

Using the first formula, I can put in the values that I have found from the above steps.

$$\rho = 6038/1.225 \times 27$$

$$\rho = 182.55$$

So the density rho in the small vacuum pad is 182.55. To find the same thing for the big vacuum pad, the steps are the same and they are stated below:

$$\rho = P/RT$$

P = pressure

R = gas constant

T = 27°C

The value for the pressure, we can get in from chapter 6 as it has been found. Thus by putting in the value, we can solve the velocity needed.

$$\rho = 28627.8/1.225 \times 27$$

$$\rho = 865.54$$

So the density rho in the big vacuum pad is 865.54. After finding these, we can find the flow rate of each vacuum pad.

To find the velocity of the oxygen, and with the dynamic viscosity of oxygen given as 17.3, we can find the velocity of the air flow:

$$v = \mu / \rho$$

μ = dynamic viscosity of different kind of gas

ρ = rho(also see a list of table)

v = velocity

Given the information above, we can calculate the velocity in both the vacuum pads. First I will calculate the small vacuum pad first. The steps are given below:

$$v = 17.3 \times 10^6 / 182.55$$

$$v = 94768.55$$

The velocity of the small vacuum pad is given as 94768.55m/s

The velocity of the big vacuum pad will be shown below:

$$v = 17.3 \times 10^6 / 865.54$$

$$v = 19987.5$$

The velocity of the big vacuum pad is given as 19987.5m/s

In order to find the volumetric flow rate, we can use the below formula for this:

$$v = \frac{4Q}{\pi D^2} = \frac{4G}{\pi \rho D^2}$$

where

v = velocity

Q = volumetric flow rate

D = diameter of the pipe.

As both the pipe internal diameter I have chosen is 7 mm and with the above formula I be able to find the volumetric flow rate for both of the vacuum pads.

$$v = 4Q/\pi D^2$$

For small vacuum pad, it will be

$$94768.55 = 4Q/\pi D^2$$

$$94768.55 \times \pi D^2 = 4Q$$

$$14.588 = 4Q$$

$$Q = 3.65 \text{ CFM}$$

For the small vacuum pad, the volumetric flow rate is 3.65 CFM

For the large vacuum pad, it will be :

$$19987.5 = 4Q/\pi D^2$$

$$19987.5 \times \pi D^2 = 4Q$$

$$3.077 = 4Q$$

$$Q = 0.769 \text{ CFM}$$

For the big vacuum pad, the volumetric flow rate is 0.769 CFM

By comparing the result, I have select this type of PC fan, the picture of this type of fan are shown below:

As from the specification of the fan, I am able to use it for my project. The specification of the fan will be shown in **Appendix C**.

I have connected this circuit to test the fan and use it to test the suction force cause by the fan speed. The diagram below show how I connect for the suction test.

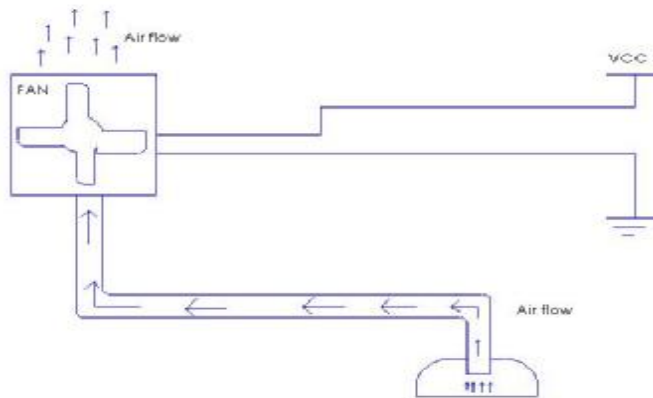


Figure 5.1: Air suction test

The above diagram is the test I used to check two things, one is to check if the fan is in working condition while the other is to check by using this fan is it able to create the suction I need. Although I manage to run the fan by the suction created by the fan is not strong enough to build up the pressure needed. There are a few possibilities, one of them is the power is not strong enough to build up the pressure. Another is because of the vacuum pad, as I used the plastic cup to replace the vacuum pad and there is some gap if it is not place on a true flat surface.

Another other possible reason is the vacuum pad I have modified; there are some air escapes out into the surrounding as well as air from the surrounding entering the vacuum pad. Thus here is the photo of the fan with its circuit control:

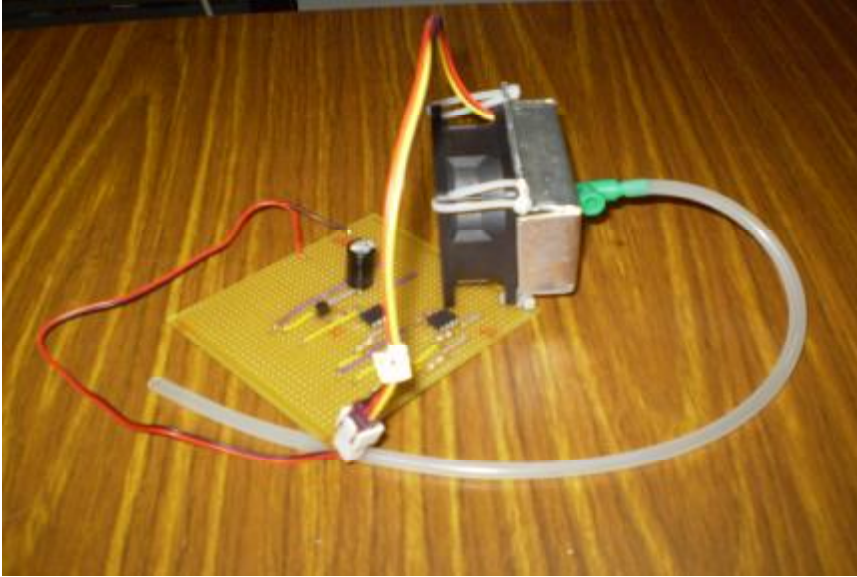


Figure 5.2: suction control assembly

5.3 Whole weight of the robot

After going through the above research, next thing I have to find of the mass and weight for the material that are used to build this robot. I manage to find out the mass of aluminum that is around 2600kg per metre cube. As the size of the aluminum used is not so large, I have calculated each individual parts and the result are shown in the table below:

PARTS NAME	VOLUME	WEIGHT (KG)	FORCE (N)
base structure	2.16 X 10⁻⁵	0.05616	0.551
support bracket	4.69 X 10⁻⁵	0.0122	0.119
fan bracket	3.2 X 10⁻⁵	0.0832	0.816
robot arm	5.65 X 10⁻⁵	0.1469	1.441
Standard parts	-	0.6665	6.538
Total	19.19 X 10⁻⁵	0.83	9.465

Table 5.1: Total weight and total force

The formula used to find the weight is individual parts is given below:

Volume of individual parts = Length X Width X Height

First to find the volume for each individual part, after that we need to find the weight for each individual parts. By using the formula given below, we are able to find the weight of them.

Weight of individual parts = $2600\text{Kg/m}^3 \times \text{Volume}$

After finding the weight, we are able to calculate the total force adding up. Then from the calculation to find the suction power we can compare if it is still within the range. The formula given below is to find the force created by the robot.

Force = mass X gravity force

After finding the weight of the whole robot, things I have to consider are:

- Is the suction created by the computer fan strong enough to the robot
- Will other factor affect on the suction force if they are added to the weight

After I have found out the weight of the whole robot, I have to find the whole force acting down on the robot. Below are the steps used to find the weight acting on the whole robot:

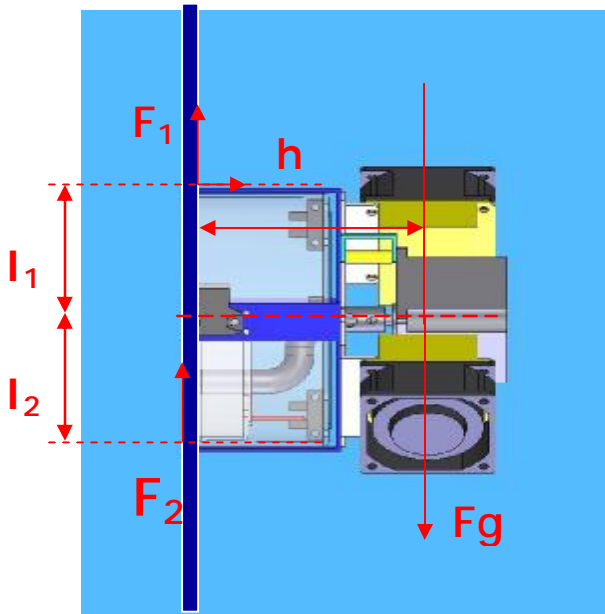


Figure 5.3: Addition force acting on robot

The force for the calculated for the vacuum pad is given as 603.8N while the overall weight /force calculated, it is 24N therefore the overall force for the vacuum pad is much more higher than what is given.

In order to find Force 1, the formula is given as:

$$F_1 = 2F_g h / l_1$$

$$F_2 = 2F_g h / l_2$$

Given for my case, l_1 will be the same as l_2 thus the result calculated is given above.

After going through the whole calculation and results, I am confident to state that the suction force used are able to overcome the weight of the robot even if other factors such as the friction coefficient are added to it. As the suction force created by the fan are much higher than the overall weight added up. Therefore adding the extra or additional force, the result is still smaller than 603.8 N. Thus the fan selected should be able to carry the whole weight.

CHAPTER 6

VACUUM CUPS /PADS

In order to build this robot, we need to know how a vacuum cleaner works as I am using this theory for this project. Now we need to discuss on how the vacuum cleaner works first. Below is the diagram of a vacuum cleaner:

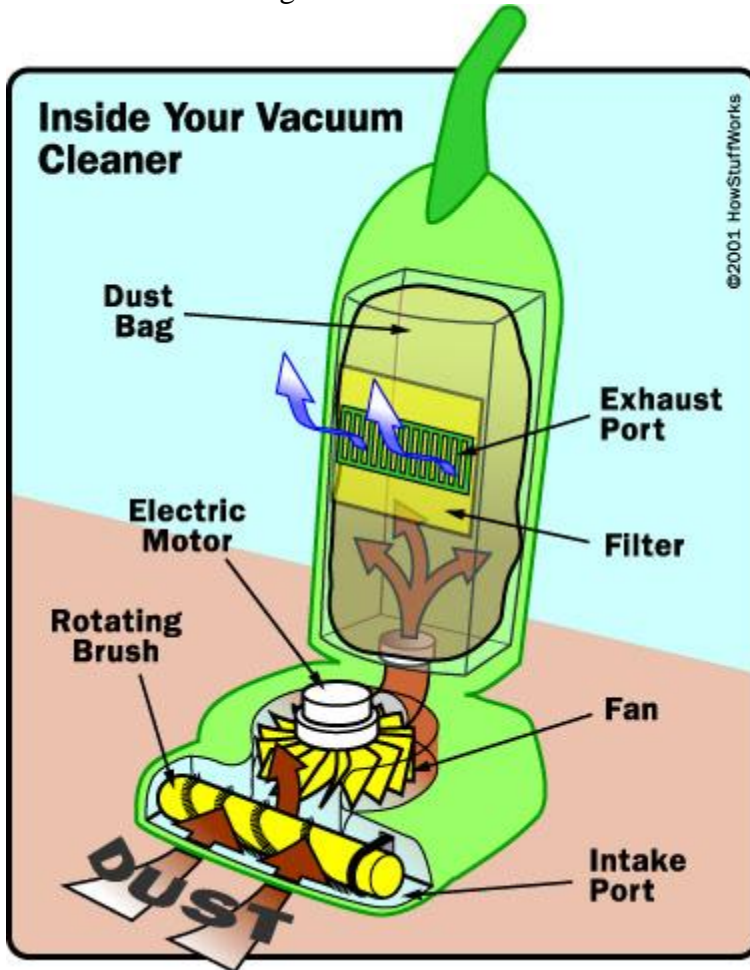


Figure 6.1: internal of a vacuum cleaner

The fan, when switch on, are used to suck in the air from the in take port and transfer to the exhaust port. As soon as the fan is turn on, the pressure level behind the fan is been drop below the pressure outside the vacuum cleaner. This is known as the ambient air pressure which causes the suction inside the vacuum cleaner. The ambient air pushes itself into the vacuum cleaner thus suction is build. A as the fan is running, there will always be a suction force available on the vacuum cleaner.

With this basic idea, I will now work on the theory of the vacuum pad/cup theory. This will be shown in the next section.

6.1 Theory of how the vacuum pads works

The air has been compressed under the pressure apply from the top of the vacuum pad, as the theory of the pressure is that the volume in the enclosure area, is indirectly proportional to that of the pressure, please look at the formula given below,

$$p = \frac{F}{A}$$

where :

p = pressure

F = normal force

A = enclosure area

This means as the area is reduced, the pressure increases, let F to be equal to 1.

Thus as the area inside the vacuum pad is reduce the pressure inside it is greater than the pressure outside the vacuum pad, this pressure difference is achieved by connecting the suction pad to a vacuum generator, which draws air out of the suction pad. When the suction pad touches the workpiece, it forms a seal between the atmosphere and the space under the suction pad. The figure below shows how the vacuum pad works.

The holding force is directly proportional to the difference between the atmospheric pressure and the pressure in the space below the suction pad. The two figures below shows the action that I have explain earlier.

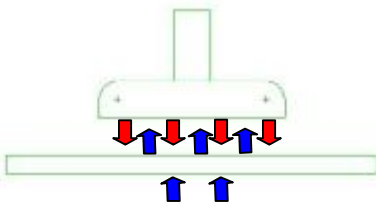


Figure 6.2: The action of the vacuum pad (before Compression)

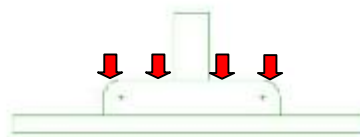


Figure 6.3: The action of the vacuum pad (after Compression)

As for how strong the vacuum pressure is depends on how much air the vacuum moves and how fast it move because the strength of the suction keeps the air turbine spinning when tension increases, All of this is measure or shown by there CFM of the fan or other

vacuum pumps. Therefore the suction capacity of a vacuum is not the most effective of creating the suction.

6.1.1 Calculation of pressure drop

Firstly we have to find the mass of the vacuum pads, both the big and small one. After that we need to know the pressure each of these vacuum pads can build up and from there we can check the flow rate for each of them. The steps to find the pressure needed is given below. By using the atmospheric pressure in the surrounding, I am able to find the rest of the values, the value for the atmospheric pressure is shown below:

Atmospheric pressure is about 10 tonnes per square metre

From here we need to find the volume of oxygen in the small vacuum pad first in order to find the pressure required in it. Here are the steps of doing so:

$$\begin{aligned} \text{Whole surface area of small vacuum pad} &= 2\pi r^2 \\ &= 2\pi(31\text{mm})^2 \\ &= 6.038 \times 10^{-3} \text{ m} \end{aligned}$$

After finding the surface area in the small vacuum pad, we can find the pressure in it.

$$\begin{aligned} \text{Pressure in the small vacuum pad} &= (\text{Atmospheric pressure})(6.038 \times 10^{-3} \text{ m}) \\ &= 603.8\text{N} \end{aligned}$$

From here we need to find the surface area in the big vacuum pad first in order to find the pressure required in it. Here are the steps of doing so:

$$\begin{aligned} \text{Volume of big vacuum pad} &= \pi r^2 h \\ &= 2\pi(67.5\text{mm})^2 \\ &= 0.0286 \text{ m} \end{aligned}$$

$$\begin{aligned} \text{Pressure in the big vacuum pad} &= (\text{Atmospheric pressure})(0.0286 \text{ m}) \\ &= 2862.78\text{N} \end{aligned}$$

Next is to calculate the pressure drop in the pipe that is used for this project, below is the formula for the pressure drop.

As I have found out some of the required valves from above calculation, I am able to find the pressure drop inside the pipe.

The formula for this is stated below:

$$\Delta p = \frac{8\rho \cdot Q^2}{\pi^2 D^4} \left(\lambda \frac{L}{D} + \sum \xi \right)$$

As for the pipe, the drawing for the pipe will be shown below:

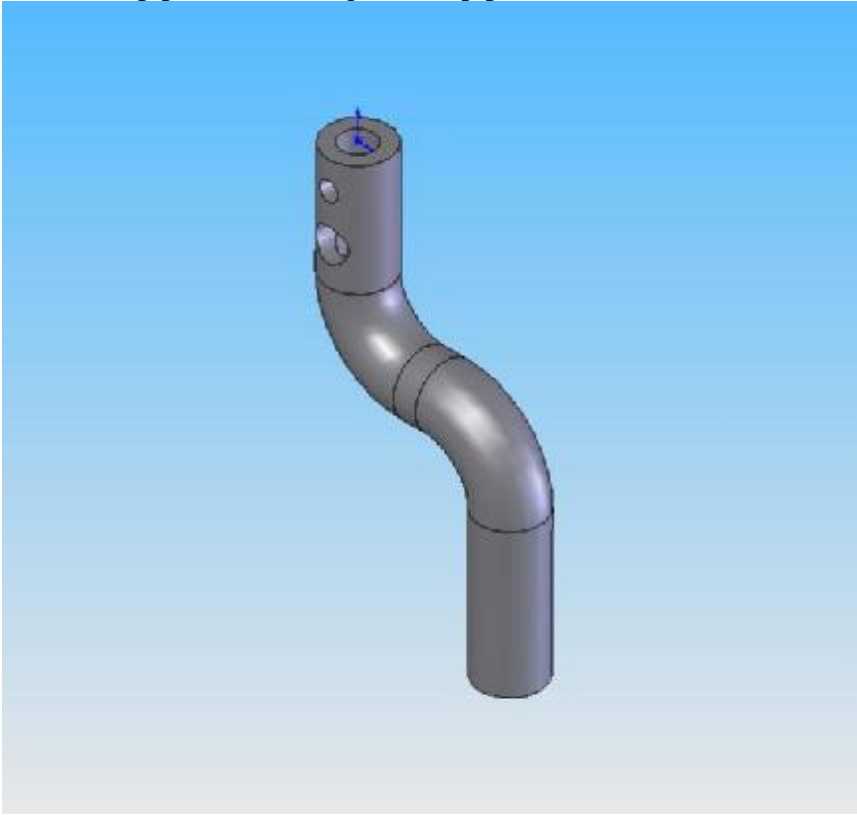


Figure 6.4: Actually drawing of pipe used

As the dimension and the actually physical of this pipe is different, I will be using the pipe which I have manufacture and I will be using this for calculation. As the actually pipe is the length is slightly longer than the one that I have insert above. I do a test for this and compare it with the calculated result. The details will be discussed in the last section of this chapter.

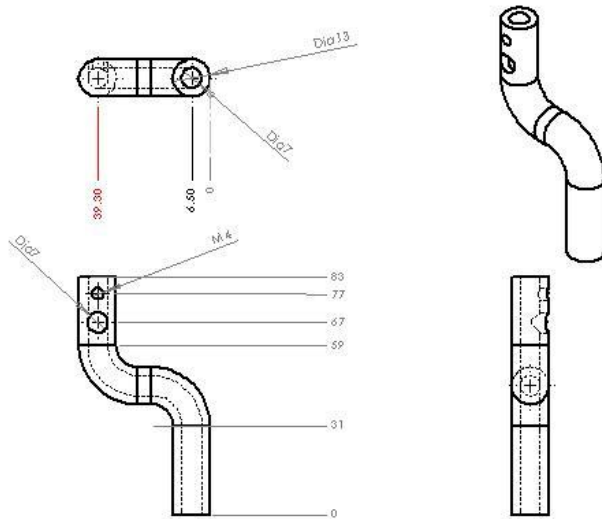


Figure 6.5: The dimension for the actual pipe

By using this formula,

$$\Delta p = \frac{8\rho \cdot Q^2}{\pi^2 D^4} \left(\lambda \frac{L}{D} + \sum \xi \right)$$

where

Δp - pressure drop

ρ - fluid density

Q - volumetric flow rate

D - pipe diameter

λ - friction coefficient

L - pipe length

sum ξ - the sum of minor losses coefficient

Before proceeding, we need to find out the Reynolds number first, its formula is given by:

$$Re_D = \frac{VD}{\nu} = \frac{\rho VD}{\mu}$$

$$\begin{aligned} Re &= 182.55(94768.55)(0.007) / 17.3 \times 10^6 \\ &= 6.999 \times 10^{-3} \\ &= 7 \times 10^{-3} \end{aligned}$$

Re number is high, this means that the inertia forces are superior than viscosity forces therefore the flow is turbulent but if Re number is low, viscosity forces are superior thus the flow is laminar.

After finding this, we can find the friction coefficient of the laminar and the formula is given as :

$$\lambda = \frac{64}{Re}$$

Thus the value for λ is given below:

$$\begin{aligned}\lambda &= 64 / 7 \times 10^{-3} \\ &= 9142.86\end{aligned}$$

$$D_p = 8(182.55) (3.65)^2 / \pi^2 D^4 [9142.86 (150/0.007) + \Sigma \zeta]$$

Here is the photo that I have taken after I have assemble up my robot, It is how it looks like when the robot arm is attached to the same vacuum pad together with its main structure of the robot.



Figure 6.6: Robot partial assembly

6.1.2 Deciding on how many vacuum pads to be used

To determine the number of vacuum pad used, we can used the formula below to calculate it: As so of the values has been found from the previous calculation, we can substitute the value in to double check is it is correct.

$$D = 11.2 \times \sqrt{m} \times c/b \times s$$

where

D = diameter of the vacuum pad

m = mass the vacuum pad

c = no of the vacuum pad

b = pressure

s = safety factor

In this case the safety factor is 4, the safety of 2 is used got horizontal direction. Thus:

$$D = 11.2 \times \sqrt{m} \times c/b \times s$$

$$0.062 = 11.2 \times \sqrt{m} \times 1/b \times 4$$

$$1.384 \times 10^{-3} = \sqrt{m} \times 1/b$$

$$= \sqrt{m} \times 1/0.847$$

$$= \sqrt{m} \times 1.1806$$

$$\sqrt{m} = 1.384 \times 10^{-3} / 1.1806$$

$$= 1.172 \times 10^{-3}$$

$$m = (1.172 \times 10^{-3})^2$$

$$= 1.374 \times 10^{-6}$$

Mass of the small vacuum pad is 1.374×10^{-6} , thus the weight of this vacuum pad is $1.374 \times 10^{-6} / 9.81$ is 1.400×10^{-7} kg/m and we convert this to gram, it is 14g for the small vacuum pad. The actual weight of this vacuum pad is around that 12g. Therefore from the result we can confirm that we are able to used one vacuum pad to do this task as the maximum weight of the vacuum pad did not exceed this value. We can do the same using the above formula to calculate for the big vacuum pad.

$$D = 11.2 \times \sqrt{m} \times c/b \times s$$

$$0.135 = 11.2 \times \sqrt{m} \times 1/b \times 4$$

$$3.013 \times 10^{-3} = \sqrt{m} \times 1/b$$

$$= \sqrt{m} \times 1/9.038$$

$$= \sqrt{m} \times 0.1106$$

$$\sqrt{m} = 3.013 \times 10^{-3} / 0.1106$$

$$= 0.0272$$

$$m = (0.0272)^2$$

$$= 7.423 \times 10^{-4}$$

Mass of the small vacuum pad is 7.423×10^{-4} , thus the weight of this vacuum pad is $1.374 \times 10^{-6} / 9.81$ is 7.567×10^{-5} kg/m and we convert this to gram, it is 75g for the big vacuum pad. Therefore from the result we can confirm that we are able to used one vacuum pad to do this task as the maximum weight of the vacuum pad did not exceed this value.

6.2 Different type of vacuum cups /pads

The different type of material for vacuum pad is shown in the appendix at the end of this report. However for my project I am using any of those vacuum pads as those in the appendix are all standard material. They are no suitable one for my project as the height of the vacuum pad is very small and not suitable for my design as I need a rotating arm in one of the big vacuum pads. The vacuum pads that I have found were too low for the

rotating arm to be fixed in it. However, I did come up with a few new ideas of they are be done and they will be discussed in the next section of this chapter.

However due to the poor material of the vacuum pad I am using now, the suction power is not strong enough as well as other leakage in my design, the robot is not able to create the suction I want. Due to the time I have left, I do not have the time to make adjustment for my project. This part is mainly for the future improvement that I have stated in chapter 9 and they will be discussed in details.

6.3 Troubleshooting

Before I start to troubleshoot the problems, I have face in this chapter, I will do a few test to check the importance of the things listed below:

One of them is the flow rate, I do two test to check the different value of the flow rate. One of them I used running water from the tap directly to fill up a bottle of capacity of 2 liters, I will then used the stop watch to take the timing for it to fill up. Then I will insert the pipe to the tap and turn on the tap and later take the timing for it to fill up the bottle. The diagram below shows the test and the table is the result of the test.

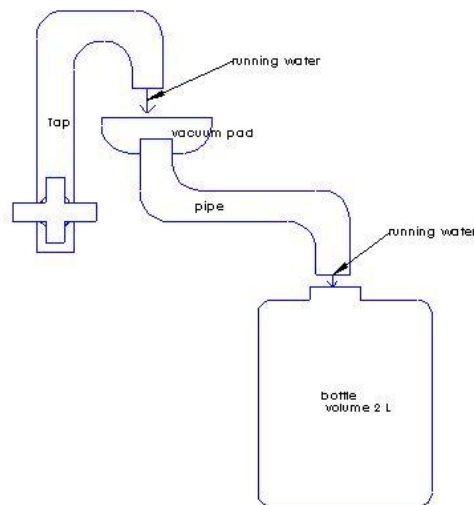


Figure 6.7: The flow rate test (collect from the pipe)

The above is the test for the flowing rate as well to compare it with the another test show below. As one of them is collecting water directly from the tap while the above one show it passing through the pip and the result are show in the table below.

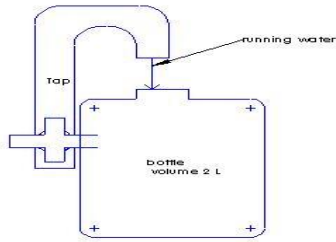


Figure 6.8: The flow rate test (collect directly from tap)

Method used	Time taken to fill the bottle(2 liters)
Running water directly from tap	1 min 24 sec and 60 millisecond
Running water from the pipe(diameter 7mm)	1 min 42 sec
Time difference	17sec and 40 millisecond

Table 6.1: The flow rate result

2 liters = 2000kg

I have round up the water running directly to the tap to 85 second for easy calculation.

$85 = 2000$

$1 = 2000 / 85$

Thus I manage to find the flow rate for one second and the result is 23.529 per sec

Now I take

Below is another I am testing as this I used smoke instead of water. This is to compare the flow rate between the water and the smoke. Below is the figure I have used to test the flow rate.

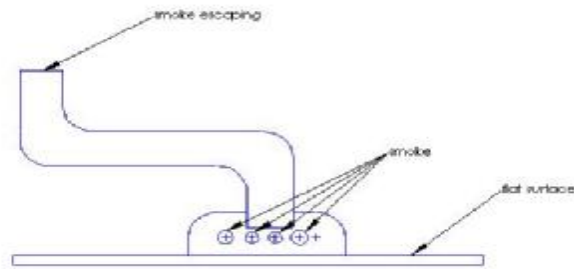


Figure 6.9: The flow rate test (used smoke for test)

As soon as I start this experiment, I saw smoke coming out of the end part within a few seconds, therefore I am not able to come out a table to collect the datas. That is why I am using the data collected from the water experiments and the formula I have found in the internet to calculate the results. I am using the results that I have calculated to assume the air flow rate of the smoke. Using this value I am able to continue with the sizing and assembling of this robot as well as to continue with my project. The sizing of the fan size is always using this as a reference. Therefore there may have some error when doing this.

CHAPTER 7

ELECTRICAL /MECHANICAL DESIGN

For this chapter, there will be two type of software to be discuss, one of them is to used to design the circuit diagram will the other is use to design the hardware part. Both the software will be discussed in the next section in detail of this chapter.

7.1 Type of software used

Firstly, I will discuss about the software I used for designing the hardware, the software I want for this is known as the solid work. As I have been using this software for my job and I find this software is much useful as compare to AutoCAD software. It is much user friendly as compare to other software.

As for the software for drawing the circuit diagram, I will be using PROTEL. We have used this software in one of our modular. So I am putting what I have learnt into this project. This software is also much user friendly as compare to other brand of software that are used to design circuit diagram.

7.2 Circuitry Diagram

I have bought a few photomicrosensors and in order to make sure this sensors works, I have came up a simple circuit diagram to test of the sensors to check they are in working condition. The photo I have taken are shown below. I put the aluminum metal between to see if the LED light up else the LED should not light up if there is nothing blocking it.

I have also came up a simple diagram to test the fan is it in working condition. The photo below show what I have done to test the fan. Together with it is a simple circuit diagram showing how I connect them up. With this simple diagram I am using now, I will modify it later for my overall fan control that will be shown in the next section.

7.2.1 Diagram for motors

I have also came up a simple circuit diagram to rotate the stepper motor in the clockwise and anti-clockwise direction. There is photo that I have taken of how the circuit is form, both of them are shown below After that I will combine this circuit with other circuits that I have mention here as well as combining them together to form a big circuit board. The figure is the circuit diagram I have mention before:

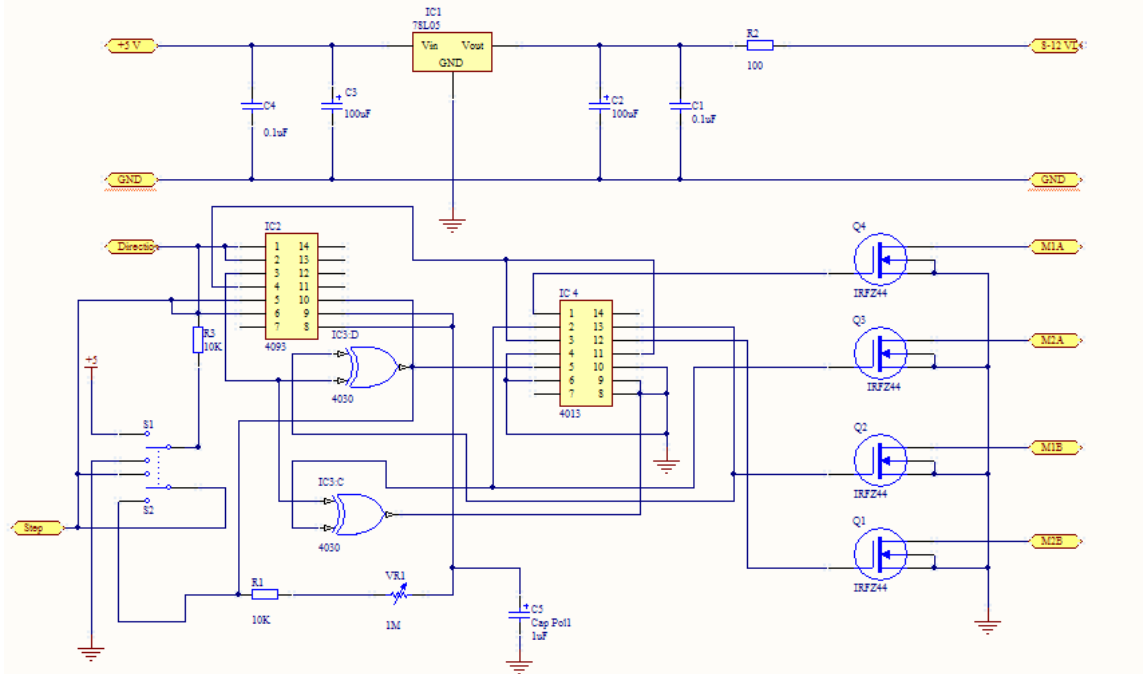


Figure 7.1: Circuit to control motor

Above is the schematic diagram for the motor control. To select where the full steps or half step from the step command and to control if it turn clockwise or anti-clockwise, it all depends on the input from the direction port show in the diagram above. IC chip HCF4093B is the Quad 2 input Nand Schmitt triggers, its purpose is to check when the input is higher than a certain chosen threshold, the output is high; when the input is below another (lower) chosen threshold, the output is low; when the input is between the two, the output retains its value. The output retains its value until the input changes sufficiently to trigger a change. Below is the picture which I have taken to try out the above diagram:

After testing the motor, I will try to test out my other circuit which I have understand or come up my own design to test my other components such as the fan and the sensors. The circuit diagram will be shown in the next section, explanation will be given to each of the circuit.

7.2.2 Diagram for computer fan

After trying on the simple fan circuit above, I using the theory to combine it with other circuit for this final project. As I need to turn on and turn off each individual fan from time to time, I have to build to two different circuits, each of them to control one fan. One of the fans is to control the turning on or off of the suction force for the small vacuum pad while the other is to control turning on or off of the suction force for the big vacuum pad. Therefore the physical size of the fan will be different however the circuit of them will be the small. The diagram for the circuit board of the fan will be shown below:

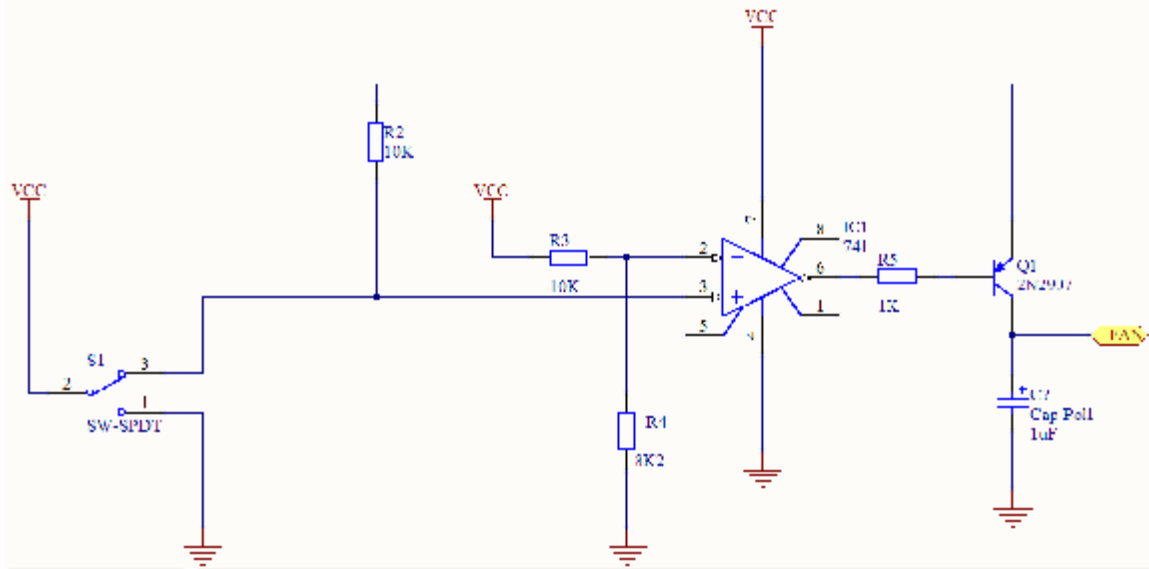


Figure 7.2: Circuit to control fan

The circuit that I design above, actually the fan depends on whether switch 1 is turned on or off. As the op-amp compares the voltage in pin 2 and pin 3, if the voltage in pin 3 is higher than that of pin 2, then it will take the voltage from Vcc. Then it will turn on the fan; else if it is lower than pin 2, then the voltage output will be zero. In this case, it will turn off the fan.

7.2.3 Diagram for sensors

There are two types of sensors I will be using; therefore, there will be two different types of sensor circuit diagrams. I will show the sensor circuit diagram for detecting the vacuum pad first. As there are four of them, this means that there will be four circuit diagrams for each of them. Below diagram will only show one of them. As the diagram will be linked to the microcontroller of the 16F690 and sensors are used as inputs, these sensors are used to control the movement of the stepper motor, which in turn controls the robot rotating arm. Thus, we can see from the diagram that as long as the sensor did not detect anything, it means it did not reach its position; it wants to go yet. If the position is reached, it will stop and wait for the next command for it to proceed to the next instruction. However, when the robot arm needs to rotate to 180°, the command in the software will tell the robot arm that it will reach either the left or right sensors first and it will ignore them and carry on even if it gets its first detection from the detecting sensors. Diagram for it will be shown below:

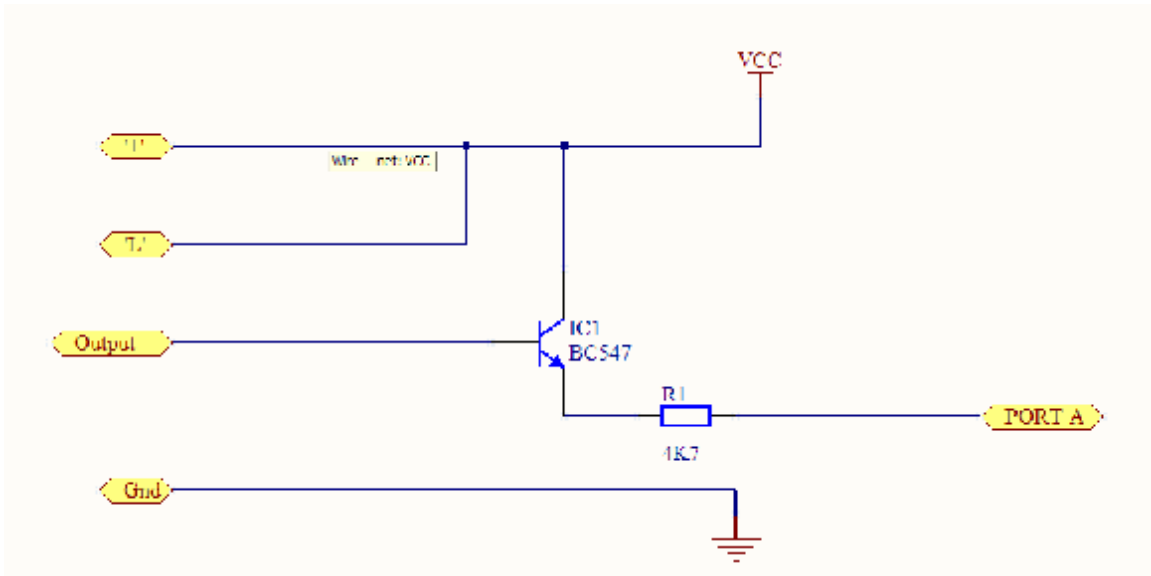


Figure 7.3: Circuit to control detecting sensors

The other type of sensors is the sensors used to detect anything blocking the robot path. As two of the sensor will be used, therefore there will be two circuit for them. These sensors are also input that are used to control which direction the rotating arm suppose to rotate. Either they are in the clockwise direction or anti-clockwise direction. The diagram for these circuits will be shown below:

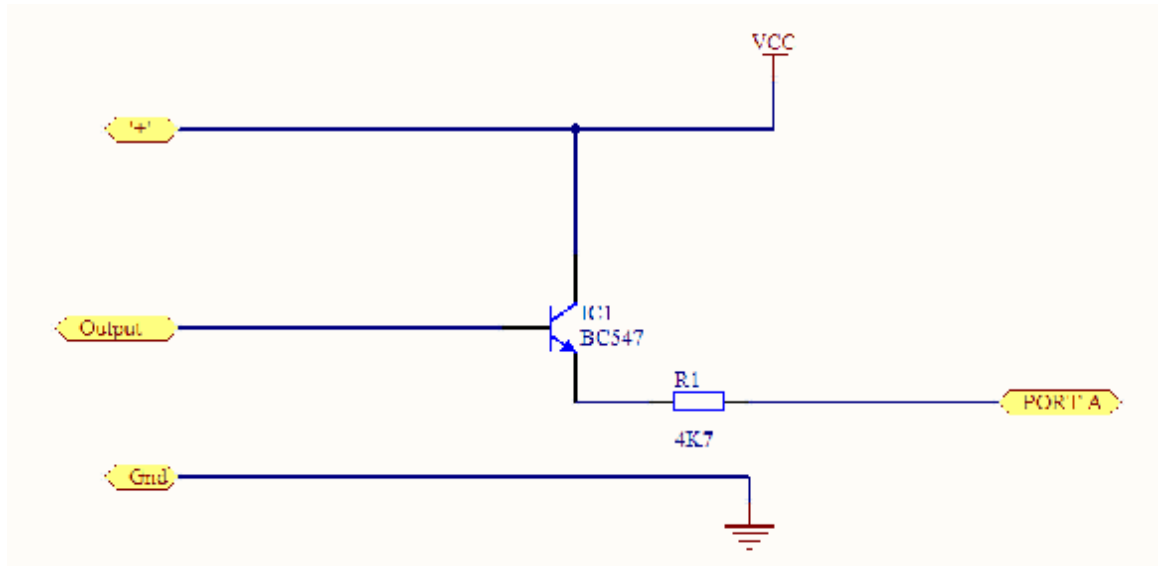


Figure 7.4: Circuit to control PM2-LH10 sensors

7.2.4 Overall diagram for my project

From the above diagram and connection, I have roughly know how should I connect them into one big circuit diagram, however after I have fabricate them out, they need not

need to be in one big circuit board as there are not much space left for my robot. As I have forgotten to leave a space for my electric part. The diagram of my vacuum suction will be shown below, as this is only one of part of my circuit diagram for this project:

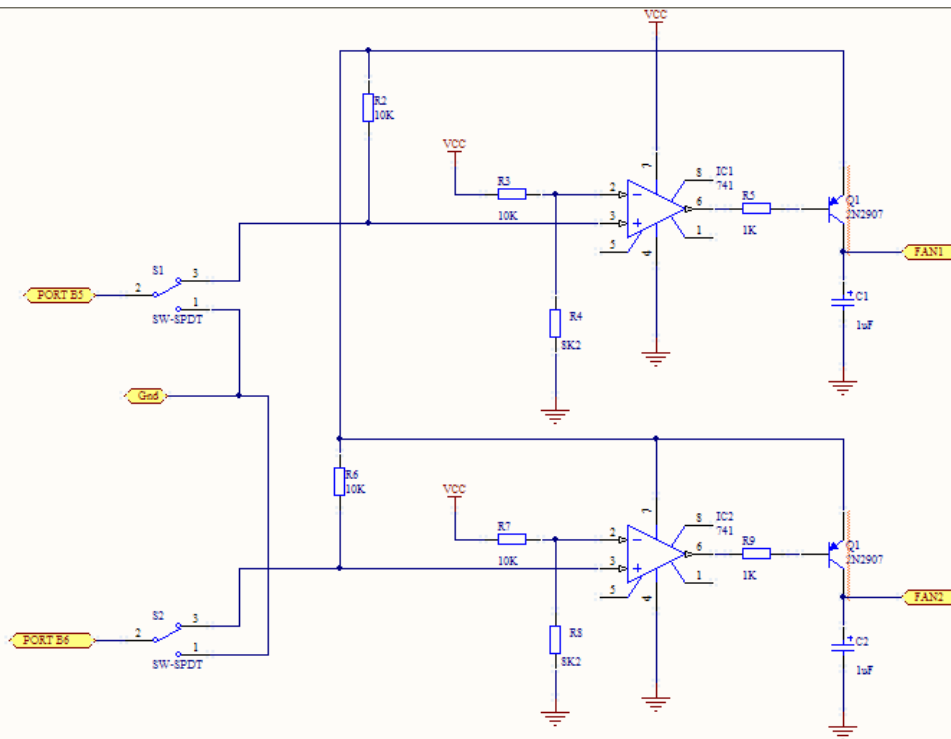


Figure 7.5: Fan circuit for project

Below circuit is for the two type of sensors I am using. But here I show the circuit for the detecting sensors first. As there are four of them together, I have combine four of them into one circuit board. The circuit diagram is show below:

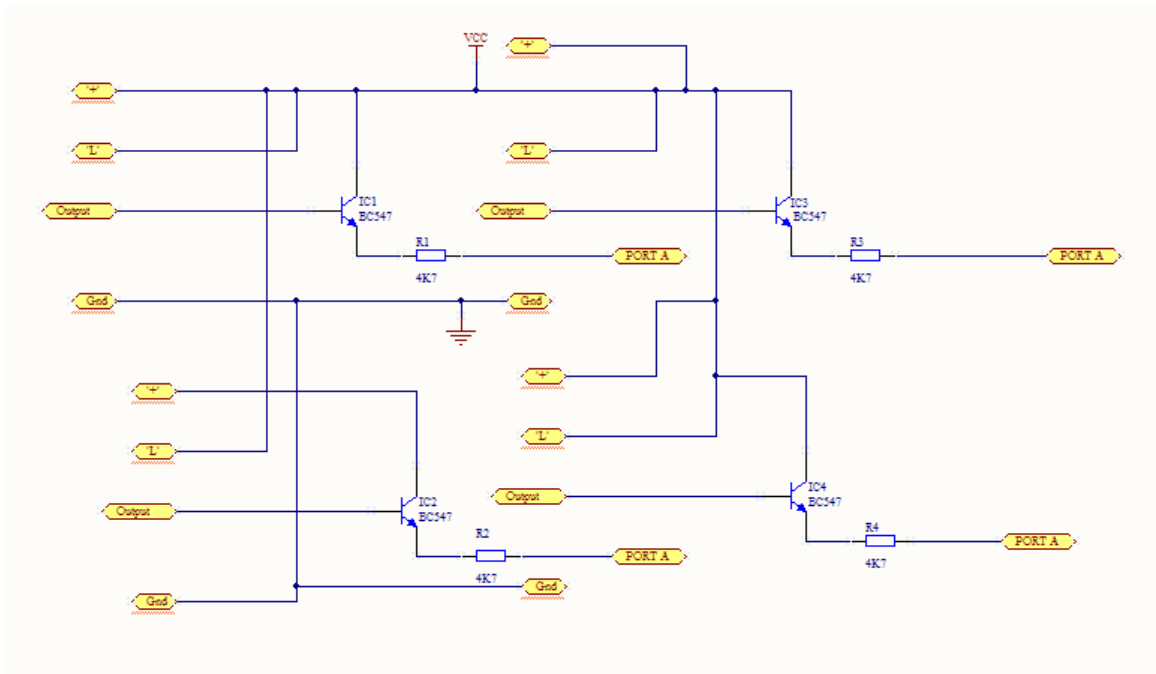


Figure 7.6: Detecting sensor circuit for project

The above circuit is only used for the detecting sensors, as for the sensing sensors, the circuit will be shown later in this report. As for this type of sensors there are only two of them only, thus the total sensors will be shown below here:

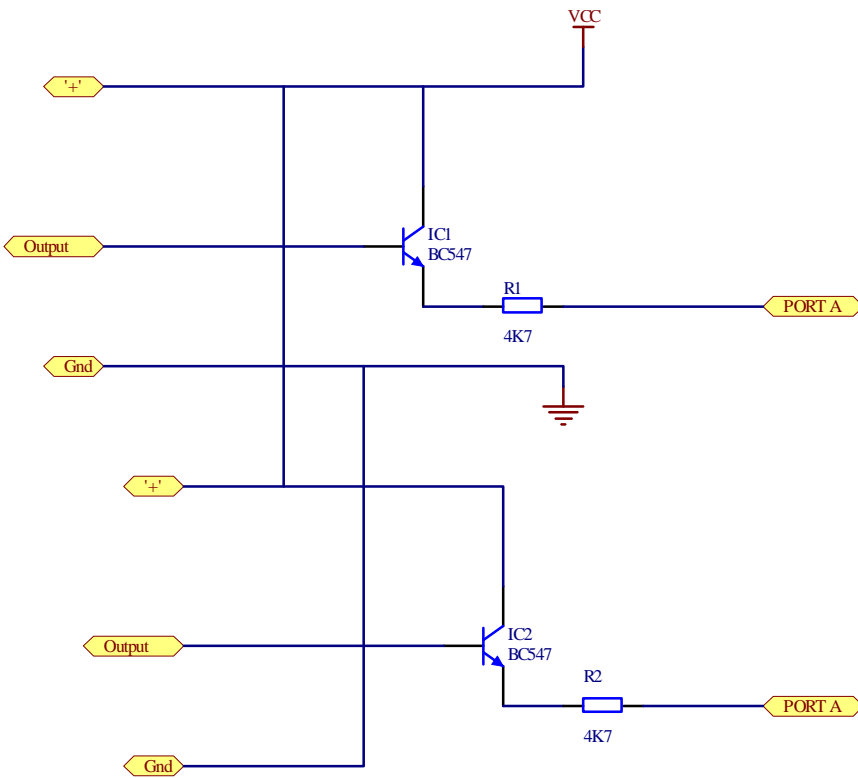


Figure 7.7: PM2-LH10 sensor circuit for project

After going through the circuit diagrams of the overall project, now I should go into the discussion of how my mechanical design should be. The next section will show you how I have fabricate each part. Just like the electric design, I will break each of them into small parts and I will rough give a rough idea how I have design them. The next section will show the dimension of each small parts as well. After all individual parts have been show, the overall assembly of the robot will be shown in 3 dimension too. As I have stated earlier the software for designing all the parts are drawn by solidwork software. All this will be discussed in detail in the next section.

7.3 Mechanical design

All the hardware design for the project will be discussed in this portion of this report. Material used for all the hardware is either aluminum or standard parts that are bought from the shops I have visited. I have also created parts to strengthen the robot as well as to join different parts together. The first thing that I will go through is the ‘bowl’ that I used them as vacuum pads. Modification has been made to each one of them.

7.3.1 Design for small vacuum pad

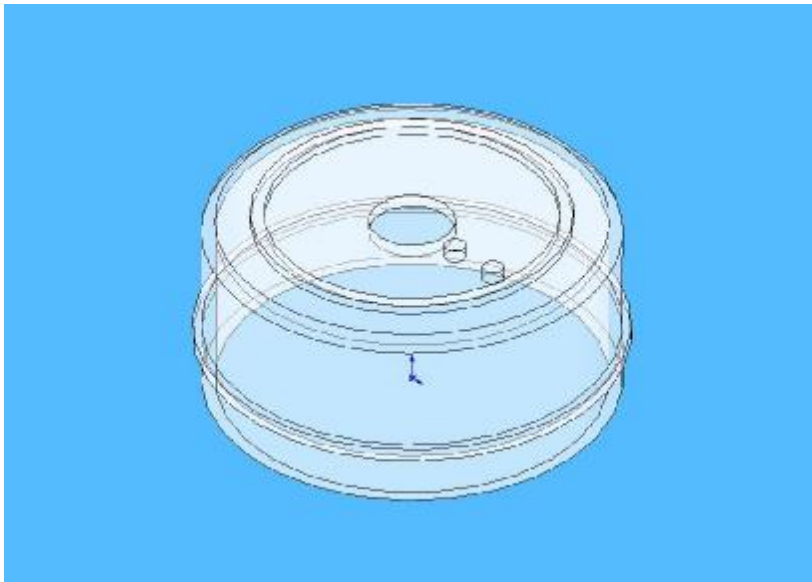


Figure 7.8: Small vacuum pad drawing

The above picture is the item I will be using for smaller vacuum pad. As this is a standard that I have bought from the shopping centre, however, I have make some modification to allow it to be used for my project. Two things will be join to this part, one of them is the pipe that allows the air to escape as soon as the fan is turn on. The other part that will be attached to it is the detecting device used for the position sensors. As this part will rotate

with the motor arm, the detecting device will stop as soon as the sensors detect it. All the rest of the parts will be shown below. The dimension of the smaller vacuum pad will be shown on next page.

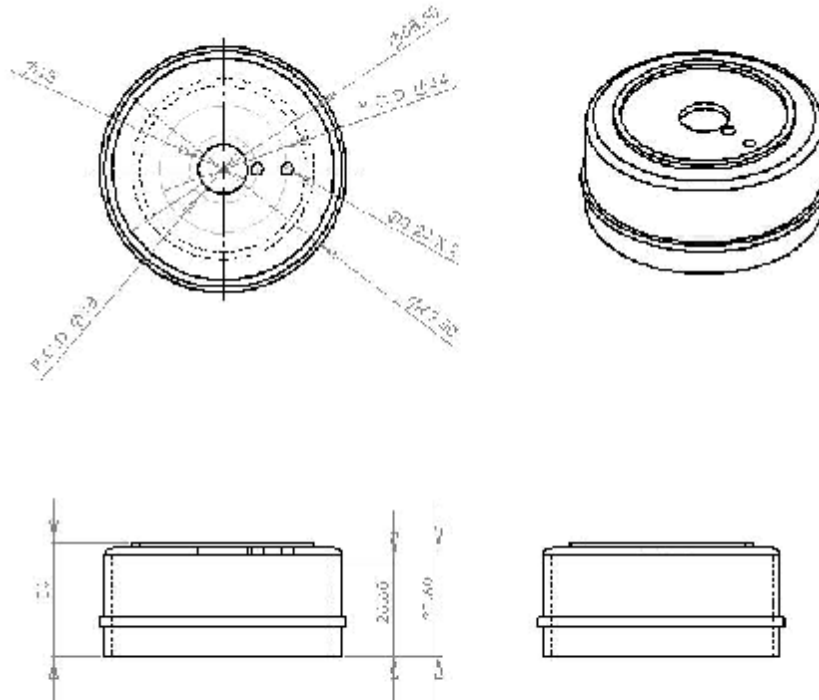


Figure 7.9: Dimension for small vacuum pad

7.3.2 Design for big vacuum pad

Next will be showing the picture of the big vacuum pad:

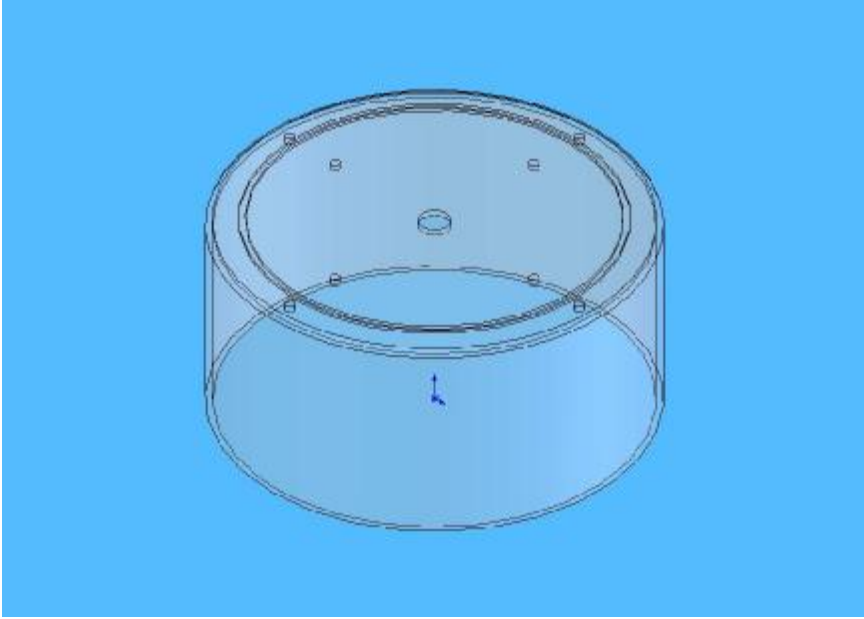


Figure 7.10: Big vacuum pad drawing

This is the main vacuum pad, the smaller vacuum pad will be inside of this part. As this part is also bought from the shopping centre and modification has been made to it for the purpose of my project. The parts attached to this part include the motor arm, detecting sensors as well as the structure. This will be the latest design, as the dimension of this part will be shown below:

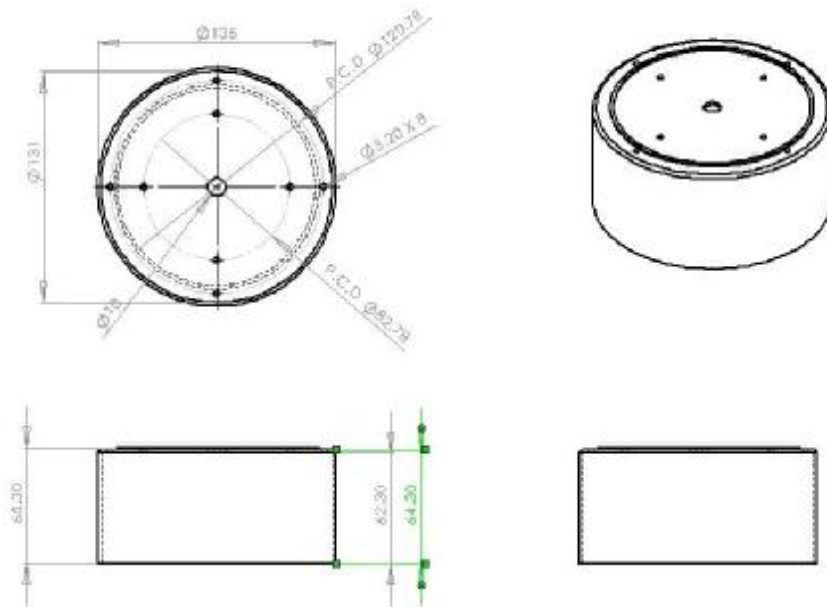


Figure 7.11: Dimension for big vacuum pad

7.3.3 Design for structure

Next will be showing the picture of the main structure:

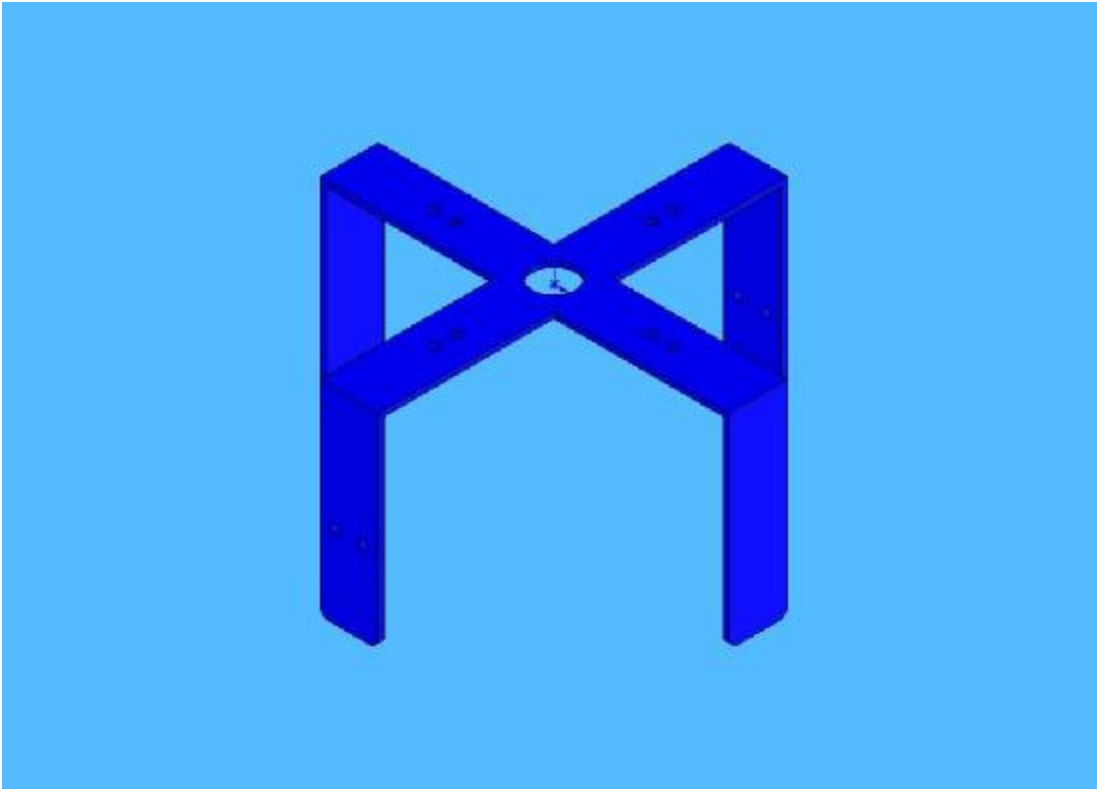


Figure 7.12: Structure drawing

For the figure above, the material I will be using is aluminum as this is where the detecting sensor will be mounted on, this is also the main structure of where the other parts will be attached to. Part of the stepper motor will be attached to here as this will rotate the main vacuum pad when the pressure is cut off. The dimension for this part is shown below as I will need to fabricate this part for the project. The other parts will also be fabricated therefore the dimension will be shown in the later part of the report.

The picture show above is the detecting sensor bracket, this is to secure the infrared sensor to. As for the material wise, it will be the same material as the main structure. As this cause the overall of the robot to be greatly reduce. The dimensioning of this part will be show below:

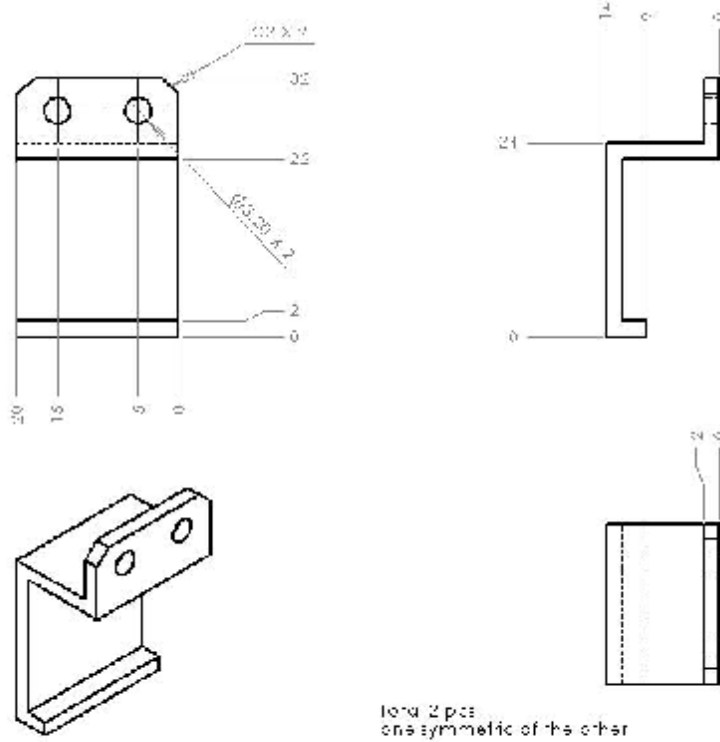


Figure 7.15: Dimension for Sensor bracket

7.3.5 Design for support bracket

Next will be showing the picture of the support bracket:

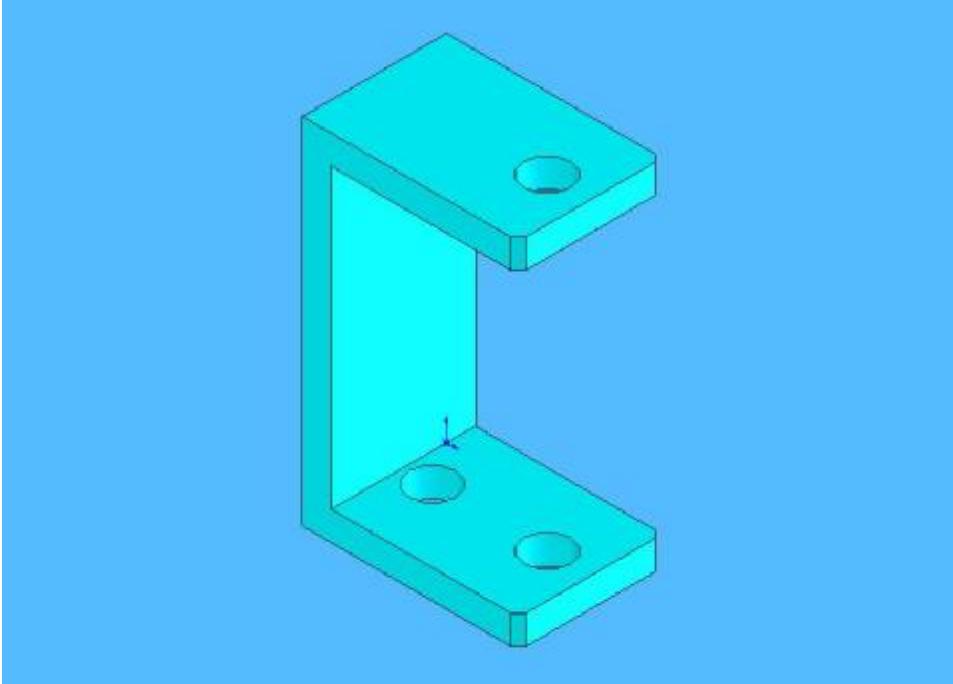


Figure 7.16: Support bracket drawing

The picture show above is the support bracket, this is to secure the stepper motor to the structure base. As for the material, it will be the same material as the main structure. As this cause the overall of the robot to be greatly reduce. There will a total of 4 pieces of this part. The dimensioning for these parts will be show below:

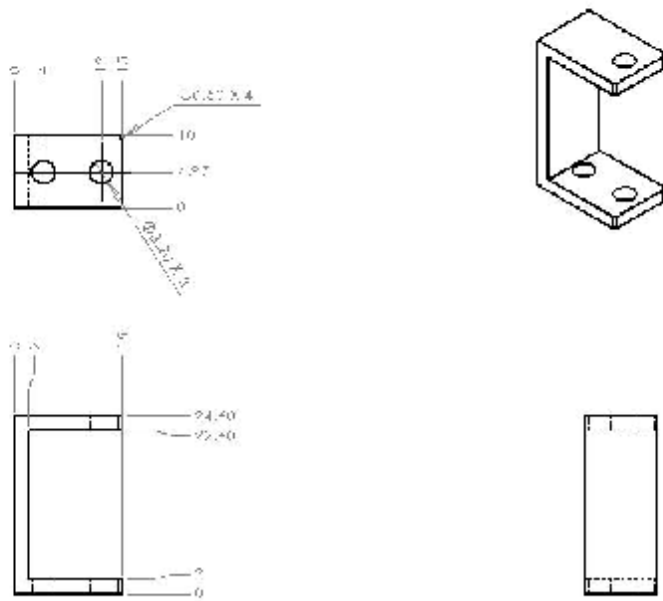


Figure 7.17: Dimension for Support bracket

7.3.6 Design for support beam

Next will be showing the picture of the support beam:



Figure 7.18: Support beam drawing

The picture show above is the support beam, this is to secure the stepper motor to the

structure base. As for the material, it will be the same material as the main structure. As this cause the overall of the robot to be greatly reduce. There will a total of 4 pieces of this part. The dimensioning for these parts will be show below:

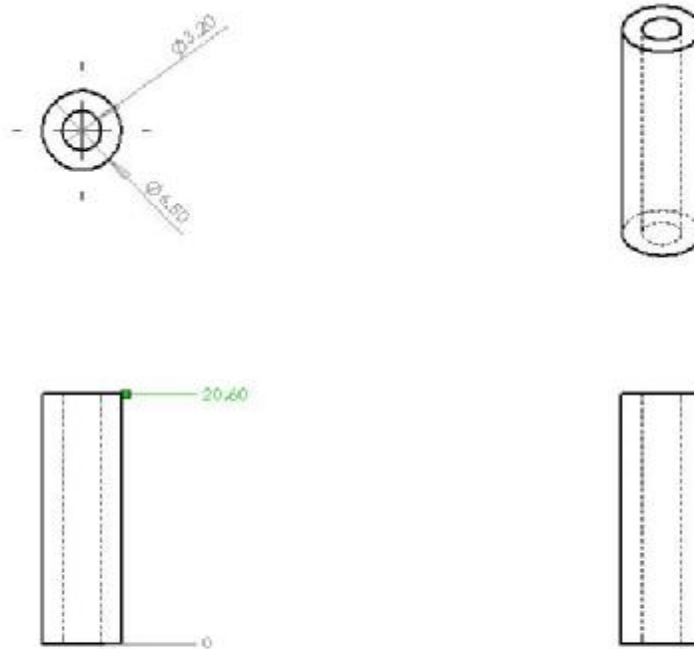


Figure 7.19: Dimension for Support beam

7.3.7 Design for fan bracket

Next will be showing the picture of the fan bracket:

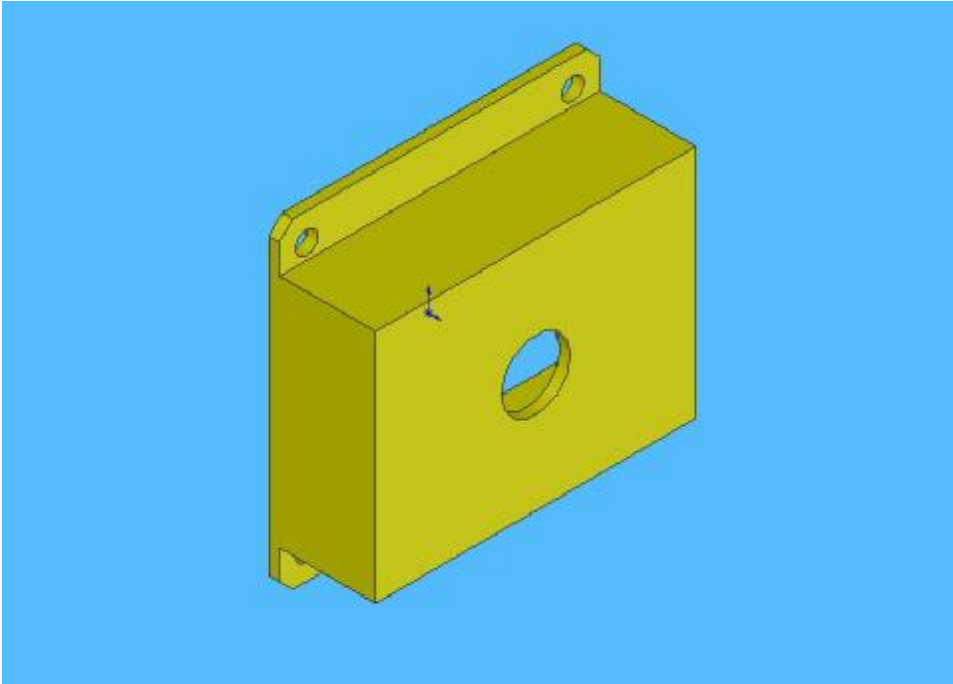


Figure 7.20: fan bracket drawing

The picture show above is the fan support, this is to secure the fan to these part. As for the material, it will be the same material as the main structure. As this cause the overall of the robot to be greatly reduce. There will a total of 2 pieces of this part. The dimensioning for these parts will be show below:

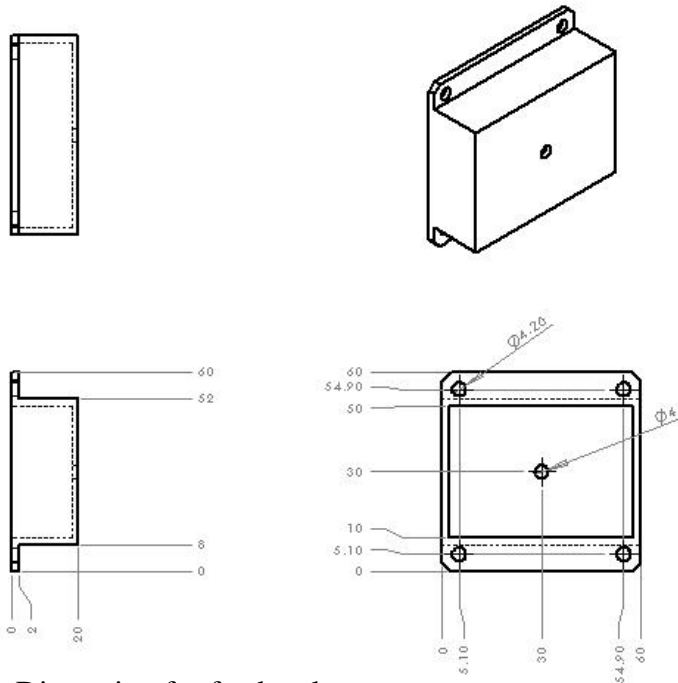


Figure 7.21: Dimension for fan bracket

7.3.8 Design for fan support

Next will be showing the picture of the fan support:

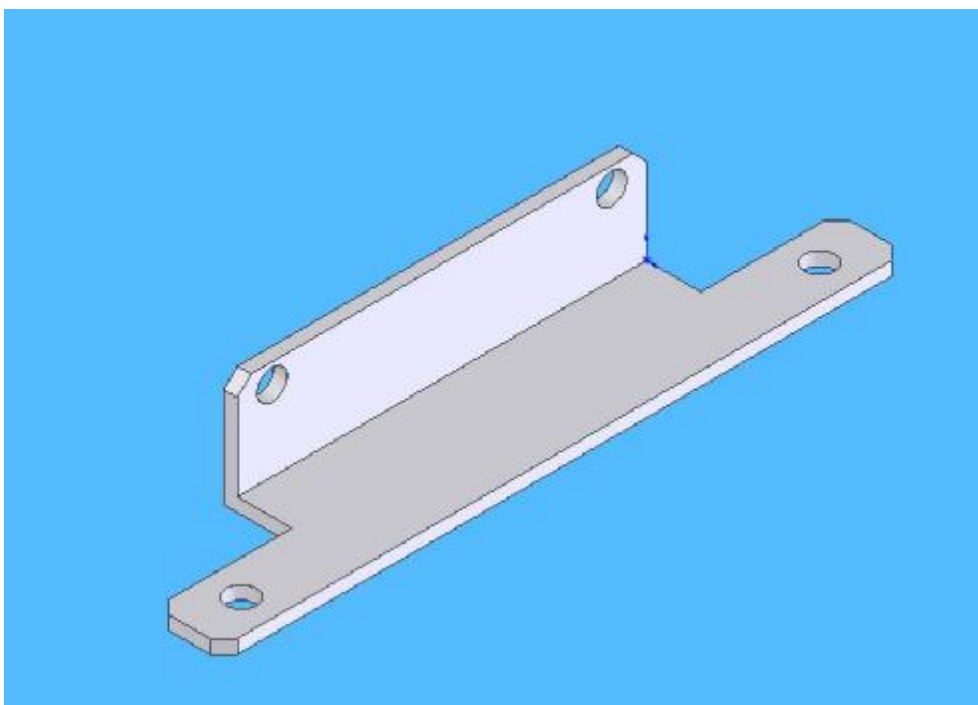


Figure 7.22: fan support drawing

The picture show above is the fan support; this is to secure the fan bracket to the structure base. As for the material, it will be the same material as the main structure. As this cause the overall of the robot to be greatly reduce. There will a total of 2 pieces of this part. The dimensioning for these parts will be show below:

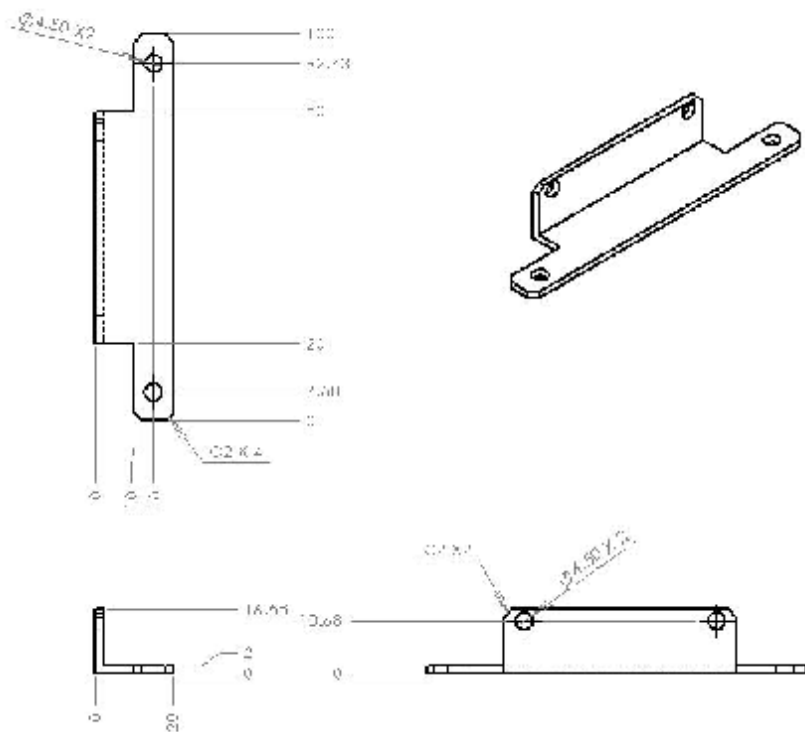


Figure 7.23: Dimension for fan support

7.4 Overall design

After doing most of the parts, I have assembly all the parts together. So the picture of the whole assembly will be show in the next page with the bottom view, isometrix view , the front view and the side view. All this will be show next.

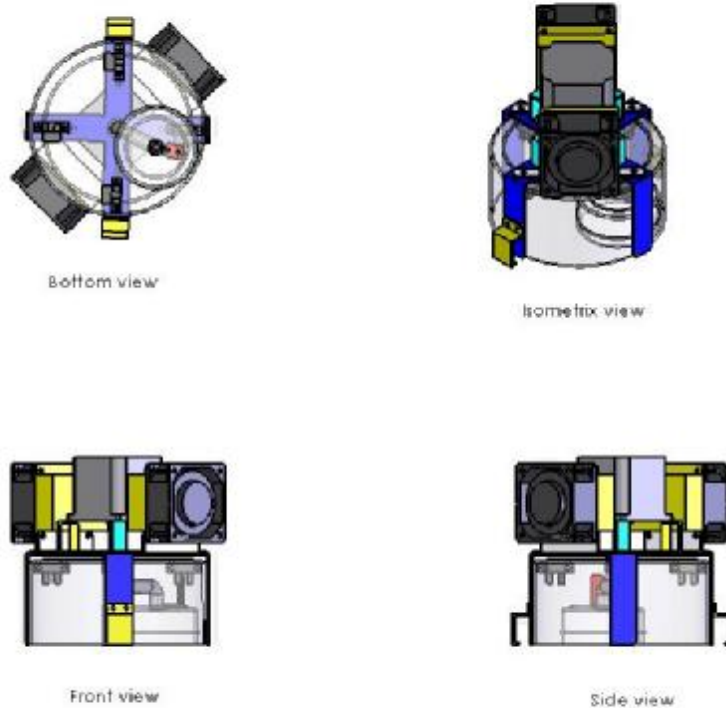


Figure 7.24: Overall drawing

7.5 Troubleshooting

After doing and assembly the fabricated parts, I discover that there are few defaults in my hardware design. They are listed below:

- The boxes connecting to the fans
- The opening in the pipe
- The pipe connecting to the motor

First I will discussed the boxes that is connected to the fan, I have design a part to fix to the fan. As the box I used to cut up is a plus sign then I bend it into the shape of the box, the corners I did not use welding instead I used those silicone to glue over the four corners, thus causing some leakage in this part. Due to this problem, the air pressure could not build up therefore the suction is not able to work. The problem has been shown in the figure below:

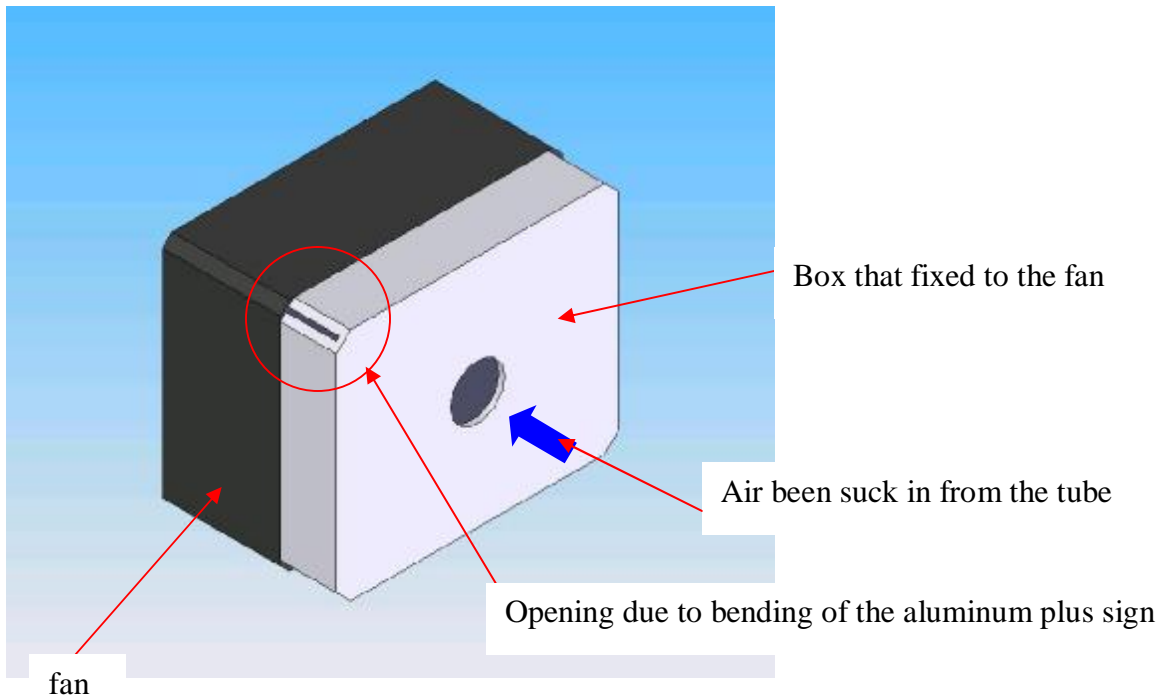


Figure 7.25: Error make for fan bracket

The opening in the pipes will also cause problems to the whole robot as I have to connect a tube in the pipe so that the air in the surrounding will not enter into the tube thus causing the pressure not able to build to its required standards. This will cause more and more air leakage for the whole robot. This is one of the reasons why the robot is not able to work.

As can be seen in one of the photo that I have taken, the pipe is connect to the rotating arm of the motor, it will cause problem as the tube will be withstand as the length of the tube has to be long enough for the moving arm to complete its cycle. If it is unable to do so, the tube will disconnect either from the fan connection or the suction area. In worst case, the whole robot will hand or the rotation arm is limited to a certain angle.

CHAPTER 8

FUTURE DEVELOPMENT /IMPROVEMENT

8.1 Improvement in the software programming

PIC programming is a new program I am using therefore there will be a lot of improvement that I can make in it. Thing that I can change are listed below:

- To reduce the whole programming command.
- Used better command to replace the one I am using now.

As I have stated above, one way of reduce the programming commands such as to replace them with better command. This is point two, as using better command will also reduced the overall programming language. This way, it will reduce the address been used up. Thus much easier to troubleshoot later on in the project.

One example for point two that I mention, is to used clear command or set command instead of loading the command with all zero or with all ones onto a PORTs or TRIS address. For my case I have tried to do so for a few of them for example to clear the data in PORT B and PORTC instead of using the move file command.

I have listed them below:

```
bcf    03h,5      ;Go back to Bank 0
bsf    03h,5      ; Go to Bank 1
clr    TRISB      ; move 0000 0000 onto PORTB, setting all
                    ; upper bits to be outputs
bcf    03h,5      ; Go back to Bank 0
bsf    03h,5      ; Go to Bank 1
clr    TRISC      ; move 0000 0000 into PORTC, setting all
                    ; pins to be outputs
bcf    03h,5      ; Go back to Bank 0
```

For my programming there is another problem I have come across when I was troubleshooting them, one of them is the position sensor. I will discuss the detail in the next section.

8.1.1 Adding a delay timer at every position

The problem for this is that the motor will stop as soon as the sensor detect that the vacuum cup has reach the position. But as the vacuum cup is round, the sensor detect as soon as the circumference of the vacuum cup has reach, the motor is not stop at a 90° but at an angle slightly smaller than the right angle. This is not purpose to be, as I have program to stop the motor at right angle. Thus this caused the programming to have some problem. As I have figure out and to present them in a better understanding condition, I have draw them out in the figure below:

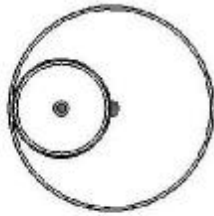


Figure 8.1a: Correct position

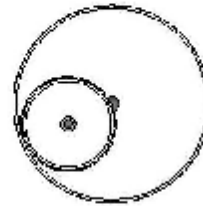


Figure 8.1b: Incorrect position

As from the figure above, we can see that figure 8.1a is the correct position but figure 8.1b is out of the position. As these figure show the motor rotate to the left but is out of position. The same will occurs at 180° and to the right 90° . The reason for this has been method before and it is state in front in this chapter.

There are two solutions for this problem, one of them is to add a delay timer for every position sensors. As adding the timer, the sensor will take some time for the motor arm to react and by choosing the correct timing, it will stop at an right angle. This way the motor arm is able to stop at the position I have intended to in the first place.

Another way is not as good as the first solution, it required to position the sensor at an angle that is larger than the right angle. The same principle occurs, but due to the angle is larger than right angle, the motor arm is able to stop at the position we want. But the design is not so good as compare to the original design.

8.2 Improvement in the hardware design

For this project, the robot weight has been greatly reduced as I has used mostly plastic and aluminum. However, the overall robot design can be improved. One of them is the motor arm, I can reduce the pipe size as well as the method I have inset the vacuum tube onto the pipe. This can be improved; one of them is custom make the robot am with the vacuum tube in it thus not allowing any air from interefecing with the air being suck out. There are other problems with this design, thus it can be improve further. Here are the list I have listed below and they will be discussed in detail in the later part of this section:

- Improve the method used for joining
- Improve the quality of the vacuum pads
- Selecting of different type of vacuum device

8.2.1 Improve the method used for joining

For this project, most of the joining of two parts together is mainly using screw and nuts. However certain fabricated parts need to use other methods to do for example welding. One good example is the fabricated part, box, used for fixing it to the fan. The raw material I bought is a standard aluminum plate, I need to fold them in order to fabricate it into a box. As the bend all the side up, I forget the most important process that is the need for it to weld up. As there is all lot of different welding methods and I do not have this skill but I did some research on the internet and from there I thinkwelding is the best for these parts. There will not be any opening in the corners; therefore no air will be able to escape if the welding is done correctly. I have put the different methods of welding in this report so that in future, students may use this method for fabricating this part.

8.2.2 Improved on the quality of the vacuum cups

During the building and testing of this project, I found out there will be problem for this robot to work. Firstly, the software programming has not been tested for this project. Although I have key in the software code for this robot, I have problem transferring data to the ic chip. But now I will discuss about problem on the vacuum pad, the material used to replace the vacuum pad are plastic bowl that I have bought from the market. The first problem with this is that the friction coefficient is not as good as compare to those that I have research out. Secondly, the overall volume is too large. It required a stronger vacuum device than those I have replaced by the computer fan. This problem will be discussed in the next section. Furthermore as I am fabricating the parts or modify the old design due to misalignment, there are more holes in this parts that allow air to escape in and out of each individual vacuum pad. Therefore I came up this idea to modify the smaller vacuum pad that I have done for this project. This idea will be discussed in the later part of this report, under the future improvement chapter.

8.2.3 Selecting of different type of vacuum device

In this section, I will have two solution for these vacuum device. One way is to continue using the computer to build up the pressure required but to improve on the material for the vacuum pad which I have mention previously. The figure will be shown in the chapter as mention in section 8.2.3. To make my explanation better the picture below will show below here too:

As can be seen in the picture above, by adding a vacuum pad inside the vacuum pad which my current design is, this way the suction take place is not within the whole volume of the vacuum pad but it is in the small vacuum pad in it. In this way, I can used back the computer fan as the vacuuming device. The advantages of this design will be discussed in the next chapter.

8.3 New things added

In this section, new things will be discuss such as new resource need to be done or new

skill to be learn in order to carry on with this project. One of them is to do research on different type of welding for this project. Unless some of the parts are fabricated by other person, else we need to know this in order to fabricate all the parts to be in perfect.

8.3.1 Different type of welding

In this section, different types of welding are shown below:

- Butt welding
 - Square butt
 - Single V butt
 - Single Bevel butt
 - Single V butt with root
 - Single Bevel butt with root
 - Single U butt
 - Single J butt
- Backing run
- Fillet
- Plug or Slot
- Spot/spot(centered)
- Seam/Seam(centered)
- JIS spot
- JIS seam

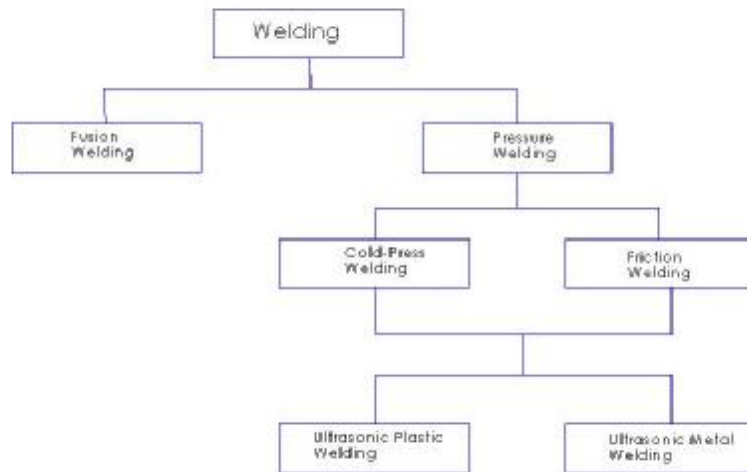


Figure 8.2: Welding chart



I have found some welding method, one of them is the fusion welding, it used by applying heat at the point of connection that fuse the pieces together. After the hardening of the mixed components, a solid joint occurs. Another type of welding is known as the pressure welding, it depends on the application of high pressure or high temperature. This result in a strong plastification and local deformation of the pieces to be joined in the welding area.

CHAPTER 9

CONCLUSION

After going through the whole report till now, I should come to a conclusion for the whole project. As there are few things I want include, firstly is the workability of this project. As you can see that I am not able to run it, the problem lies on the vacuum pad I am using for this project. It is not a good material as we have to consider the friction co efficiency, the surface area of contact as well as the power of the computer fan. As from this experience, the computer fan is not a good choice of creating the suction power. As the CFM of the computer is not strong enough to hold the whole robot, and it will take a long time for the pressure to build up. Thus this is the main reason the robot is about to climb up the wall. But if the material of the vacuum pad can be change, or the surface area of contact to be increase, the robot might be able to climb the wall. Although the robotis not able to climb the wall, I have learn a lot of things from it.

This has been a good experience for me as I have learnt what the things to consider when doing a project are. Things such as planning, research and resource as well as the experience of using a new programming language. I believe I am able to do things better in the future for any future project. Below are the things that I am not able to complete or improvements can be make to this project for the students who will be taking over this project.

They are listed here:

- Improvement can be make to the software programming.
- Better workmanship when fabricating the individual parts.
- Research on better computer fans or other better equipment to replace the idea here.
- Do research on Welding as it is needed for this project.

REFERENCE

Most of my researches come from the internet but not all of them are from the internet. One of them happened to be one of the programming I will be using for this project.

There are shown below:

- PIC your personal Introductory Course Second Edition By John Morton

Here are also the website I visited for more information about this type of software.

- <http://www.mikroe.com/en/>
- www.samengstrom.com/nxl/1669/pic_programming_page.en.html
- www.mstracey.btinternet.co.uk/pictutorial/picmain.htm

As for how other types of wall climbing robot move and the mechanism to control it are all in the website stated here :

- www.eoat.com/articles/article9.htm
- <http://nanolab.me.cmu.edu/publications/papers/Menon-ROBIO2004.pdf>
- <http://sun-valley.stanford.edu/papers/BretlMRL;2003pdf>
- <http://ai.stanford.edu/~lalombe/papers/iser04/iser04.pdf>
- www.fzi.de/KCMS/kcms_file.php?action=link&id=259
- http://www.netmotion.com/htm_files/mc_sensors_pe.htm
- [http://oeiwcsnts1.omron.com/pdfcatalog.nsf/5F47969049C9D8B286256C590048079A/\\$FILE/C21AppGuide0902.pdf](http://oeiwcsnts1.omron.com/pdfcatalog.nsf/5F47969049C9D8B286256C590048079A/$FILE/C21AppGuide0902.pdf)

Here are all the website about the computer fan datasheets come from

- www.directron.com/thinfan1.html
- www.directron.com/casefans.html
- www.directron.com/cases---case-fans.html
- <http://sg.hardwarezone.com/news/view.php?id=4516>
- <http://sg.hardwarezone.com/news/view.php?id=6841&cid=3>
- <http://www.hardwarezone.com.sg/news/view.php?id=5630>

Here is the website where all the air flow, size of the pipes are found

- <http://www.pipeflowcalculations.com/flowrate/theory.htm>

APPENDIX A

University of Southern Queensland
Faculty of Engineering and Surveying

ENG 4111/4112 Research Project PROJECT SPECIFICATION

TOPIC: Wall Climbing Robot
SUPERVISOR: Prof. John Billingsley

PROJECT AIM: The project seeks to investigate the factors to consider in building a wall climbing robot. Such as the weight of the object, pressure required for the object being hang on the vertical direction and gravity pulling it down.

PROGRAMME: Issue A, 14 MAR 2007

1. Research on background of different type of wall climbing robots and how they work.
2. Research on different type of motors that be used on my project (wall climbing robot) and I will choose a suitable motor for it.
3. Research on propeller/ fan for creating of the pressure used for my project. Will do research on pressure device and do experiment for my robot to climb the wall.
4. Research on material hanging on a vertical surface, power required to overcome the gravity force as well as the weight of the robot.
5. Research on the vacuum cup and other likewise device that will be used as the climbing device for my robot.
6. Research on different type of material used for the vacuum cups and which kind of the material used will be the best for this project.
7. Research on different types of software that will be used to control my robot and will it be fully automatic or is it controlled by a remote controller.
8. After that I will build the model of my robot and will test run on it.

AGREED:

VinZ Kang Boon Geh (Student) John Billingsley(Supervisor)
 (Dated) 13/4/2007

Pressure Units

	<u>Pascal</u> (Pa)	<u>Bar</u> (bar)	<u>Technical atmosphere</u> (at)	<u>Atmosphere</u> (atm)	<u>Torr</u> (mmHg)	<u>Pound-force per square inch</u> (psi)
1 Pa	$\equiv 1 \text{ N/m}^2$	10^{-5}	10.197×10^{-6}	9.8692×10^{-6}	7.5006×10^{-3}	145.04×10^{-6}
1 bar	100 000	$\equiv 10^6 \text{ dyn/cm}^2$	1.0197	0.98692	750.06	14.504
1 at	98 066.5	0.980665	$\equiv 1 \text{ kgf/cm}^2$	0.96784	735.56	14.223
1 atm	101 325	1.01325	1.0332	$\equiv 1 \text{ atm}$	760	14.696
1 torr	133.322	1.3332×10^{-3}	1.3595×10^{-3}	1.3158×10^{-3}	$\equiv 1 \text{ mmHg}$	19.337×10^{-3}
1 psi	6 894.76	68.948×10^{-3}	70.307×10^{-3}	68.046×10^{-3}	51.715	$\equiv 1 \text{ lbf/in}^2$

APPENDIX B

Bill Of Materials

ITEM	QTY	COST
1/2" round tubing	1	\$11.00
Computer fan	3	\$18.00
Stepper motor	1	\$24.00
USB serial converter	1	\$28.00
PIC microprocessor	1	\$80.00
300 X 300 X 2 Aluimum	1	\$30.00
100 ohm resistor	1	\$0.50
1K resistor	2	\$1.00
10K resistor	4	\$3.50
4K7 resistor	6	\$6.00
8K2 resistor	4	\$2.00
0.1uF(capacitor)	2	\$1.00
100uF(capacitor)	2	\$2.00
IRFZ44	4	\$6.00
IC4013	1	\$2.50
IC4030	2	\$5.00
IC4093	1	\$2.50
IC7805	1	\$1.50
BC547	6	\$15.00
2N2907	2	\$5.00
Plastic container	2	\$3.50
Double 'A' battery	8	\$12.00
battery holder	1	\$20.00
3" heat sink cable	3	\$15.00
2" heat sink cable	3	\$15.00
cable(green)	1	\$15.00
cable(yellow)	1	\$15.00
cable(red)	1	\$15.00
cable(black)	1	\$15.00
cable(white)	1	\$15.00
M3 screw	10	\$20.00
M4 screw	20	\$40.00
EE-SX671	4	\$100.00
PM2-LH10	2	\$50.00
Total Cost		\$595.00

Source code:

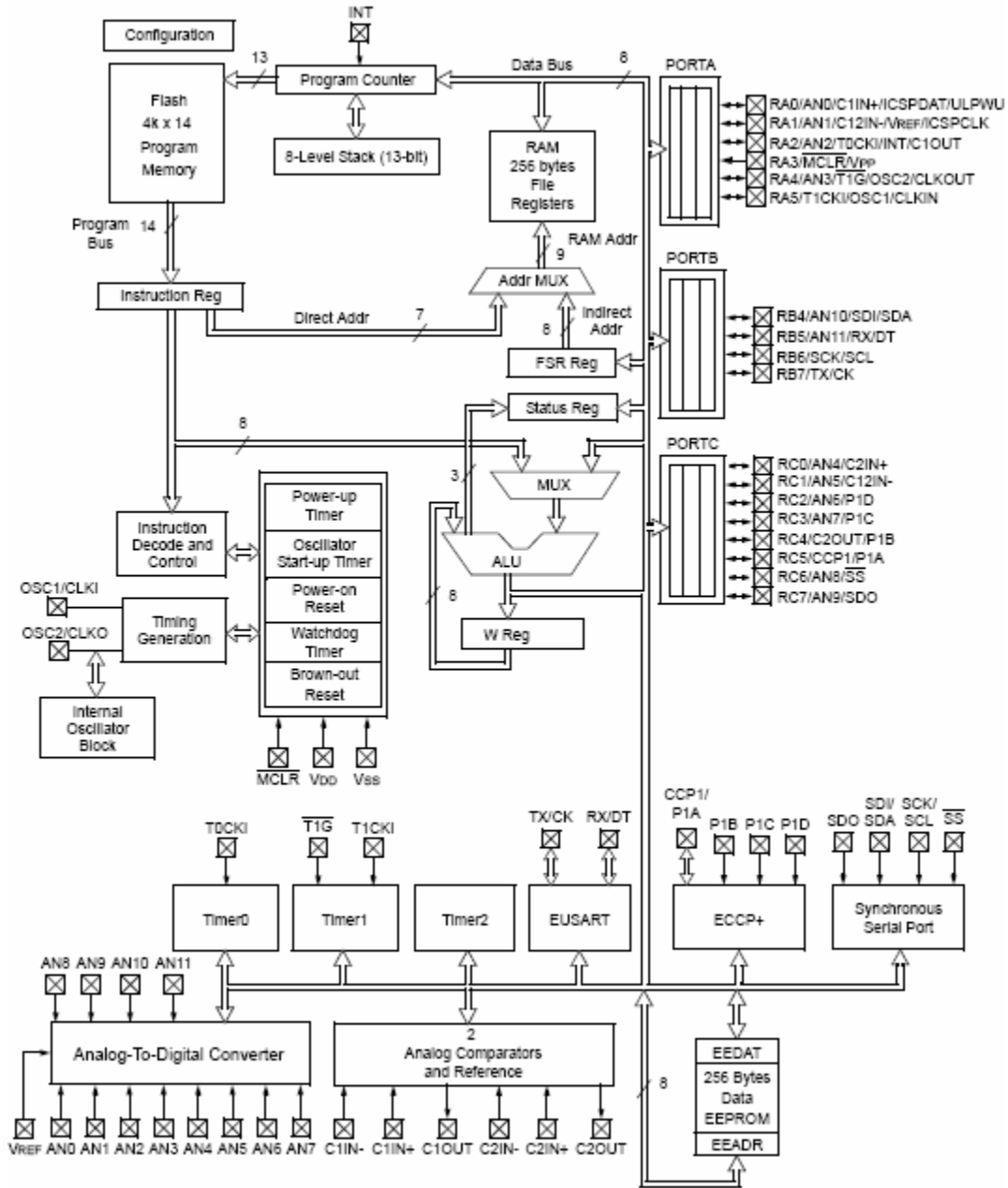
<u>Code</u>		<u>Comment</u>
addlw	number	adds a no. with the no. in the working register
addwf	FileReg,f	adds the no. in the working register to the no. in a file register and puts the result in the file register
addwf	File Reg,w	adds the no. in the working register to the no. in a file register and puts the result back into the working register leaving the file register unchanged.
Andlw	number	ANDs a no. with the no. in the working register, leaving the result in the working register
andwf	FileReg,f	ANDs the no. in the working register with the no.in a file register and puts the result in the file register
bcf	FileReg, bit	clears a bit in a file register
bsf	FileReg,bit	sets a bit in a file register
btfs	FileReg,bit	test a bit in a file register and skips the next instruction if the result is clear
btfss		test a bit in a file register & skips the next instruction if the result is set
call	Any Sub	makes the chip call a subroutine, after which it will return to where it left off
clrf	FileReg	clears the no. in a file register
clrw		clears the no. in the working register
clrwdt		clears the no. in the watchdog timer
comf		complements the no. in a file register, leaving the result in the file register
decf	FileReg,f	decrements a file register & puts the result in the file register
decfsz	FileReg,f	decrements a file register & if the result is zero it skips the next instruction. The result is put in the file register

goto	Anywhere	makes the chip go to somewhere in the program which you labeled
incf	FileReg,f	increments a file register & puts the result in the file register
incfsz	FileReg,f	increments a file register and if the result is zero it skips the next instruction. The result is put in the file register
iorlw	number	inclusive ORs a no. with the no. in the working register
iorwf		inclusive Ors the no. in the working register with the no. in a file register & puts the result in the file register
movfw	FileReg	moves the no in a file register into the working register
movlw	number	moves a no. into the working register
movwf nop	FileReg	moves the no. in the working register into a file register stands for no operation
option		takes the no. in the working register and moves it into the option register
retfie		returns from a subroutine & enables the Global Interrupt Enable bit
retlw	number	return from asubroutine with a particular no. in the working register
return		return from a subroutine
rlf	FileReg,f	rotates the bits in a file register to the left, putting the result in the file register
rrf	FileReg,f	rotates the bits in a file register to the right, putting the result in the file register
sleep		send the PIC to sleep, lower power consumption mode
sublw	number	subtracts the no. in the working register from a no.
subwf	FileReg,f	subtracts the no. in the working register from the no. in a file register & puts the result in the file register

swapf	FileReg,f	swaps the two halves of the 8 bit binary no. in a file register, having the result in the file register
tris	PORTX	uses the no. in the working register to specify which bits of a port are inputs and which are outputs
xorlw	number	exclusive Ors a no. with the no. in the working register
xorwf	FileReg,f	exclusive Ors the no. in the working register with the no. in a file register & puts the result in the file register

APPENDIX C

PIC16F690 data sheet:



Name	Function	Input Type	Output Type	Description
RA0/AN0/C1IN+/ICSPDAT/ ULPWU	RA0	TTL	—	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN0	AN	—	A/D Channel 0 input.
	C1IN+	AN	—	Comparator 1 positive input.
	ICSPDAT	TTL	CMOS	ICSP Data I/O.
	ULPWU	AN	—	Ultra Low-Power Wake-up input.
RA1/AN1/C12IN-/VREF/ICSPCLK	RA1	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN1	AN	—	A/D Channel 1 input.
	C12IN-	AN	—	Comparator 1 or 2 negative input.
	VREF	AN	—	External Voltage Reference for A/D.
	ICSPCLK	ST	—	ICSP™ clock.
RA2/AN2/T0CKI/INT/C1OUT	RA2	ST	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN2	AN	—	A/D Channel 2 input.
	T0CKI	ST	—	Timer0 clock input.
	INT	ST	—	External interrupt.
	C1OUT	—	CMOS	Comparator 1 output.
RA3/MCLR/VPP	RA3	TTL	—	General purpose input. Individually controlled interrupt-on-change.
	MCLR	ST	—	Master Clear with internal pull-up.
	VPP	HV	—	Programming voltage.
RA4/AN3/T1G/OSC2/CLKOUT	RA4	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN3	AN	—	A/D Channel 3 input.
	T1G	ST	—	Timer1 gate input.
	OSC2	—	XTAL	Crystal/Resonator.
	CLKOUT	—	CMOS	Fosc/4 output.
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	T1CKI	ST	—	Timer1 clock input.
	OSC1	XTAL	—	Crystal/Resonator.
	CLKIN	ST	—	External clock input/RC oscillator connection.
RB4/AN10/SDI/SDA	RB4	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN10	AN	—	A/D Channel 10 input.
	SDI	ST	—	SPI data input.
	SDA	ST	OD	I ² C data input/output.
RB5/AN11/RX/DT	RB5	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN11	AN	—	A/D Channel 11 input.
	RX	ST	—	EUSART asynchronous input.
	DT	ST	CMOS	EUSART synchronous data.

Name	Function	Input Type	Output Type	Description
RB6/SCK/SCL	RB6	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	SCK	ST	CMOS	SPI™ clock.
	SCL	ST	OD	I ² C™ clock.
RB7/TX/CK	RB7	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	TX	—	CMOS	EUSART asynchronous output.
	CK	ST	CMOS	EUSART synchronous clock.
RC0/AN4/C2IN+	RC0	ST	CMOS	General purpose I/O.
	AN4	AN	—	A/D Channel 4 input.
	C2IN+	AN	—	Comparator 2 positive input.
RC1/AN5/C12IN-	RC1	ST	CMOS	General purpose I/O.
	AN5	AN	—	A/D Channel 5 input.
	C12IN-	AN	—	Comparator 1 or 2 negative input.
RC2/AN6/P1D	RC2	ST	CMOS	General purpose I/O.
	AN6	AN	—	A/D Channel 6 input.
	P1D	—	CMOS	PWM output.
RC3/AN7/P1C	RC3	ST	CMOS	General purpose I/O.
	AN7	AN	—	A/D Channel 7 input.
	P1C	—	CMOS	PWM output.
RC4/C2OUT/P1B	RC4	ST	CMOS	General purpose I/O.
	C2OUT	—	CMOS	Comparator 2 output.
	P1B	—	CMOS	PWM output.
RC5/CCP1/P1A	RC5	ST	CMOS	General purpose I/O.
	CCP1	ST	CMOS	Capture/Compare input.
	P1A	ST	CMOS	PWM output.
RC6/AN8/SS	RC6	ST	CMOS	General purpose I/O.
	AN8	AN	—	A/D Channel 8 input.
	SS	ST	—	Slave Select input.
RC7/AN9/SDO	RC7	ST	CMOS	General purpose I/O.
	AN9	AN	—	A/D Channel 9 input.
	SDO	—	CMOS	SPI data output.
V _{ss}	V _{ss}	Power	—	Ground reference.
V _{DD}	V _{DD}	Power	—	Positive supply.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

File Address	File Address	File Address	File Address
Indirect addr. ⁽¹⁾ 00h	Indirect addr. ⁽¹⁾ 80h	Indirect addr. ⁽¹⁾ 100h	Indirect addr. ⁽¹⁾ 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h	PORTA 105h	TRISA 185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h	PORTC 107h	TRISC 187h
08h	88h	108h	188h
09h	89h	109h	189h
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDAT 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 ⁽¹⁾ 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	18Eh
TMR1H 0Fh	OSCCON 8Fh	EEADRH 10Fh	18Fh
T1CON 10h	OSCTUNE 90h	110h	190h
TMR2 11h	91h	111h	191h
T2CON 12h	PR2 92h	112h	192h
SSPBUF 13h	SSPADD ⁽²⁾ 93h	113h	193h
SSPCON 14h	SSPSTAT 94h	114h	194h
CCPR1L 15h	WPUA 95h	WPUB 115h	195h
CCPR1H 16h	IOCA 96h	IOCB 116h	196h
CCP1CON 17h	WDTCON 97h	117h	197h
RCSTA 18h	TXSTA 98h	VRCON 118h	198h
TXREG 19h	SPBRG 99h	CM1CON0 119h	199h
RCREG 1Ah	SPBRGH 9Ah	CM2CON0 11Ah	19Ah
1Bh	BAUDCTL 9Bh	CM2CON1 11Bh	19Bh
PWM1CON 1Ch	9Ch	11Ch	19Ch
ECCPAS 1Dh	9Dh	11Dh	PSTRCON 19Dh
ADRESH 1Eh	ADRESL 9Eh	ANSEL 11Eh	SRCON 19Eh
ADCON0 1Fh	ADCON1 9Fh	ANSELH 11Fh	19Fh
20h	A0h	120h	1A0h
General Purpose Register	General Purpose Register	General Purpose Register	
96 Bytes	80 Bytes	80 Bytes	
7Fh	EFh	16Fh	
	accesses 70h-7Fh	accesses 70h-7Fh	accesses 70h-7Fh
Bank 0	Bank 1	Bank 2	Bank 3

Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.
Note 2: Address 93h also accesses the SSP Mask (SSPMSK) register under certain conditions. See Registers 13-2 and 13-3 for more details.

TABLE 2-1: PIC16F685/687/689/690 SPECIAL REGISTERS SUMMARY BANK 0

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Value on all other Resets ⁽¹⁾
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxxx xxxxx	xxxxx xxxxx
01h	TMR0	Timer0 Module Register								xxxxx xxxxx	uuuuu uuuuu
02h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
03h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxxx	000q quuuu
04h	FSR	Indirect Data Memory Address Pointer								xxxxx xxxxx	uuuuu uuuuu
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	-. .xx xxxxx	-. .uu uuuuu
06h	PORTB	RB7	RB6	RB5	RB4	—	—	—	—	xxxxx ----	uuuuu ----
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxxx xxxxx	uuuuu uuuuu
08h	—	Unimplemented								—	—
09h	—	Unimplemented								—	—
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of Program Counter					---0 0000	---0 0000
0Bh	INTCON	GIE	PEIE	TOIE	INTE	RABIE	TOIF	INTF	RABIF ⁽²⁾	0000 000x	0000 000x
0Ch	PIR1	—	ADIF	RCIF ⁽³⁾	TXIF ⁽³⁾	SSPIF ⁽³⁾	CCP1IF ⁽⁴⁾	TMR2IF ⁽⁴⁾	TMR1IF	-000 0000	-000 0000
0Dh	PIR2	OSFIF	C2IF	C1IF	EEIF	—	—	—	—	0000 ----	0000 ----
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1								xxxxx xxxxx	uuuuu uuuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1								xxxxx xxxxx	uuuuu uuuuu
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	0000 0000	uuuuu uuuuu
11h	TMR2	Timer2 Module Register								0000 0000	0000 0000
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
13h	SSPBUF ⁽³⁾	Synchronous Serial Port Receive Buffer/Transmit Register								xxxxx xxxxx	uuuuu uuuuu
14h	SSPCON ^(3, 5)	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
15h	CCPR1L ⁽⁴⁾	Capture/Compare/PWM Register 1 (LSB)								xxxxx xxxxx	uuuuu uuuuu
16h	CCPR1H ⁽⁴⁾	Capture/Compare/PWM Register 1 (MSB)								xxxxx xxxxx	uuuuu uuuuu
17h	CCP1CON ⁽⁴⁾	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
18h	RCSTA ⁽³⁾	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG ⁽³⁾	EUSART Transmit Data Register								0000 0000	0000 0000
1Ah	RCREG ⁽³⁾	EUSART Receive Data Register								0000 0000	0000 0000
1Bh	—	Unimplemented								—	—
1Ch	PWM1CON ⁽⁴⁾	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	0000 0000
1Dh	ECCPAS ⁽⁴⁾	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	0000 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxxx xxxxx	uuuuu uuuuu
1Fh	ADCON0	ADFM	VCFG	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	0000 0000	0000 0000

Legend: — = Unimplemented locations read as '0'. u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented
Note:
 1: Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.
 2: MCLR and WDT Reset do not affect the previous value data latch. The RABIF bit will be cleared upon Reset but will set again if the mismatched exists.
 3: PIC16F687/PIC16F689/PIC16F690 only.
 4: PIC16F685/PIC16F690 only.
 5: When SSPCON bits SSPM<3:0> = 1001, any reads or writes to the SSPADD SFR address are accessed through the SSPMSK register. See Registers 13-2 and 13-3 for more detail.

TABLE 2-2: PIC16F685/687/689/690 SPECIAL FUNCTION REGISTERS SUMMARY BANK 1

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Value on all other Resets ⁽¹⁾
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
81h	OPTION_REG	RABPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
83h	STATUS	IRP	RP1	RP0	T0	PD	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
88h	—	Unimplemented								—	—
89h	—	Unimplemented								—	—
8Ah	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	---0 0000
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF ⁽²⁾	0000 000x	0000 000x
8Ch	PIE1	—	ADIE	RCIE ⁽³⁾	TXIE ⁽³⁾	SSPIE ⁽³⁾	CCP1IE ⁽⁴⁾	TMR2IE ⁽⁴⁾	TMR1IE	-000 0000	-000 0000
8Dh	PIE2	OSFIE	C2IE	C1IE	EEIE	—	—	—	—	0000 ----	0000 ----
8Eh	PCON	—	—	ULPWUE	SBOREN	—	—	POR	BOR	--01 --qq	--0u --uu
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	-110 q000	-110 x000
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	---u uuuu
91h	—	Unimplemented								—	—
92h	PR2 ⁽⁴⁾	Timer2 Period Register								1111 1111	1111 1111
93h	SSPADD ^(3, 6)	Synchronous Serial Port (I ² C mode) Address Register								0000 0000	0000 0000
93h	SSPMSK ^(3, 6)	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111	1111 1111
94h	SSPSTAT ⁽³⁾	SMP	CKE	D \bar{A}	P	S	R \bar{W}	UA	BF	0000 0000	0000 0000
95h	WPUA ⁽⁵⁾	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	--11 -111
96h	IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	--00 0000
97h	WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	---0 1000
98h	TXSTA ⁽³⁾	CSRC	TX9	TXEN	SYNC	SEN5	BRGH	TRMT	TX9D	0000 0010	0000 0010
99h	SPBRG ⁽³⁾	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
9Ah	SPBRGH ⁽³⁾	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
9Bh	BAUDCTL ⁽³⁾	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
9Fh	ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	-000 ---

Legend: -- Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

- Note**
- Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.
 - MCLR and WDT Reset do not affect the previous value data latch. The RABIF bit will be cleared upon Reset but will set again if the mismatched exists.
 - PIC16F687/PIC16F689/PIC16F690 only.
 - PIC16F685/PIC16F690 only.
 - RA3 pull-up is enabled when pin is configured as MCLR in Configuration Word.
 - Accessible only when SSPM<3:0> = 1001.

TABLE 2-3: PIC16F685/687/689/690 SPECIAL REGISTERS SUMMARY BANK 2

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Value on all other Resets ⁽¹⁾
Bank 2											
100h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
101h	TMR0	Timer0 Module Register								xxxx xxxx	uuuu uuuu
102h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000
103h	STATUS	IRP	RP1	RP0	T \bar{O}	P \bar{D}	Z	DC	C	0001 1xxx	000q quuu
104h	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	uuuu uuuu
105h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--uu uuuu
106h	PORTB	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	uuuu ----
107h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter				---0 0000	---0 0000	
10Bh	INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF ⁽²⁾	0000 000x	0000 000x
10Ch	EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	0000 0000
10Dh	EEADR	EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	0000 0000
10Eh	EEDATH ⁽³⁾	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	--00 0000
10Fh	EEADRH ⁽³⁾	—	—	—	—	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---- 0000	---- 0000
110h	—	Unimplemented								—	—
111h	—	Unimplemented								—	—
112h	—	Unimplemented								—	—
113h	—	Unimplemented								—	—
114h	—	Unimplemented								—	—
115h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	1111 ----
116h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	0000 ----	0000 ----
117h	—	Unimplemented								—	—
118h	VRCON	C1VREN	C2VREN	VRR	VP6EN	VR3	VR2	VR1	VR0	0000 0000	0000 0000
119h	CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	0000 -000
11Ah	CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	0000 -000
11Bh	CM2CON1	MC1OUT	MC2OUT	—	—	—	—	T1GSS	C2SYNC	00-- --10	00-- --10
11Ch	—	Unimplemented								—	—
11Dh	—	Unimplemented								—	—
11Eh	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
11Fh	ANSELH	—	—	—	—	ANS11	ANS10	ANS9	ANS8	---- 1111	---- 1111

Legend: -- = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented
Note 1: Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.
 2: MCLR and WDT Reset does not affect the previous value data latch. The RABIF bit will be cleared upon Reset but will set again if the mismatched exists.
 3: PIC16F685/PIC16F689/PIC16F690 only.

TABLE 2-4: PIC16F685/687/689/690 SPECIAL FUNCTION REGISTERS SUMMARY BANK 3

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR/BOR Reset	Value on all other Resets ⁽¹⁾	
Bank 3												
180h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx	
181h	OPTION_REG	RABFU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111	
182h	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	0000 0000	
183h	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxxx	000g quuuu	
184h	FSR	Indirect Data Memory Address Pointer								xxxxx xxxxx	uuuu uuuuu	
185h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111	
186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----	
187h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111	
188h	—	Unimplemented								—	—	
189h	—	Unimplemented								—	—	
18Ah	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter				---	0 0000	---	0 0000
18Bh	INTCON	GIE	PEIE	TOIE	INTE	RABIE	TOIF	INTF	RABIF ⁽²⁾	0000 000x	0000 000x	
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	0--- q000	
18Dh	EECON2	EEPROM Control Register 2 (not a physical register)								-----	-----	
18Eh	—	Unimplemented								—	—	
18Fh	—	Unimplemented								—	—	
190h	—	Unimplemented								—	—	
191h	—	Unimplemented								—	—	
192h	—	Unimplemented								—	—	
193h	—	Unimplemented								—	—	
194h	—	Unimplemented								—	—	
195h	—	Unimplemented								—	—	
196h	—	Unimplemented								—	—	
197h	—	Unimplemented								—	—	
198h	—	Unimplemented								—	—	
199h	—	Unimplemented								—	—	
19Ah	—	Unimplemented								—	—	
19Bh	—	Unimplemented								—	—	
19Ch	—	Unimplemented								—	—	
19Dh	PSTRCON ⁽³⁾	—	—	—	STRSYNC	STRD	STRC	STRB	STRA	---0 0001	---0 0001	
19Eh	SRCON	SR1	SR0	C1SEN	C2REN	PULSS	PULSR	—	—	0000 00--	0000 00--	
19Fh	—	Unimplemented								—	—	

Legend: -- = Unimplemented locations read as '0', u = unchanged, x = unknown, g = value depends on condition, shaded = unimplemented
Note 1: Other (non Power-up) Resets include MCLR Reset and Watchdog Timer Reset during normal operation.
 2: MCLR and WDT Reset does not affect the previous value data latch. The RABIF bit will be cleared upon Reset but will set again if the mismatched exists.
 3: PIC16F685/PIC16F690 only.

Computer fan datasheet:

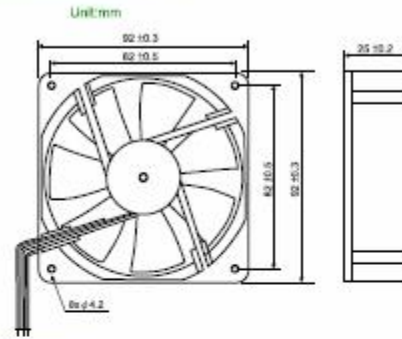
D9225 Series

Size: 92x92x25mm (3.62"x3.62"x0.98")

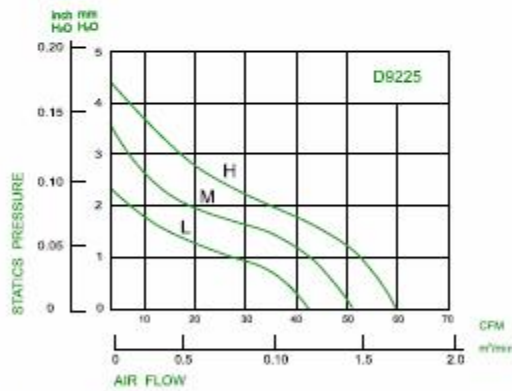


DIMENSION

HOUSING : Thermoplastic PBT, UL94V-0
 IMPELLER : Thermoplastic PBT, UL94V-0 BEARING : Sleeve or Ball
 OPERATING TEMPERATURE : Sleeve Bearing- 10°C to + 70°C
 Ball Bearing- 30°C to + 75°C
 INSULATION RESISTANCE : 10 meg Ohms Min. @ 500 VDC
 (Between Frame and Terminal)
 DIELECTRIC STRENGTH : 5 mA Max. at 500 VAC 60 Hz
 for 1 minute
 (Between Frame and Terminal)
 WEIGHT : 120g (4.23 oz.)



AIR PERFORMANCE



SPECIFICATION

MODEL NO.	RATED VOLTAGE (V)	OPERATING VOLTAGE (V)	RATED CURRENT (A)	INPUT POWER (W)	SPEED (RPM)	MAX AIR PRESSURE		AIR FLOW		NOISE dB(A) @ 1M
						mmH ₂ O	InchH ₂ O	CFM	m ³ /min	
D9225V12H	12	10.2 - 13.8	0.32	3.84	3000	4.30	0.17	58.70	1.66	38
D9225V12M	12	10.2 - 13.8	0.20	2.40	2500	3.50	0.14	50.50	1.43	34
D9225V12L	12	10.2 - 13.8	0.16	1.92	2000	2.30	0.09	41.50	1.17	28
D9225V24H	24	20.4 - 27.6	0.18	4.32	3000	4.30	0.17	58.70	1.66	38
D9225V24M	24	20.4 - 27.6	0.14	3.36	2500	3.50	0.14	50.50	1.43	34
D9225V24L	24	20.4 - 27.6	0.10	2.40	2000	2.30	0.09	41.50	1.17	28

Fan specification:

Fan dimension: 60 X 60 X 25
 Rated Voltage: 12 V
 Started Voltage: 7 V
 Rated current: 0.17A ±10%

Power input: 1.8 W \pm 10%
 Fan speed: 4550r.p.m \pm 10%
 Static Pressure: Max 5.30 mmH₂O
 Max. Air flow: 21 CFM
 Noise: 28dB
 Life Time: 50,000 hrs
 Operating Temp: -10°C - 70°C
 Net Weight: 60 G

Vacuum data sheet:

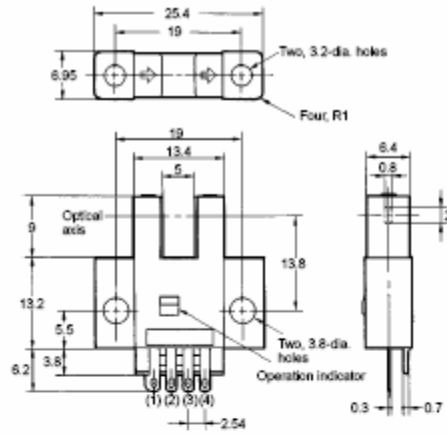
ANVER Material Blend Code	Common Material Name	Shore A Hardness* (Durometer) + / - 5	Temperature Range ** °F (°C)	Abrasion Wear Resistance	Oil, Grease Resistance	UV Weather Aging Resistance	Typical Color (Depends on Cup Style)
For most General Purpose Industrial Applications							
NBR	Nitrile (Buna-N)	40 - 60	-40° to +230° F (-40° to +110° C)	●●●●	●●●●	●●●	Black, Blue
	NBR is an excellent all around material for general industry. High overall value due to its top performance in many categories. Our proprietary blend which features the latest name brand ingredients have greatly increased the performance of this proven material.						
CR, NEO, NE, N50	Neoprene (Chloroprene)	40 - 60	-40° to +230° F (-40° to +110° C)	●●●●	●●●●	●●●	Blue, Red or Black
	CR, NEO, NE, N50 is an excellent hard wearing all around material for general industry with a nice rubber feel and memory.						
PUR	Polyurethane Anverflex™	30 - 65	-13° to +176° F (-25° to +80° C)	●●●●	●●●●	●●●	Blue, Green, White all colors
	Anverflex PUR is a new proprietary blend which is tough, and long wearing. It is non-marking and is long lasting as pressure and heat is used to cure - vulcanize the material in the molding process. Suitable for many applications where heat is not a factor. Shiny gloss finish.						
For High Heat, Non-Marring Packaging and Food Use							
Mat. Code	Common Name	Hardness	Temp.	Wear	Oil, Grease	UV / Weather	Typical Color
SIT	Silicone	40 - 60	-94° to +600° F (-70° to +316° C)	●●●	●●●●	●●●●	Translucent Clear
	SIT is excellent for high heat or food packaging. Soft and pliable, meets FDA Title 21 and German spec. BGVV (BGA) Part XV for contaminant-free load handling. Contains no dyes that can leach out when handling baked goods, drugs, glassware, hot products from molds,						

	etc. Our new formulation minimizes almost completely the yellowing with age which use to occur with clear silicone in the past.						
SI	Silicone	40 - 60	-94° to +392° F (-70° to +200° C)	••	••	••••	Solid White, Orange, Red
	SI is excellent for high heat applications such as mold part removal or where heat resistance is required for large cups/seals. Silicone is also a good choice for cold outside applications as it stays fairly soft to lower temperatures than Nitrile or Neoprene.						
S45, S60	Silicone	40 - 60	-58° to +401° F (-50° to +205° C)	••	••	••••	Orange, Red
	S45 is excellent for EOAT high heat applications such as mold part removal from plastic injection machines requiring a soft touch. Higher durometer for bellows cups.						
MD (SI)	Metal Detectable Silicone	40 - 60	-94° to +600° F (-70° to +316° C)	••	••	••••	Brown, Tan Brownish Red
	MD is a unique Silicone formulation made specifically for high heat applications in the baking food industry. This special rubber blend is designed for use with metal detection equipment used to safeguard food from metal objects as it is conveyed down the line.						
For the Printing, Paper and Wood Industries							
NR	Natural (Gum) Rubber	35 - 50	-40° to +176° F (-40° to +80° C)	••••	•	•	Tan, Grey, Green, Orange or Black
	NR is widely used and specialized for the printing, paper, and wood industries. It is low cost, non-marking, wears well and has a natural negative charge which allows it not gum up with ink or cut paper dust. While great for picking paper sheets up this is basically the same material used in rubber bands and is not suitable for general use. This material does not respond well to UV light, ozone, chemicals, oils, water etc.						
Specialty Elastomers							
Mat. Code	Common Name	Hardness	Temp.	Wear	Oil, Grease	UV / Weather	Typical Color
ESD	Electro-Static Dissipative Nomastat™	50 - 60	-76° to +401° F (-60° to +205° C)	••	••	••	Black
	Electro-Static Dissipative Nomastat™ ESD is a specialized proprietary rubber that can handle heat and is static dissipative for any industry where static buildup is a problem or explosion proof requirements exist. Applications are typically in the electronics, high tech, automation, packaging and printing industries. It actually bleeds any static build-up out through a ground of the machine or other ground designed into the system. Neutralized additives are blended in a special elastomer blend to diffuse static buildup. This elastomer blend has a insulating static-dissipating value of 97K Ω *CM @ 500 Volts.						
TPU	Thermal Polyurethane	75	-13° to +176° F (-25° to +80° C)	••••	••••	••••	Translucent Brown, Green, Black (Darkens w/ Age)

	<p>TPU is an extremely tough material which darkens with UV exposure. It is sold under various proprietary trade names and the main market is hard caster wheels. We use TPU in static press applications such as bottle gripper bladders but rarely on cups anymore. We found it long wearing in static applications but usually too expensive to justify its use in vacuum cups which flex a great deal. To compensate for the material's hardness, cups using this material must be designed with thinner walls which break down faster. While initially promising a few years ago, TPU's expense, hardness, lack of memory have proven to far outweigh its increased wear performance over other recently improved materials.</p>						
FPM	<p>Viton® Fluorocarbon</p>	60 - 65	-4° to +482° F (-20° to +250° C)				Usually Black, Blue
	<p>FPM is specialized for high heat jobs. It is best for static compression applications and has a stiff, somewhat dead feel for vacuum cups. It is also relatively expensive which limits its vacuum cup use. The newer High Heat Silicones have replaced Viton for most applications.</p>						
VYL	<p>Vinyl***</p>	30 - 70	+32° to +158° F (0° to +70° C)				Clear Base Blue or other colors
	<p>VYL is soft, low cost, and readily available in many grades for general light-duty or household use. Poured vinyl's or the clear vinyl used for push-on type suction cups are not suitable for industrial use. Vinyl will work for push on cups but they break down if repeatedly used for vacuum cups used in automation as they do not have good memory, tensile strength or mechanical properties.. Vinyl also handles heat poorly. The injection molded blended vinyl ANVER uses in its industrial vacuum cups is better, but you loose a key advantage of low cost to other materials.</p>						
NM	<p>ANVER Nomathane™</p>	50 - 70	-32° to +356° F (-0° to +180° C)				Blue, Purple others
	<p>NM is a new ANVER proprietary material which is non marking yet high wearing even in hot conditions. It is a silicone-free hybrid blend which is non marking and will not leave any residue, mold release agent or ghost-mark on products which need to be painted after handling. This material also handles high temperature and is extremely long wearing making it ideal for the plastic injection mold tooling (EOAT) industry. While not inexpensive this material is priced similar to other top quality urethanes but it offers higher overall value due to its excellent performance. In many applications where heat and non-marking are issues, Nomathane is the best blend available on the market.</p>						
GR	<p>ANVER Andurasoft™</p>	35	-40° to +176° F (-40° to +80° C)				Light Green
	<p>GR is a new ANVER proprietary material which is long wearing for yet extremely soft. It is a Rubber hybrid blend which is designed to be able to handle moderately rough materials such as sawn stone, textured plastic, fiberglass mat and saw cut wood surfaces. The rubber is so soft it can replace foam seals in many applications. It is the softest molded material available for large suction cups and is more economical than foam seals due to its durability.</p>						

EE-SX671 data sheet:

EE-SX670/670P
 EE-SX670A/670R
 EE-SX470/470P

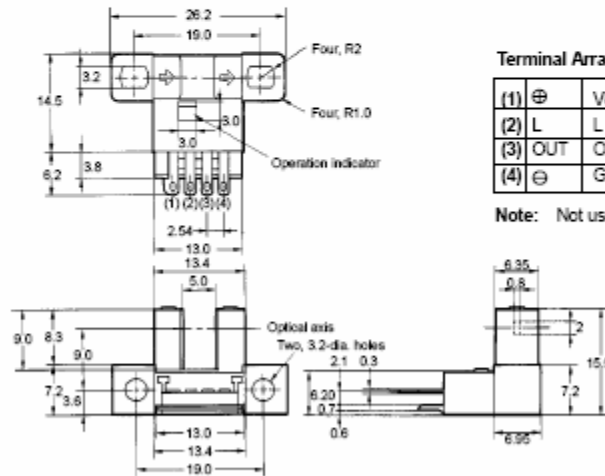


Terminal Arrangement

(1) ⊕	Vcc
(2) L	L (see note)
(3) OUT	OUT PUT
(4) ⊖	GND (0 V)

Note: Not used in EE-SX471

EE-SX671/671P
 EE-SX671A/671R
 EE-SX471/471P



Terminal Arrangement

(1) ⊕	Vcc
(2) L	L (see note)
(3) OUT	OUT PUT
(4) ⊖	GND (0 V)

Note: Not used in EE-SX471

PM2-LH10 datasheet

All models



This product is not a safety sensor. Its use is not intended or designed to protect life and prevent body injury or property damage from dangerous parts of machinery. It is a normal object detection sensor.

Mounting

- When fixing the sensor with screws, use M3 screws and the tightening torque should be 0.49 N·m or less. Further, use small, round type plain washers (ϕ 6 mm ϕ 0.236 in).

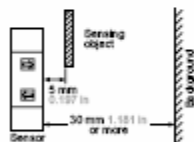


Wiring

- Make sure to connect terminals correctly as the sensor does not incorporate a reverse polarity protection circuit.
- If the sensor is being used in a noisy environment, examine the extent of noise. Further, if equipment, such as motor, solenoid or electromagnetic valve, which generates a large surge, is present near the sensor, connect a surge absorber to the equipment.

Setting

- The optimum setting distance (distance to convergent point) is 5 mm 0.197 in. The sensor is not affected even by a specular background if it is located 30 mm 1.181 in, or more, away from the sensor.



(However, the specular background should be a plane surface, directly facing the sensor. A spherical or curved background may be detected.)

Others

- Do not use during the initial transient time (50 ms) after the power supply is switched on.
- Take care that the product does not come in direct contact with oil, grease, or organic solvents, such as, thinner, etc.

Connector type

Cautions in plugging or unplugging a connector



- Do not plug or unplug a connector more than 10 times.
- Be sure not to give stress more than 5 N to a terminal of both a connector and a sensor. If you do not follow the above cautions, it will cause a poor contact.

Procedures of plugging or unplugging a connector

- Insert a connector straight into a sensor until the connector lug is locked by the sensor hook.



- When unplugging, give as much stress as a connector lug can be relieved from a hook. Then unplug it.



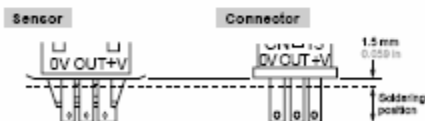
Caution: Be sure to hold a connector when plugging or unplugging it. Do not hold a terminal or a cable when plugging or unplugging the connector. Otherwise, it will cause a poor contact.



Soldering (Both connector CN-13 and sensor)

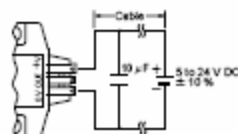
- If soldering is done directly on the terminals, strictly adhere to the conditions given below.

Soldering temperature	260 °C 500 °F or less
Soldering time	10 sec. or less
Soldering position	Refer to the below figure



Wiring

- The cable length must be 2 m 6.562 ft, or less, with 0.3 mm², or more, cable. If the cable is extended for more than 2 m 6.562 ft, connect a capacitor of 10 μ F approx. between +V and 0V terminals.



APPENDIX D

Below are the table for half stepping for controlling clockwise and anti-clockwise direction.

This table shows the control for clockwise direction:

STEPS	Q1	Q2	Q3	Q4
1	1	0	1	0
2	1	0	0	0
3	1	0	0	1
4	0	0	0	1
5	0	1	0	1
6	0	1	0	0
7	0	1	1	0
8	0	0	1	0

Table A: Half-wave (clockwise direction)

The table for this shows the control for anti-clockwise direction:

STEPS	Q1	Q2	Q3	Q4
1	1	0	1	0
2	0	0	1	0
3	0	1	1	0
4	0	1	0	0
5	0	1	0	1
6	0	0	0	1
7	1	0	0	1
8	1	0	0	0

Table B: Half-wave (anti clockwise direction)

The next section will show the PORT B, PORT C, TRISB and TRISC, please look at the details as well:

PORT B address is at 06h/106h

R/W-x R/W-x R/W-x R/W-x U-0 U-0 U-0 U-0

RB7	RB6	RB5	RB4	-	-	-	-
-----	-----	-----	-----	---	---	---	---

bit 7

bit 0

Bit 7-4 PORT I/O pin bit

1 = Port pin > V_{IH}

0 = Port pin < V_{IL}

Bit 3-0 Unimplemented: Read as '0'

PORT C address is at 07h/107h

R/W-x R/W-x R/W-x R/W-x R/W-x R/W-x R/W-x R/W-x

RC7	RC6	RA5	RA4	RA3	RA2	RA1	RA0
-----	-----	-----	-----	-----	-----	-----	-----

bit 7

bit 0

Bit 7-0 PORT I/O pin bit

1 = Port pin > V_{IH}

0 = Port pin < V_{IL}

TRISB address is at 86h/186h

R/W-1 R/W-1 R/W-1 R/W-1 U-0 U-0 U-0 U-0

TRISB7	TRISB6	TRISB5	TRISB4	-	-	-	-
--------	--------	--------	--------	---	---	---	---

bit 7

bit 0

Bit 7-4 PORT B Tri-State Control bit

1 = Port B pin configured as input (tri-stated)

0 = Port B pin configured as output (tri-stated)

Bit 3-0 Unimplemented : Read as '0'

TRISC address is at 87h/187h

R/W-1 R/W-1 R/W-1 R/W-1 R/W-1 R/W-1 R/W-1 R/W-1

TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
--------	--------	--------	--------	--------	--------	--------	--------

bit 7

bit 0

Bit 7-0 PORT C Tri-State Control bit

1 = Port C pin configured as input (tri-stated)

0 = Port C pin configured as output (tri-stated)