TELE-OPERATED ROBOT USING I^2C PROGRAMMING PROTOCOL

By

AHMAD FARHAN BIN HASBI

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

TELE-OPERATED ROBOT USING I^2C PROGRAMMING PROTOCOL

by

Ahmad Farhan bin Hasbi

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

Mrs. Noor Hazrin Hany M Haniff
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

December 2007

CERTIFICATION OF ORIGINALITY

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

Ahmad Farhan bin Hasbi

ABSTRACT

This project deals with a development of a tele-operated robot that utilizes I^2C bus as the main communication medium. The development stages are comprised of structure and mechanism construction, electronic circuit development and programming algorithms generation. The tele-operated robot functions as a transporter. It has the ability to transport objects from one location to another. This is done by the mechanism part that includes the gripper and the lifter part. The function of the gripper part is to grip or release desired objects. The lifter part on the other hand, lifts the gripped object to a desired height. This makes the tele-operated robot a versatile transporter medium that can be used in many industries such as to transport hazardous chemical waste and to place objects of interest at a shelf in hypermarkets. The communication between micro-controllers that controls the robot operation utilizes the I^2C bus. This bus is categorized under the serial bus. Serial bus is chosen as the main communication link due to its capability of reduces circuit complexity and eases system modifications when needed. In conclusion, the tele-operated robot is best implemented with I^2C bus as the communication medium between micro-controllers that control the robot's movement but not practical for the remote control communication medium to the robot. This is due to the fact that a robot is usually controlled from a far distance where else I^2C bus has a limited length of bus. Wireless communication such as blue-tooth, infra red and web based control are better suited for this purpose.

ACKNOWLEDGEMENTS

Alhamdulillah, with blessings from Allah S.W.T, this project has been completed successfully according to the targeted time frame. I would like to express my gratitude to Universiti Teknologi PETRONAS is particular for giving me a great opportunity to carry out this meaningful project.

My heartfelt gratitude and appreciation is extended to Mrs. Noor Hazrin Hany M. Haniff for her spirit of education as my supervisor. I have learned so much from her and this project would be less meaningful without her presence.

My acknowledgement would be incomplete without giving credit to all Final Year Project (FYP) Committee of Electrical and Electronics Engineering Department and all technicians, especially Mrs. Siti Hawa Hj Mohd Tahir that has equipped students with essential skills for self-learning. Its well-rounded graduate philosophy has proven to be useful in the project's implementation.

Finally, my appreciation goes to my fellow friends and my family who has helped me in various aspects and gave undivided support in completing this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Master	The device that initializes a transfer, generates the clock signal, and terminates the transfer
Slave	The device addressed by the master
Arbitration	The procedure that only allows one master to control the bus at a time
SDA	Serial Data
SCL	Serial Clock
RPM	Rotations per Minute
PWM	Pulse Width Modulation

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Robotics is vastly used in human's life nowadays. Tasks that are normally done by human are taken over by robotics periodically. They are used mostly in areas where harmful and difficult environment are present. The project involves a development of a robot that is remotely controlled by a remote-control. It has the capability of transporting objects of interest from one location to another. It can be used to stacks up books in libraries and to place goods on shelves in warehouses. The mobile robot uses wired remote-control. Due to this, the robot is known as the tele-operated robot. This project also focuses on the practicality of using a type of serial bus named Inter-Integrated Circuit (I^2C) as the communication medium between the components of the circuits. The basic understanding of the bus that includes the communication protocol, configurations, advantages and limitations are also covered. The bus which is vastly used in semi-conductors has become a standard in semiconductors manufacturing that includes sensors. Serial bus is considered as the better alternative to parallel bus in terms for reducing circuits' complexity and makes it easy for system modifications when needed.

1.2 Problem Statement

Lifting and organizing of objects in warehouses and hypermarkets are normally done by human. This led to time wastage since certain objects are heavy and some of the objects need to be placed at high level. This scenario caused difficulties to the employees that are assigned to the tasks and this can cause an increase of cost of that particular company. Thus, a mechanism needs to be developed that can replaced the human dependencies and can perform works in a more efficient way. Communication between chip to chip of devices is commonly done by parallel bus.

Parallel bus allows only one bit of instructions flow through it. That is to say that the communication requires many wires connected between the devices to perform certain commands or actions. Commonly, parallel bus requires 8 to 32 wires at one time. This will increase the manufacturing cost of products. I^2C Bus enables chip to chip communication using 2 wires instead of multiple wires as in parallel bus. Numbers of actions and commands can be executed by using the same two wires. It minimizes the cost and the complexity of the circuit without affecting the performance and the ability of the systems or devices.

1.3 Objectives

The main objective of this project is to design and develop a working tele-operated robot that has the ability to transport objects of interest from one location to another. The tele-operated uses I^2C bus for the communication between micro-controllers that control the robot operation.

1.4 Scope of Study

The project focuses on three essential aspects of a robot development. The aspects are structure and mechanism construction, electronic circuit development and programming algorithms generation. These three parts are integrated together to form a robot that can be controlled by any control medium or strategy. The structure and mechanism part deals with design and development of the robot's structure using different dimensions of aluminium rods and bars. The electronic circuit development deals with electronic components that can control the robot's operation. The programming algorithms on the other part deals with operation sequence and communication protocol between the parts of the robot.

1.5 Relevancy of Project

The tele-operated robot is integration of electronic field with mechanical field. In electronic side, electronic circuits that comprises of a remote-control and robot's controller are developed to control the robot maneuver and the mechanism. The operation sequence of the electronic circuits is controlled by the programming algorithms generated. The algorithms enable the system to transmit and receive particular signals. It also controls the operation of external devices such as DC motors. In the mechanical side, the structure and mechanism are designed and constructed by considering some foundation aspects such as weight distribution, load capacity and motors specifications.

1.6 Feasibility of Project

This project had been divided into two parts. They are the electronic part and the mechanical part. Throughout the project, focuses will be on the study of the I^2C bus and its application, the current methods used, the existing product available in the market and building a working prototype. A testing electronic circuit will be designed and constructed to test the communication capability between microcontrollers by using the I^2C bus. The data sent from the remote control should be successfully received by the other microcontroller. A testing platform is constructed to simulate the mechanisms that are controlled by microcontrollers that are connected together by the I^2C bus. With all the resources provided, it will be feasible within the time frame given.

CHAPTER 2
LITERATURE REVIEW

2.1 Overview

2.1.1 Definition of Robot

Robot is defined as an electromechanical system that conveys a sense that it has infant or agency of its own [13]. It has the ability to make choices based on its surrounding. A robot has numbers of movement axes that enables it to perform any desired tasks. Combinations of axes joints make up a robot. There are many different definitions of robots that are perceived by different organizations. According to Robotics Institute of America (RIA), robot is a re-programmable multifunctional manipulator designed to move materials, parts, tools and specialized devices through variable programmed motions for the performance of a variety of tasks [13]. They have classified four classes of robot [13]. And they are as listed below:

- A. Handling devices with manual control
- B. Automated handling devices with pre-determined cycles
- C. Programmable, servo controlled robots with continuous of point-to-point trajectories
- D. Capable of C specifications and also acquires information from the environment for intelligent motion

On the other hand, Japanese Industrial Robot Association (JIRA) had classified six classes of robot and they are as listed below [13]:

- Manual-handling devices actuated by an operator
- Fixed sequence robot
- Variable sequence robot with easily modified sequence of control
- Playback robot which can record a motion for later playback
- Numerical control robot with a movement program to teach its tasks manually

- Intelligent robot that can understand its environment and able to complete the task despite changes in the operating conditions

Despite different definitions perceived by organizations, the most accepted one is the one defined by the International Standard Organization (ISO). According to ISO, robot is a machine that is automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications [13].

2.1.2 Mobile Robots

A mobile robot is an automatic machine that is capable of movement in the given direction [12]. It is not stationary on one physical location. There are many types of mobile robot navigation. They are as listed below [12]:

- *Manual Remote or Tele-Op*

  A tele-operated robot is normally controlled via joystick. The joystick could be a wired-joystick or wireless-joystick. The robot is controlled by the operator at a distance. The function of the tele-operated control is to protect the operator from harm caused by the tasks environment.

- *Guarded Tele-Op*

  A guarded tele-operated robot has the ability to sense the condition of the surrounding in spite of being controlled by the operator. It has the ability to sense obstacles even when it is being controlled by the operator. The guarded tele-operated robot uses sensors to sense its' surrounding.

- *Line Following Robot*

  A line following robot is an autonomous robot that moves along a line. It operates by keeping its position at the center of the line. The robot moves until something has blocks its path. At this point, it stops and waits until the line is not blocked anymore.

- *Autonomously Randomized Robot*

  An autonomously randomized robot performs random motions. This includes jump off the wall and turn at certain degree without set any conditions to be met before that actions take place.

- *Autonomously Guided Robot*

  An autonomously guided robot is a type of robot that operates automatically with the information it perceived from the surrounding. It knows which path it should take and how it is supposed to move from the information perceived. This type of robot is also known as the Intelligent Robot or Artificial Intelligent (AI) robot.

2.1.3 Medical Robot

Robots are used periodically in medical field. They are normally used to perform surgeries. The usage of robots in surgeries offers precision and miniaturization of performance. There are basically three classes of robot surgery and they are as listed below [10]:

- Remote Surgery
- Minimally Invasive Surgery
- Unmanned Surgery

Remote surgery is performed where the surgeon controls the robot movement from a distance [10]. Specific buttons are pressed to perform certain actions such as move from one place to another, pick and place objects and also cut the patient disease areas. Minimally Invasive Surgery limits the surgeon control over the robot actions [12]. The Unmanned Surgery does not involve the controls from the surgeon. The robot performs actions based on its Artificial Intelligence (AI) created and programmed into it. It is able to diagnose and treat the patient according to the situation and the type of disease.

The da Vinci Surgical System which is becoming a popular robotic controlled surgery consists of three components. They are the surgeon's console, patient-side robotic cart with two arms manipulated by the surgeon and a high definition three-dimensional vision system [10]. The surgeon can control and manipulate the robot movement using the console. Then the arms behave as commanded by the surgeon. The surgeries that had been performed by using the da Vinci Surgical System include the surgeries on prostate cancer and hysterectomy [10]. Other applications that are related to robot surgery are cardiac surgery, gynecology, orthopedics and pediatrics.

2.1.4 Industrial Robot

The International Standard Organization (ISO) defines industrial robot as an automatically controlled, re-programmable, multipurpose manipulator programmed in three or more axes [9]. Industrial robot that resembles a robotic arm is mostly used for welding, painting, ironing, picking and placing objects, packaging and product inspection. The usage of robots in industries mostly in manufacturing sector increases productivity by providing enough speed, accuracy and vision guidance [9]. The only limitation of having a robotic arm in industries is the flexibility in the hand part. It requires the robotic arm to be able to hold different objects with different shapes, sizes and orientations without any problems.

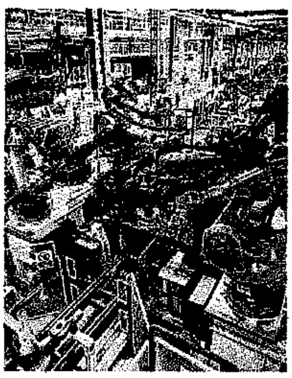


Figure 1: KUKA Robot

2.1.5 Military Robot

Influential countries such as the United States and the United Kingdom use robots for military purposes. The autonomous or remote-controlled robots are used to serve many functions. They include taking surveillance photo of the enemy territories and launching missiles from the air [11]. However, the usage of robots in military operations poses serious risks. The enemy may hacked into the robot's system and turn it around against the robot's owner [11]. The robot also has the limitation against moving across different terrains and obstacles.

Figure 2: Robots used for Military Operations

2.2 Principle of Operation

2.2.1 Telerobotics

Tele-operated robot is defined as the device that is controlled remotely by human operator [15]. The tele-operated robot that has the ability to perform autonomous work is known as the tele-robot [15]. There are two major components that form a tele-operated robot. They are the visual system and the control applications [15]. The visual system enables the operator to see the surrounding that the robot travels. From there, the operator can instruct the robot to move in certain directions or perform specific movements. However, there are some problems pertaining to the robot's vision system [15]. They are as listed below:

- Inadequate resolution of the camera
- Lag in the mechanical and computer processing of the movement and the response
- Optical distortion due to the lens condition

- Hand-eye coordination

For the vision system of the tele-operated robot, it is required to be equipped with interface between the robot vision and the operator. This is to ensure that the operator is able to see what the robot saw at that moment. The robot's interface is normally consists of a monitor, a mouse and a keyboard [15]. The images are displayed on the monitor and the operator can controls the robot by using the keyboard.

There are many types of control applications of a tele-operated robot. One of them is the usage of a remotely operated vehicle (ROV) for marine applications [15]. The ROV is normally used for deep-water exploration and inspection. It is also used to reach a location that is too dangerous for divers. This includes the location with high pressure or temperature.

2.2.2 Wired Communication

The interface between the robot and the operator can be in a wired form. The wired communication medium is known as the bus. The communication bus could be a serial bus or a parallel bus. Parallel bus enables data to be sent bit after bit simultaneously. This requires numbers of wires to sent one bit each simultaneously. For instance, 8 address wires sends 8 bit simultaneously to the receiving device. Since parallel bus requires 8, 16 or more data wires to send different data, the communication would be much easier but requires many communication wires. This increases the complexity and the cost of manufacturing.

Data and addresses sent through one wire must have its rules and synchronization. This is important as data and addresses that sent through common wire tend to have conflict with one another. Conflicts between them may cause the data or addresses to corrupt. This type of bus is known as the serial bus where every devices share a common wire to transmit and receive data [2]. On the other hand, data and addresses sent through 8 to 32 wires simultaneously are known as the parallel bus. In serial bus, all wires are eliminated except the data wire. Since the same wire is shared, all of the data, the selection (addresses) and the direction info (read/write) are multiplexed. A set of rules need to be developed for the situation to be realized.

There many types of serial programming available nowadays. They include Universal Serial Bus (USB), Controller Area Network (CAN) and Inter-Integrated Circuit (I^2C). These types of serial programming vary in terms of data transmission speed to suit any kinds of application. Table 1 shows the comparison of the serial bus types in terms of data transmission speed.

Table 1: Comparison of Serial Bus [2]

Type	Data Transmission Speed
CAN (1 wire)	33 kHz
I^2C	100 kHz
CAN (Fault Tolerant)	125 kHz
I^2C (High Speed Mode)	3.4 MHz
USB (1.1)	1.5 MHz or 12 MHz
High-Speed USB (2.0)	480 MHz

The type of serial bus to be selected for different applications must consider 3 essential factors [2]. The factors are as listed below:

- Speed or data rate
- Number of devices allowed to be connected along the same bus
- Total length of the wiring

Table 2 shows the comparison of three types of serial bus in terms of the 3 factors above.

Table 2: Comparison of Serial Bus Type in Terms of 3 Factors [2]

Type	Data Rate (bits/s)	Length (meter)	Nodes Typical Number	Node Number Limiting Factor
I^2C	400 k	2	20	400pF maximum
CAN 1 wire	33 k	100	32	Load resistance and transceiver current drive
USB (low speed, 1.1)	1.5 M	3	2	Bus specifications
Hi-Speed USB (2.0)	480 M	25	127	Bus and hub specifications

Universal Serial Bus (USB) is the most complicated type of serial bus. It has the ability to service different applications with different data rates and requirements. USB has 4 pin connector which comprises of power supply (2 pins), data and clock [2]. It is limited to short distance data communication depending on the configuration. Many devices can be connected to the USB since it is synchronous and transmits data in special packets like a network [2]. The advantages of USB are as listed:

- Simple plug and play
- Up to 127 devices can be connected together
- Interfaces to other communication exist

Controller Area Network (CAN) was first proposed by Bosch. The main objective for the development of CAN is to achieve reliable communications in critical control system applications [2]. To date, CAN is only used for automotive applications. Figure 3 shows the protocols of the CAN bus. CAN bus wires are pulled by resistors to their resting state called a 'recessive' state. A voltage called the 'dominant' state is forced when a transceiver drives the bus [2]. The identifier indicates the meaning of the data. All nodes receive and filter the identifier and decide whether to act on the data or not. Since many devices can act on the data, it is called the 'multicast' [2]. All devices can check the message for transmission error.

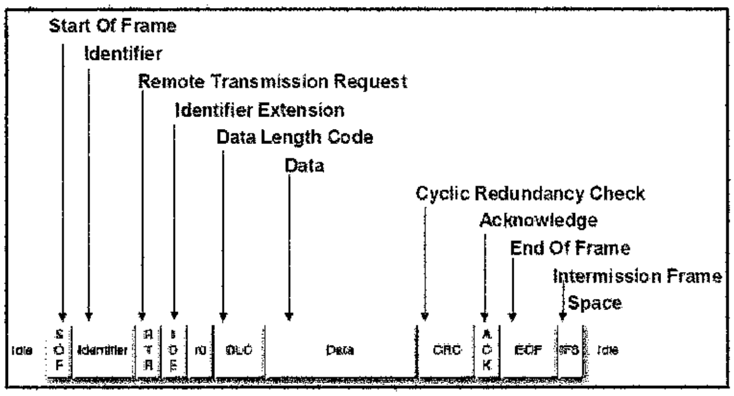


Figure 3: CAN Protocols [2]

Inter-Integrated Circuit (I^2C) was first established by Philips to control the operation of its television. I^2C is a simple bi-directional 2-wire bus [2]. The two wires are the Serial Clock (SCL) and Serial Data (SDA). In the industry nowadays, I^2C has become an industry standard and is widely used by major IC manufacturers. Its main operation is on the master-slave communication. Master device is the device that controls the whole system or other devices. The devices that are controlled by the master device are known as the slave devices [4]. Slave devices could be receiver-only devices or transmitters with the capability to receive and send data. I^2C is also capable to operate in multi-master mode [2]. In this mode, several devices can control other devices along the same bus. Each device or IC connected to the I^2C bus (slave device) is identified by its own unique address. The master device will select the slave device based on the unique addresses [2]. The selected slave device will receive the data send by the master device. The selection of slave devices and data transmission is done through specific I^2C bus protocol. The protocol follows 4 steps process [4] that is shown below:

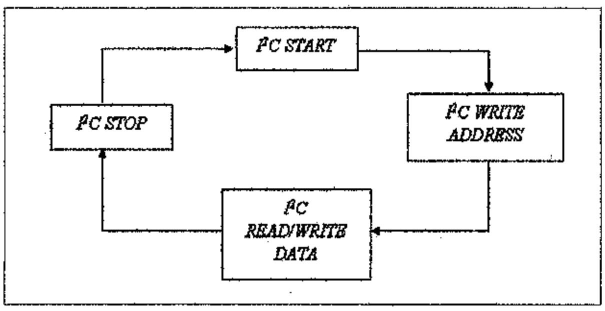


Figure 4: 4-Steps of I^2C Protocol

2.3 Existing Product in the Market

The current product that exists in the market is the Automated Guided Vehicle (AGV). It operates autonomously and uses sequential algorithm programmed into it. The AGV needs sensing assistance such as sensors to analyze its environment [13]. The signal from the sensors enables the AGV to react or perform specific actions in accordance with its surrounding. It uses the Artificial Intelligence (AI) to perform specific operations with the existence of obstruction near it. The AGV is widely used in warehouses where it id used to transport goods from one place to another. The limitations or risks posed by the AGV include the following [13]:

- The AGV is out of control if it's systems fail
- Constant monitoring is needed to ensure the AGV perform its' tasks as instructed
- The AGV needs to be re-programmed to carry out other tasks than the specified one

Thus, a tele-operated robot is the better alternative since the robot operation can be controlled from a distance by the operator. This type of operation can reduce any mistake that can be done by using the autonomous one.

CHAPTER 3

METHODOLOGY / PROJECT WORK

3.1 Procedure Identification

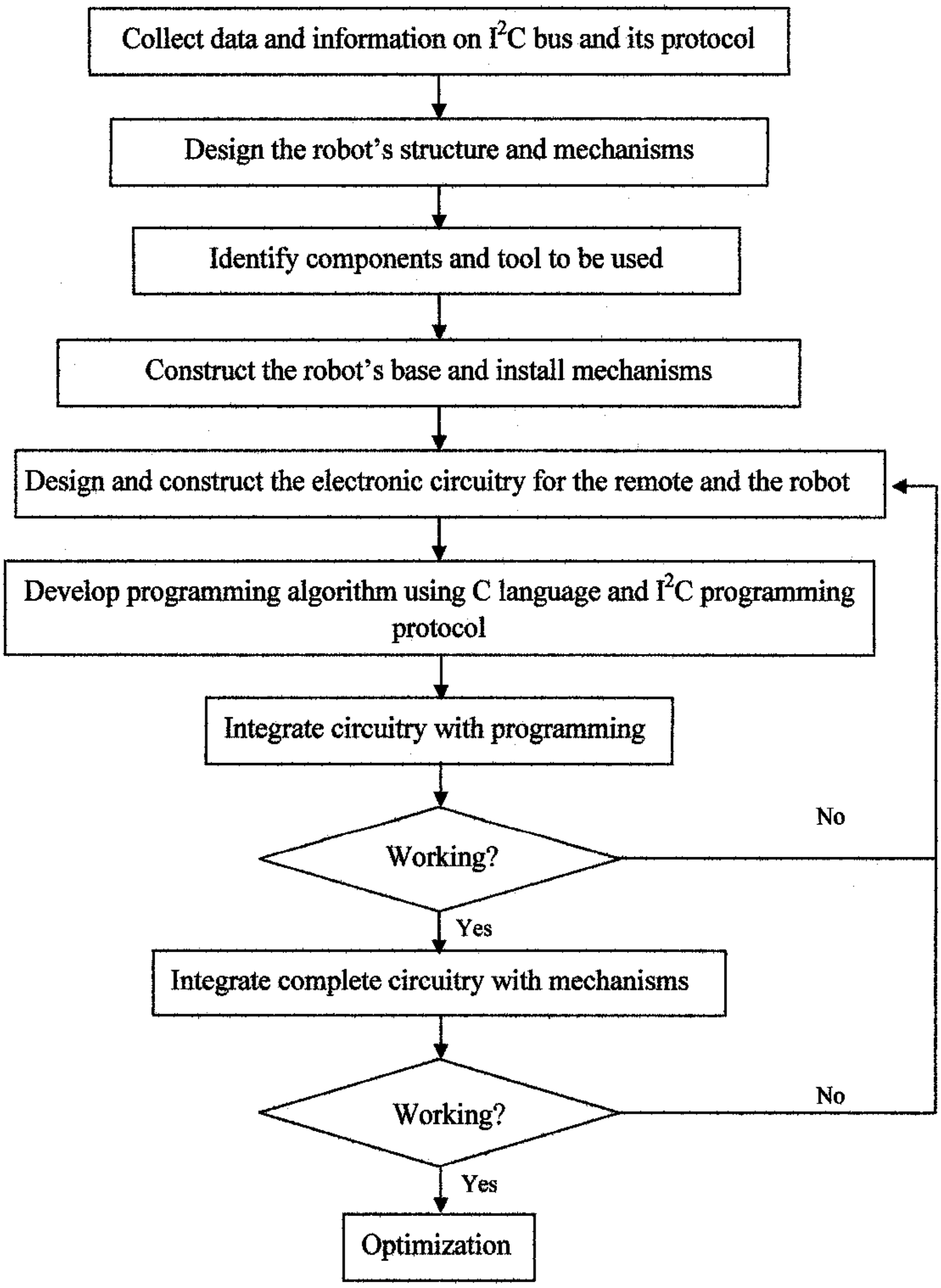


Figure 5: Procedures for Designing and Constructing Tele-Operated Robot

3.2 Project Implementation Steps

The project is implemented base on the activities flow diagram shown in the previous page. The activities cover the three aspects of the tele-operated robot development; structure construction, electronic circuits development and programming algorithms generation. Listed below are the methodology used for the tele-operated robot development.

3.2.1 Data Collection

Data pertaining to the three aspects of the robot development are collected prior to the robot construction. This includes the existing product in the market, technical specifications of the I^2C bus, the components specifications and the electronic circuits. The data are obtained through journals, internet, books and also datasheets.

3.2.2 Design the Structure of the Robot

The design of the robot is first sketched. The sketch includes the type of materials to be used and their measurements. The parts of the material that will be drilled and made holes are also marked so that correct joints between different parts can be done. The design includes the base and the mechanism of the robot. The base is designed to have adequate width to cater the robot's stability. The gripper and the lifter part of the mechanism are designed in a away that they can sustain the weight of the load. Couplings to attach the wheels and the motor shafts together are also designed.

3.2.3 Components Identification

The components identification stage covers the identification and selection of electronic components and the materials for the structure. The identification and selection of the components are based on the requirement and the specification of them. The components that met the required operating conditions are selected. In the mechanical side, aluminium bars and rods are selected since they are light in weight and stable enough to cater the load. The DC motor for the lifter part is chosen since it has the capability of sustaining big load due to its gearing system that enables the motor to have higher torque. The components are also selected based on the budget allocated for this project.

3.2.4 Structure Construction

The structure of the robot is constructed based on the design drawn earlier with the usage of the selected materials. The individual parts of the structure are first marked and cut using the automatic cutter machine. The individual parts that have been already cut are then join together. They are joined by using rivets and screws. The permanent joints used rivets, while temporary ones used screws. The constructed structure is then ensured to be stable enough since the lifter part is about 1 meter in height. DC motors are assembled at the desired location on the structure. The motors are used for controlling the robot maneuver, the gripper and also the lifter. They are equipped with individual couplings. The couplings are used to join the shaft of the motors with the rotating parts of the structure.

3.2.5 Electronic Circuits Development

The electronic schematics obtained from the data collection stage are then developed. There are a total of seven circuits developed for the robot control system. The circuits include the remote-control, the robot controllers, power supply and motor drivers. The circuits are developed and tested individually before being integrated together. Every circuit is first developed on bread-boards and tested their functionality. The circuits are integrated together and then transferred onto veroboards and tested as the final product.

3.2.6 Programming Algorithms Generation

The programming algorithms are generated for the use of the micro-controllers that control the operation sequence of the robot. The micro-controllers used I^2C bus communication protocol for inter-micro-controllers communication. The algorithms are written into the micro-controller by using a compiler program and also a micro-controller burner. Different styles of programming algorithms has been generated and tested to suit the required control operations.

3.2.7 Electronic Circuits and Programming Algorithms Integration

The integration of the electronic circuits and the programming algorithms are meant to observe the capability of the micro-controllers to control the whole operations of the robot. They are integrated together by writing the algorithms into the micro-controllers.

3.2.8 Three Parts Integration

The integration of the structure, the electronic circuits and the programming algorithms are done by connecting the electronic circuits and the motors together. This is to ensure the controllability of the robot movement by the remote-control. The selected part of the robot should move and operated based on which buttons of the remote-control is being pressed.

3.2.9 Troubleshooting

The troubleshooting stage comprised of the three aspects mentioned. The troubleshooting part of the structure includes the re-positioning of the DC motors onto the structure. The electronic circuits part troubleshooting is done by checking the connectivity of the components along the veroboards and checking the output voltage at the micro-controllers pins. This can be done by using multi-meter and logic probe.

As for the troubleshooting of the programming algorithms, modified algorithms are generated and re-write into the micro-controllers.

3.3 Tools Required

The project is delivered by constructing a working model of tele-operated robot that communicates using the I^2C Bus and its programming protocol. The tools and equipments required to carry out the project are as listed below:

- Logic Probe
- Multimeter
- Soldering Iron
- Solder Sucker
- Riveter
- Drilling Machine
- Automatic Cutter Machine
- Programmable Integrated Circuit (PIC) Programmer
- PIC C Compiler

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Structure Construction

4.1.1 Base Construction

The structure construction of the robot is divided into two parts; the base part and the mechanism part. The idea is to construct the base platform first to test the workingness of the circuits' part when they are integrated with the hardware. Then the mechanism part is designed and added onto the base.

The robot's base provides the basic testing platform for the remote control circuit and programming algorithm via I^2C communication. Shown below in Figure 6, 7, 8 and 9 are the completed robot's base from different kind of views. The parts and materials used in the construction are as listed below:

- 25mm x 50mm rectangular hollow aluminium bar
- Caster ball
- 2 DC motors
- 2 wheels complete with couplings

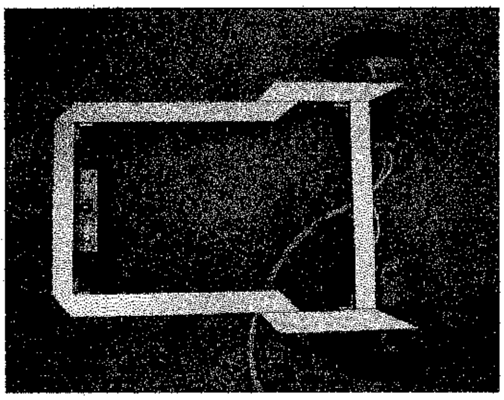


Figure 6: From Top View

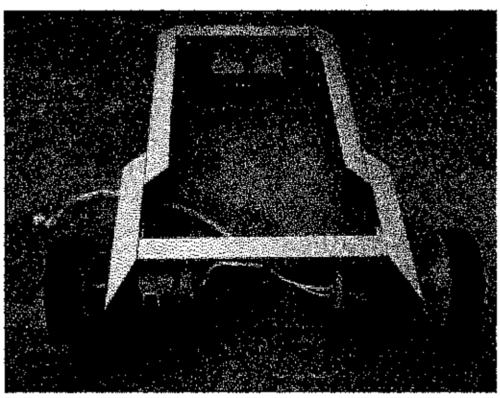


Figure 7: From Back View

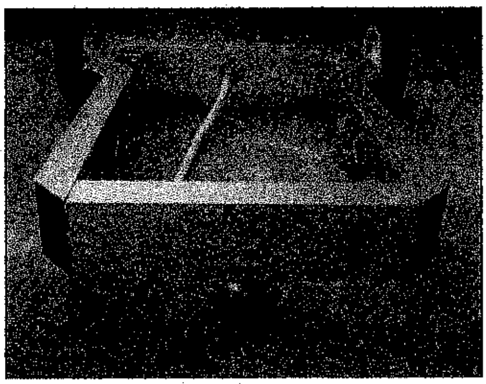


Figure 8: From Front View

Figure 9: From Side View

The base is made of hollow rectangular aluminium tube with the dimension of 50mm x 25mm in the cross section. Two dc motors is placed at the back of the base. The dc motors control the robot movement. The dc motors had been coupled with their respective wheels. Couplings are used to couple the dc motor and the wheel together. This is to ensure that the wheels rotation synchronize with the dc motors' rotation. A caster ball is mounted at the front part of the robot's base. The usage of the free rotation caster ball eases the movement of the robot where the movement will only be controlled by the two back wheels and does not depend on the front wheel. The robot's base must be wide enough to sustain any forces acting on it when the mechanisms are installed to it. The dimension of the robot's base gives an impact on the centre of gravity of the robot and the total weight distribution.

4.1.2 Mechanism Construction

The tele-operated robot has the capability to lift and transport objects from one place to another. A mechanism has been constructed to meet this requirement. The mechanism (refer Figure 10) is attached to DC motors that are controlled by the micro-controllers. The mechanism includes the lifting part and the gripping part. The structure of the mechanism part used different sizes of aluminium tubes.

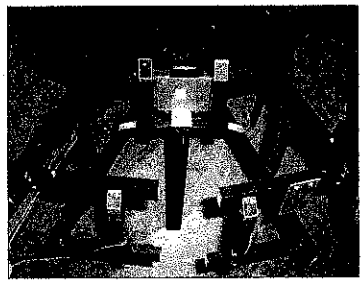


Figure 10: Mechanism Part

4.2 Electronic Circuits

All the mechanisms of the tele-operated robot are controlled by electronic circuitries. The controlling part integrates the hardware with the software. Components like the Programmable Integrated Circuit (PIC) need to be programmed in order for the mechanisms to work. The components used for the circuits are as listed in the previous section.

4.2.1 Components Selection

- Micro-controllers

  The micro-controller used for the research is the Programmable Integrated Circuit (PIC). It has a built-in memory, built-in functions, input and output ports. Programming languages such as C, C++ and assembly language can be used to program the PIC. There are numbers of PIC families. They are segregated in terms of their capabilities and characteristics. They can vary from those that have 18 pins and 40 pins. PIC that has more number of pins will have more built-in functions and greater capability. The most common PIC families are the 16F8x and the 16F8xx. Figure 11 and 12 in the next page show the PIC 16F877 and 16F84 family that will be used in the research work. PIC works on 5V DC power supply. Over voltage may cause the PIC to explode. The built-in functions of the PIC include the following:

  - Inter-Integrated Circuit (I^2C)

- Pulse Width Modulation (PWM)
- Analog to Digital Converter (A/D)
- Transmitter and Receiver for RS232

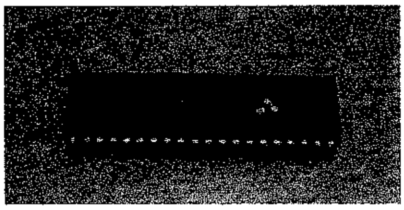


Figure 11: 16F877 PIC (40 pins)

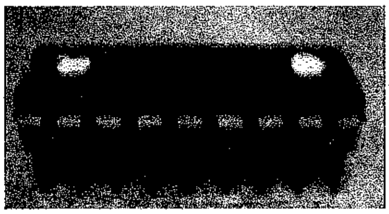


Figure 12: 16F84 PIC (18 pins)

- Resonator

  The crystal clock or resonator provides the pulse to the PIC. These pulses are needed for the PIC to operate. It has the variety range of values depending on the need of reaction period. Basically there are two types of crystal clock; 2-pins and 4-pins. The 4-pins crystal clock does not require ceramic capacitors to be connected to it like the 2-pins. Figure 13 shows the 2-pins crystal clock and Figure 14 shows the 4-pins resonator.

Figure 13: 2-pins

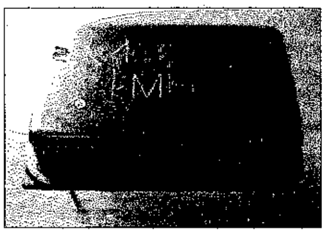


Figure 14: 4-pins

- Voltage Regulator

  Voltage regulator is used to convert or regulate voltage from one level to another. PIC requires 5V power supply and some power supply cannot supply that specific amount of voltage. Thus, voltage regulator comes into the picture. Voltage regulator has three pins; input, ground and output. 7805 voltage regulator regulates input power supply to 5V output dc, where else 7806 regulates to 6V output dc. The input must have higher voltage level compared to the required output voltage. Figure 15 shows the voltage regulator.

Figure 15: Voltage Regulator

- Transistor

  Bi-polar Junction Transistor (BJT) is made up of two different substrates known as the P-substrate and the N-substrate. The P-substrate has the 'hole' as its majority carrier and electrons as its minority carrier. The N-substrate has the vice-versa majority and minority carrier. BJT is multi-function semiconductor component. It can be used to amplify signals and can act as an automatic switch. BJT has 3 pins known as the Collector (C), the Base (B) and the Emitter (E). To turn on the BJT, the voltage barrier between the Base and the Emitter junction need to be overcome. The amount of voltage supplied to the Base pin must be higher than the threshold voltage of the Base-Emitter junction which is known as the V_{BE}. For silicone-type BJT, the V_{BE} is 0.7V while for germanium-type BJT, the V_{BE} is 0.3V. Generally, BJT is classified into 2 categories; NPN-type and the PNP-type. The common model for the NPN-type is 2N3904 and for the PNP-type is 2N3906.

Figure 16: Transistor

- Relay

  Relay is used to separate two circuits and can act as a switch when it is triggered. It consists of 2 ends of a coil, a common pin, a normally closed pin and a normally opened pin. The 2 ends of the relay's coil are connected to the primary circuit that will trigger the relay. The common pin together with the normally opened pin or the normally closed pin is connected to the secondary circuit. At a time, either normally closed pin or normally opened pin is connected to the secondary circuit. The unused pin will be grounded. The term 'normally opened' is defined as the pin that is not connected to the common pin during the time which the relay is not triggered by the primary circuit. The normally closed circuit on the other hand has the vice-versa condition. To trigger the relay, the threshold voltage of the relay needs to be overcome. For instance, the value of the relay is 5V. To trigger the 5V type relay, a voltage level supplied to the relay through its coil must be higher than 5V. Once it is triggered, the common pin will latched together with normally opened pin.

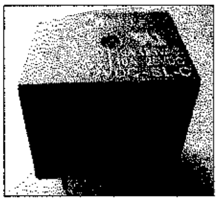


Figure 17: Relay

- Diode

  Diode is used to allow only one direction of current to flow through it. It does not allow the current to flow in the opposite direction. This is known as the forward-bias operation. Normally, diode is used to regulate voltage from the alternating form to the direct form. On the other hand, diode is normally used for circuits' protection purposes to prevent voltage feedback that may cause damages to circuits from happening. Listed below are some of the diode types that have different operating current limit that suits different kind of requirements:

  - BY299 – also known as the power diode that can sustain up to 8A of current
  - LN4148 – also known as the zener diode that operates in reverse-bias mode and can sustain up to 800mA of current

- 1N4007 – can sustain up to 7A of current

From the three mentioned above, only the 1N4007 type is not equipped with 'fast recovery' mode. This mode is vital in preserving the life of the micro-controller connected to it.

Figure18: Diode

- DC Motor

  DC motors have their own rotation rate known as the RPM. RPM (Rotations per Minute) is normally inversely proportional to their torque. The RPM of the motor used is low since the mechanisms connected to them must be controlled carefully to ensure its accuracy. The power supply for DC motors is normally 12V up to 24V. Figure 19 shows the DC motor.

Figure 19: DC Motor

4.2.2 Remote Control and Robot Controller

The tele-operated robot involves the communication between the remote circuit and the robot's controller circuit. The PIC of the remote circuit gives command to the PIC of the robot's controller circuit to execute any commands given. The PIC of the robot's controller circuit will then controls the movement of the mechanisms connected to it.

The figure below shows the 2 slave PIC circuits connected to the master PIC circuit. The 2 slaves are meant to control the robot's wheel movement and the mechanisms movement separately. Each slave PIC will be activated based on which buttons of the remote circuit are being pressed. All three circuits shares the same crystal resonator to enable synchronize operation between them. This condition is the basic part for the I^2C to work. The value of the crystal resonator tested is 4 MHz. This means that the operation of the circuit is repeated 4 million times in one second based on the following formula:

Period (t) = 1 / Frequency (Hz)

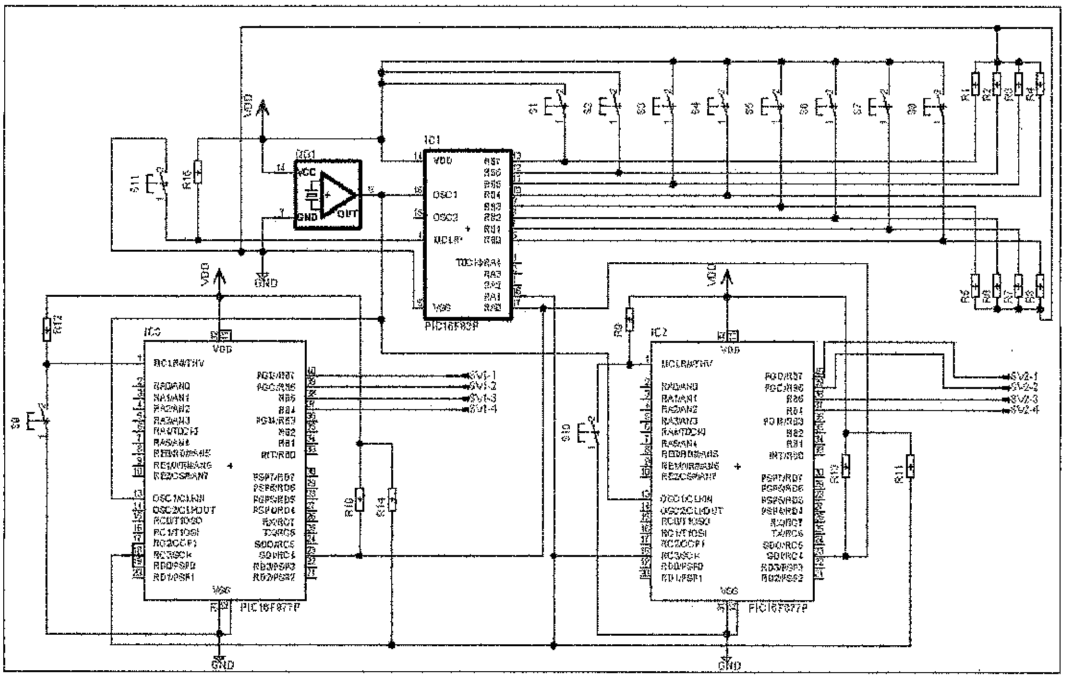


Figure 20: Master and Slave Circuits Integrated Together

Every device connected to the I^2C bus needs to have individual set of pull-up resistors that connect the bus to the power supply. This is done due to the need of pulling the I^2C bus high whenever the bus is idle. An idle bus indicates that any Masters along the I^2C bus can initiate the START condition to enable them to communicate with other devices along the bus. The value of the pull-ups varies according to the value of the power supply voltage, bus capacitance and the number of devices connected along the bus.

The wiring pattern of the I^2C bus needs to be taken into consideration to minimize interference along the bus lines. The wiring pattern shown in Figure 21 is used if the bus lines exceeds 10cm and includes the V_{DD} and V_{SS} lines.

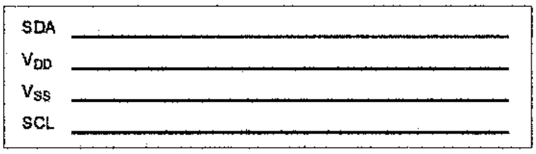


Figure 21: Wiring Pattern [4]

4.2.3 Power Supply

A 12V battery will supply power to the whole circuits. However, components like the micro-controller (PIC) and servo require 5V and 6V supply respectively. Over voltages for these 2 components can caused damages. Thus, the 12V supply needs to be converted into 5V and 6V. In this case, a voltage converter circuit has been designed and constructed as shown in Figure 22 below. The components used for the conversion to take place are the 7805 and 7806 voltage converters. The 7805 converts the 12V supply into 5V, where else 7806 convert the 12V supply into 6V. The LEDs on the circuit board are used to indicate that the voltage converters are working. The voltage converters have 3 pins; supply from battery, ground and the converted output voltage. A capacitor needs to be connected between the supply from battery's pin and the converted output voltage's pin. The presence of the capacitor is to ensure that the output voltage is a stable one. Heat-sink is connected to the voltage converters to allow the dissipation of heat generated at the voltage converters.

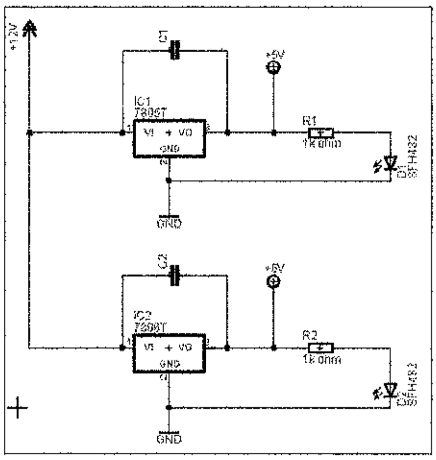


Figure 22: Voltage Converter Circuit

4.2.4 DC Motor Driver with Speed Control

DC motors cannot be driven directly from the micro-controller (PIC). This is because the voltage output from the PIC is in the range of 3 to 5 volts. The standard operating voltage of dc motors are between 8 to 12 volts. Thus, external circuit needs are designed and constructed to control the operation of the dc motors and to interface them with the PIC. This is where motor driver circuit comes into the picture. The motor driver circuit has the ability to control the direction and the speed of the dc motors. Relays will be used when it comes to control only the rotation direction of the dc motors. To control the rotation speed on the other hand requires a specific component called the h-bridge. In the project, two types of DC motor drivers are used. 'DC motor driver with speed control' is used to control the robots' wheels while the 'DC motor driver without speed control' is used to control the mechanisms' movement.

Figure 23 shows the schematic diagram of the DC motor driver with PWM. PWM means Pulse Width Modulation. It is used to vary the speed of the dc motors' rotation. In the schematic given, the main component used is the L298 h-bridge. It can control the speed and the direction of the dc motors rotation. One L298 can control 2 dc motors at a time. It receives specific signal from the PIC to perform specific command.

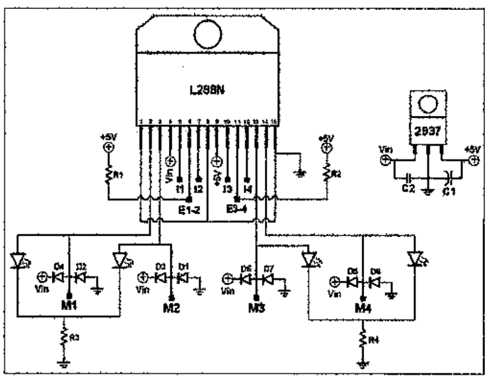


Figure 23: DC Motor Driver with Speed Control Circuit Schematic

4.2.5 DC Motor Driver without Speed Control

H-bridge is not required when it comes to control the rotation direction of the DC motors with constant speed. Relays are used to replace the h-bridge in this case. One DC motor requires one motor driver circuit that uses relays as shown in the figure below. The output from the micro-controller is connected to the transistor. The transistor will act as an automatic switch when the voltage level provided to the Base (B) pin of it exceeds its threshold voltage value. In this case, the threshold voltage value is 0.7V since it is a silicon-type transistor. Each relay controls one end of the DC motor. Thus, 2 relays are required to control one DC motor. The direction of the DC motor rotation is varied by varying the output of the micro-controller (refer Table 3).

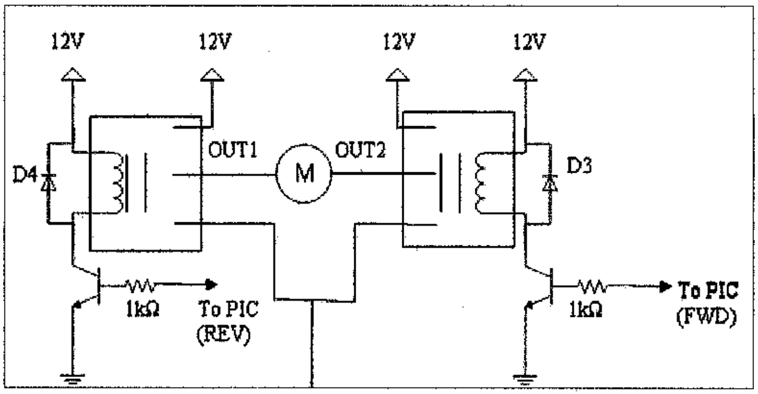


Figure 24: DC Motor Driver without Speed Control Circuit Schematic

Table 3: DC Motor Rotation Direction

Relay 1 Coil State	*Relay 2 Coil State*	*DC Motor Direction*
0	0	No Rotation
0	1	Clockwise
1	0	Counter Clockwise
1	1	Not Allowed

4.3 Programming Algorithms

4.3.1 Conditional Master and Conditional Slaves

The main idea of the programming algorithm for the "Conditional Master and Conditional Slaves" is to allow only one slave PIC to execute certain command when it is called by the master PIC. This is done by assigning the 2 slaves with 2 different addresses to differentiate them with one another. In this case, the slave 1 is assigned with the address of 0x10 while slave 2 is assigned with 0x20.

Each slave PIC would only execute when individual 4 buttons are being pushed at the master circuit. That means each slave will execute 4 commands only. Slave 1 has been assigned to control the robot maneuver while slave 2 to controls the robot's mechanism. Whenever buttons to control the robot's maneuver are pressed, the master PIC will 'call' slave 1 and sends data to it while slave 2 remains in idle state. The same goes when the robot's mechanisms buttons are being pressed where slave 2 will execute commands and slave 1 remains in idle state. Both slaves will be in idle state when neither buttons are being pressed.

The outcome from the testing is that the communication between them works. Each slave is able to execute command when the corresponding buttons were pushed. However, some problems had risen during the testing. The output LEDs at slave 1 are dim and not bright as compared to the LEDs at slave 2. This may happened due to low level of output voltage at the LED. The troubleshooting part of this problem has been completed where resistors are connected in series with the LED. The output voltage at the specified pins has the value between 3.3V to 5V whenever the output is HIGH.

4.3.2 Real-time Master and Conditional Slaves

The programming algorithm for the "Real-time Master and Conditional Slaves" repeats the 4-steps of the I^2C serial programming; start, call address, write data and stop. The repetition is done to call and send data to the both slave micro-controllers at the same time. In reality, slave 0x20 should have some delay compared to the slave 0x10 since it received data from the master micro-controller later than slave 0x10. Although the delay between them exists, it is negligible for slow or medium rate communication.

The objective of the programming algorithm generated is achieved where both slave 0x10 and 0x20 responded to the push buttons that are being pushed at the master micro-controller side. As expected, 0x20 responded later than 0x10 with a very minimum period of time and almost undetectable. However, some problems are detected as the circuit is tested. The problems are shown below:

- Only one slave responded as the push buttons are being pressed
- None of the slaves responded and seems to be in the idle state

Even though the problems stated above occurred, their occurrences are at the minimal level where most of the time both slaves responded to the data sent by the master micro-controller.

The 'real-time master and conditional slaves' is more practical and ideal compared to the other since the master PIC continuously sends binaries to slave PIC. The execution part only happens in the slave PIC. This ensures that any modifications of the programming algorithm only need to be done in the slave PIC. This is what a standard remote-control is made of. On the other hand, the 'real-time master and conditional slaves' is not suitable when it comes to devices that perform their own operation that are not related to the whole system. This is due to the fact that the devices are continuously called by the Master even when the remote buttons that are not related to them are pressed.

4.4 End Product

Figure 25, 26 and 27 show the end product that integrates the 3 parts of the robot. The parts are the structure and mechanism, the electronic circuits and the programming algorithms.

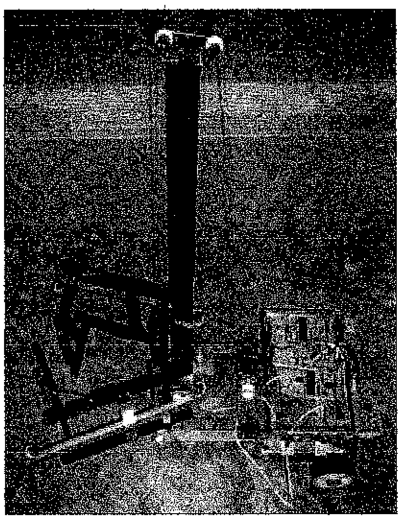


Figure 25: The End Product

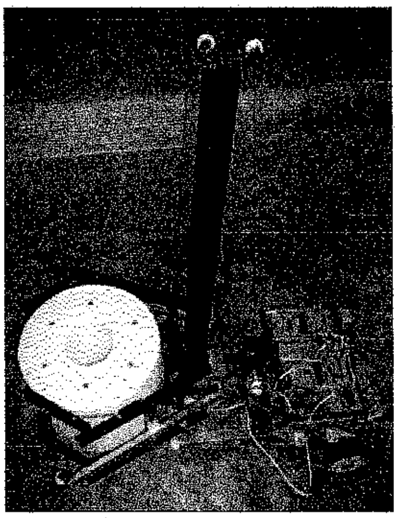


Figure 26: The Robot Grips the Object

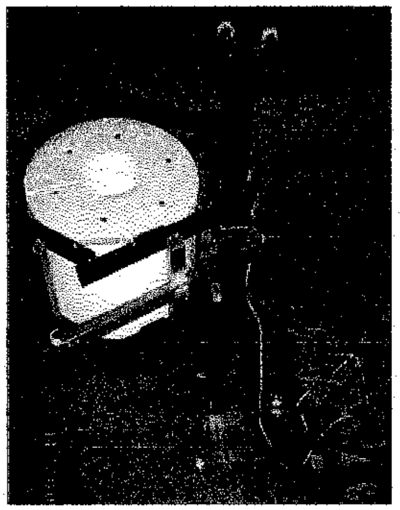


Figure 27: The Robot Lifts the Object

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

In this project, the objective of developing a tele-operated robot that has the capability as a transporter is achieved. It is able to transport objects from a location to another by using its gripper and lifter. I^2C bus has been fully utilized in developing the tele-operated robot. The project has lead to understanding of many robotics aspects including the mechanism construction, electronic circuit development and programming algorithms generation. The project has shown that I^2C bus is an ideal communication bus between micro-controllers since it has the ability to cater the problems faced by parallel bus usage in electronic systems. I^2C bus has the ability to reduce the circuits' complexity, reduce the cost, has better accuracy and eases systems modifications. The understanding of I^2C operations enables any electronic components in systems to be manipulated at ease. The project also highlighted some set-backs in terms of the communication between the electronic circuits. This includes the I^2C bus length limitation, real-time programming algorithms operation and the proper selection of pull-up resistors. The improvements that can be made in the future includes replace the wired communication with wireless communication and add more capabilities to the robot. This includes by adding vision system and making the robot able to cross different kinds of terrain without losing its balance.

REFERENCES

[1] Micro Computer Control, I^2C Bus Technical Overview. Retrieved from www.mcc-us.com

[2] Jean-Marc Irazabal and Steve Blozis (2003). I^2C Bus Application Notes. Philips Semiconductor

[3] Philips Semiconductor (2000). The I^2C Bus Specification. Philips Semiconductor

[4] Philips Semiconductor (1995). How to use the I^2C Bus . Philips Semiconductor

[5] ROBOCON 2007 Committee. ROBOCON 2007 Themes and Rules. Retrieved from www.vtv.org.vn/robocon

[6] Seattle Robotics. Servo Motor, Retrieved from www.seattlerobotics.org

[7] Matanya Elchanani, Tarek M. Sobh and Raul Mihali. Remote Learning: A World-Wide Web operated Robot Arm. University of Bridgeport

[8] Lukas Hohl, Ricardo Tellez, Olivier Michel and Auke Jan Ijspeert (2005). Aibo and Webots: Simulation, wireless remote control and controller transfer. Swiss Faderal Insttute of Technology

[9] Industrial Robot. Retrieved from http://en.wikipedia.org/wiki/Industrial_robot

[10] Medical Robot. Retrieved from http://en.wikipedia.org/wiki/Medical_robot

[11] Military Robot. Retrieved from http://en.wikipedia.org/wiki/Military_robot

[12] Mobile Robot. Retrieved from http://en.wikipedia.org/wiki/Mobile_robot

[13] Robot. Retrieved from http://en.wikipedia.org/wiki/Robot

[14] Robot Control. Retrieved from http://en.wikipedia.org/wiki/Robot_control

[15] Tele-robotics. Retrieved from http://en.wikipedia.org/wiki/Tele-robotics

APPENDICES

APPENDIX A

PIC 16F84 MICRO-CONTROLLER (PIN LAYOUT)

PIC16F8X

MICROCHIP

18-pin Flash/EEPROM 8-Bit Microcontrollers

Devices Included in this Data Sheet:

- PIC16F83
- PIC16F84
- PIC16CR83
- PIC16CR84
- Extended voltage range devices available (PIC16**LF**8X, PIC16**LCR**8X)

High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single cycle except for program branches which are two-cycle
- Operating speed: DC - 10 MHz clock input
  DC - 400 ns instruction cycle

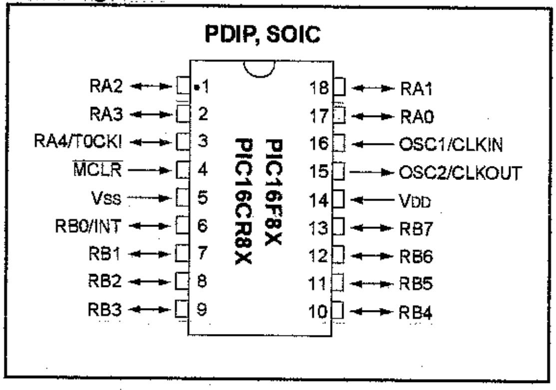
Device	Program Memory (words)	Data RAM (bytes)	Data EEPROM (bytes)	Max. Freq (MHz)
PIC16F83	512 Flash	36	64	10
PIC16F84	1 K Flash	68	64	10
PIC16CR83	512 ROM	36	64	10
PIC16CR84	1 K ROM	68	64	10

- 14-bit wide instructions
- 8-bit wide data path
- 15 special function hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt on change
  - Data EEPROM write complete
- 1000 erase/write cycles Flash program memory
- 10,000,000 erase/write cycles EEPROM data memory
- EEPROM Data Retention > 40 years

Peripheral Features:

- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 20 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Pin Diagrams



Special Microcontroller Features:

- In-Circuit Serial Programming (ICSP™) - via two pins (ROM devices support only Data EEPROM programming)
- Power-on Reset (POR)
- Power-up Timer (PWRT)
- Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Code-protection
- Power saving SLEEP mode
- Selectable oscillator options

CMOS Flash/EEPROM Technology:

- Low-power, high-speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 6.0V
  - Industrial: 2.0V to 6.0V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 μA typical @ 2V, 32 kHz
  - < 1 μA typical standby current @ 2V

APPENDIX B

PIC 16F877 MICRO-CONTROLLER (PIN LAYOUT)

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

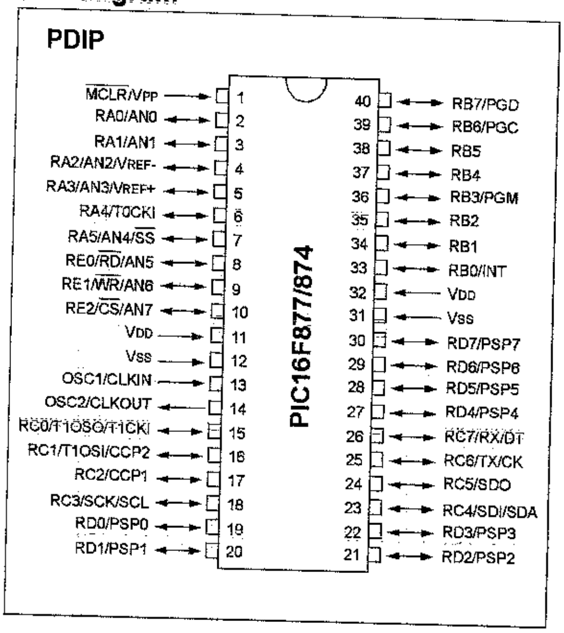
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM)
  Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 μA typical @ 3V, 32 kHz
  - < 1 μA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^{2}C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

APPENDIX C

RELAY

宁波天波港联电子有限公司

NINGBO TIANBO GANGLIAN ELECTRONICS CO.,LTD

HJR-3FF 技术指标

:编号:
SPECIFICATION

: NUMBER: P-08KB201A

DATE: 2001/11/02

CONTACT DATA触点参数

act Form触点形式	1A/1C	
act Material触点材料	Silver Alloy	
act Ratings触点负载	1A:10A 240VAC, 15A 125VAC 12A 120VAC/24VDC	1C: 7A 240VAC/10A 120VAC/24VDC 15A 125VAC
Switching Voltage最大转换电压	250VAC/30VDC	
Switching Current最大转换电流	15A	
Switching Power 最大转换功率	2770VA/240W	
act Resistance接触电阻	100mΩMax	at 6VDC 1A
Expectancy Electrical 电气寿命	100, 000	Operations(at30Operations/minute)
Machanical机械寿命	10, 000, 000	Operations

GENERAL DATA一般参数

ation Resistance绝缘阻值		100MΩMin at 500VDC
ctric Strength Between Open Contacts触点间耐压		750VAC(for one minute)
een Contacts and coil触点与线圈间耐压		1500VAC(for one minute)
ate Time吸合时间		10ms
se Time释放时间		5ms
erature Range环境温度		-40℃to+85℃（105℃）
Resistance冲击	Operating Extremes动作极限	10G
	Damage Limits破坏极限	100G
tion Resistance振动		10-50Hz 1.5mm
dity湿度		40—85%
ıt重量		Approx 10g
Standard安全标准		UL CUL TUV CQC

COIL DATA线圈参数

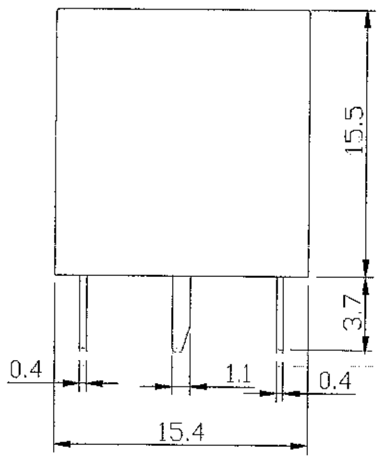
inalVoltage 定电压 (VDC)	Coil Resistance at20℃±10%(Ω) 线圈阻值		Max Operate Voltage 最大吸合电压 VDC	Min ReleaseVoltage 最小释放电压 VDC	Max ApplicableVoltage 最大过载电压 VDC
	0.36W	0.45W			
3	25Ω		2.25	0.3	3.9
5	70Ω		3.75	0.5	6.5
6	100Ω		4.5	0.6	7.8
9	225Ω		6.75	0.9	11.7
12	400Ω		9	1.2	15.6
18	900Ω		13.5	1.8	23.4
24	1600Ω		18	2.4	31.2
48	6400Ω±15%	5100Ω±15%	36	4.8	62.4

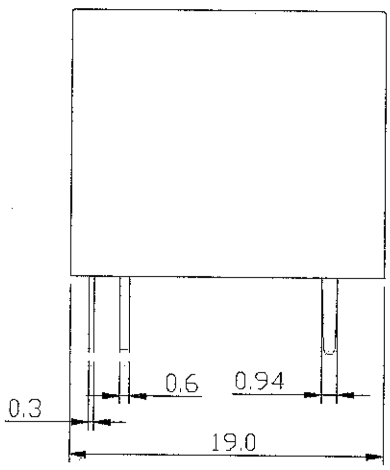
■ORDERING CODE定购代码

HJR-3FF—12VDC—S—Z

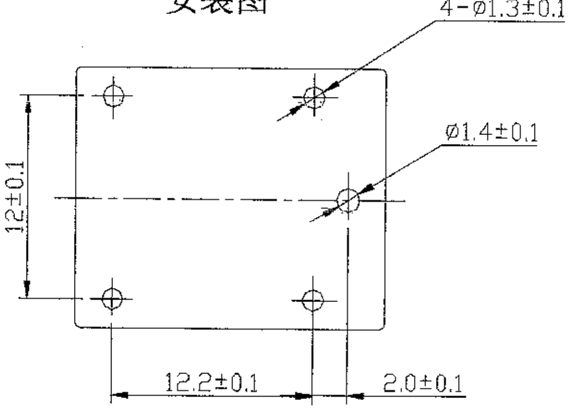
- Z Form C 触点转换形式
- H Form A
- Sealed 密封
- Coil Nominal Voltage 线圈额定电压
- Relay Model 继电器型号

■OVERALL AND MOUNTING DIMENSIONS安装图

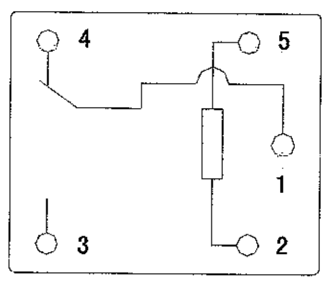




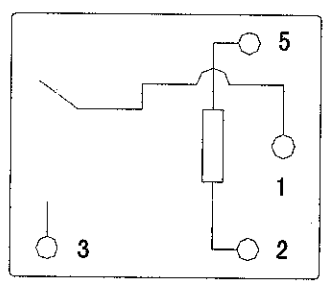
安装图



接线图



Form C



Form A

APPENDIX D

TRANSISTOR 2N3904 (PIN LAYOUT)

FEATURES

- Low current (max. 200 mA)
- Low voltage (max. 40 V).

APPLICATIONS

- High-speed switching.

DESCRIPTION

NPN switching transistor in a TO-92; SOT54 plastic package. PNP complement: 2N3906.

PINNING

PIN	DESCRIPTION
1	collector
2	base
3	emitter

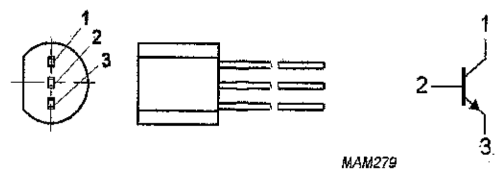


Fig.1 Simplified outline (TO-92; SOT54) and symbol.

LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CBO}	collector-base voltage	open emitter	–	60	V
V_{CEO}	collector-emitter voltage	open base	–	40	V
V_{EBO}	emitter-base voltage	open collector	–	6	V
I_C	collector current (DC)		–	200	mA
I_{CM}	peak collector current		–	300	mA
I_{BM}	peak base current		–	100	mA
P_{tot}	total power dissipation	$T_{amb} \leq 25$ °C; note 1	–	500	mW
T_{stg}	storage temperature		–65	+150	°C
T_j	junction temperature		–	150	°C
T_{amb}	operating ambient temperature		–65	+150	°C

Note

1. Transistor mounted on an FR4 printed-circuit board.

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
$R_{th\,j\text{-}a}$	thermal resistance from junction to ambient	note 1	250	K/W

Note

1. Transistor mounted on an FR4 printed-circuit board.

CHARACTERISTICS

$T_{amb} = 25\ °C$.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
I_{CBO}	collector cut-off current	$I_E = 0$; $V_{CB} = 30$ V	–	50	nA
I_{EBO}	emitter cut-off current	$I_C = 0$; $V_{EB} = 6$ V	–	50	nA
h_{FE}	DC current gain	$V_{CE} = 1$ V; note 1			
		$I_C = 0.1$ mA	60	–	
		$I_C = 1$ mA	80	–	
		$I_C = 10$ mA	100	300	
		$I_C = 50$ mA	60	–	
		$I_C = 100$ mA	30	–	
V_{CEsat}	collector-emitter saturation voltage	$I_C = 10$ mA; $I_B = 1$ mA; note 1	–	200	mV
		$I_C = 50$ mA; $I_B = 5$ mA; note 1	–	200	mV
V_{BEsat}	base-emitter saturation voltage	$I_C = 10$ mA; $I_B = 1$ mA; note 1	–	850	mV
		$I_C = 50$ mA; $I_B = 5$ mA; note 1	–	950	mV
C_c	collector capacitance	$I_E = i_e = 0$; $V_{CB} = 5$ V; f = 1 MHz	–	4	pF
C_e	emitter capacitance	$I_C = i_c = 0$; $V_{EB} = 500$ mV; f = 1 MHz	–	8	pF
f_T	transition frequency	$I_C = 10$ mA; $V_{CE} = 20$ V; f = 100 MHz	300	–	MHz
F	noise figure	$I_C = 100\ \mu A$; $V_{CE} = 5$ V; $R_S = 1\ k\Omega$; f = 10 Hz to 15.7 kHz	–	5	dB
Switching times (between 10% and 90% levels); see Fig.2					
t_{on}	turn-on time	$I_{Con} = 10$ mA; $I_{Bon} = 1$ mA; $I_{Boff} = -1$ mA	–	65	ns
t_d	delay time		–	35	ns
t_r	rise time		–	35	ns
t_{off}	turn-off time		–	240	ns
t_s	storage time		–	200	ns
t_f	fall time		–	50	ns

Note

1. Pulse test: $t_p \leq 300\ \mu s$; $\delta \leq 0.02$.

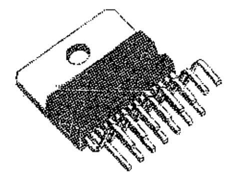
APPENDIX E

L298 H-BRIDGE

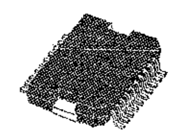
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)



Multiwatt15



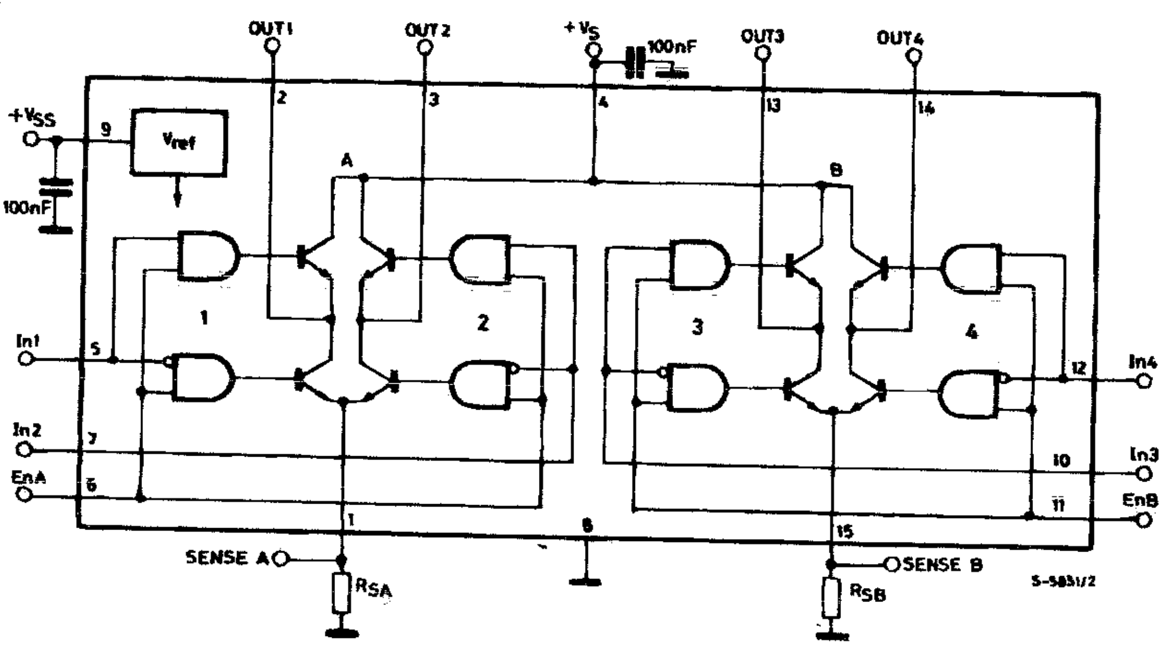
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

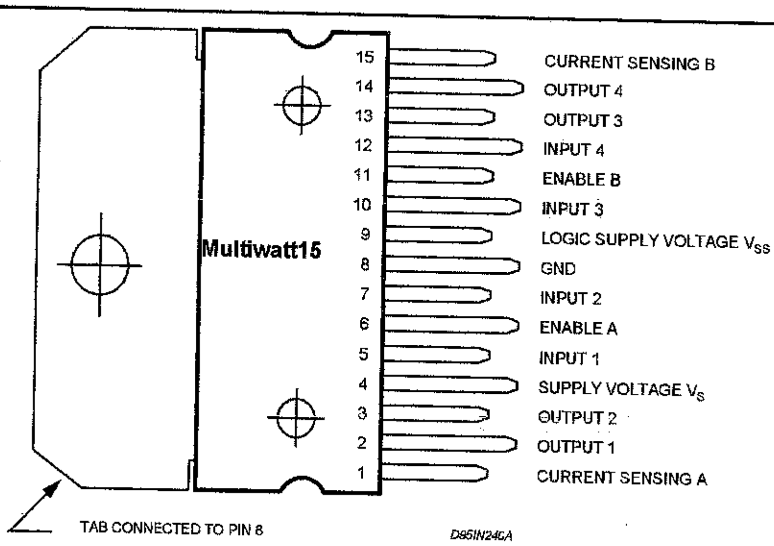
BLOCK DIAGRAM

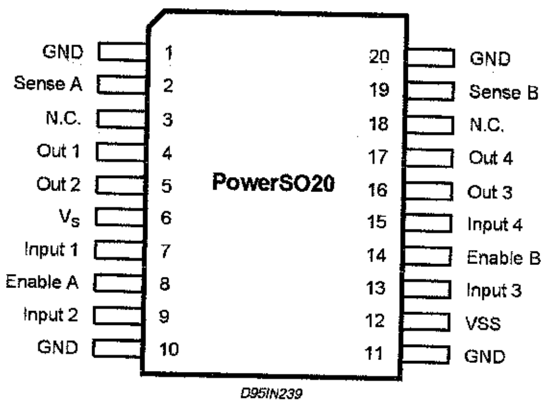


ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100μs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)





THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition	V_{IH} +2.5		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0 V_i = L V_i = H		13 50	22 70	mA mA
		V_{en} = L V_i = X			4	mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0 V_i = L V_i = H		24 7	36 12	mA mA
		V_{en} = L V_i = X			6	mA
V_{iL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V_{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		VSS	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_i = L			–10	µA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	Vi = H ≤ V_{SS} –0.6V		30	100	µA
V_{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L			–10	µA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} –0.6V		30	100	µA
$V_{CEsat\,(H)}$	Source Saturation Voltage	I_L = 1A I_L = 2A	0.95	1.35 2	1.7 2.7	V V
$V_{CEsat\,(L)}$	Sink Saturation Voltage	I_L = 1A (5) I_L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	I_L = 1A (5) I_L = 2A (5)	1.80		3.2 4.9	V V
V_{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

APPENDIX F

PROGRAMMING ALGORITHMS FOR CONDITIONAL MASTER AND CONDITIONAL SLAVES

```
//////////////////////////////////////////////////////////////////////////////
//     slave: 0x10 (slave 1 to control robot movement)                     //
//            0x20 (slave 2 to control robot mechanisms)                   //
//                                                                          //
//     input: B0 = robot forward                                            //
//            B1 = robot reverse                                            //
//            B2 = robot turn left                                          //
//            B3 = robot turn right                                         //
//            B4 = mechanism up                                             //
//            B5 = mechanism down                                           //
//            B6 = mechanism grip                                           //
//            B7 = mechanism release                                        //
//                                                                          //
//     I2C pins = A0 (SDA)                                                  //
//                A1 (SCL)                                                  //
//                                                                          //
//////////////////////////////////////////////////////////////////////////////


#include <16F84A.h>
#fuses XT,NOWDT,NOPROTECT,NOPUT
#use delay(clock=4000000)

#use I2C(MASTER,SDA=PIN_A0,SCL=PIN_A1, slow)

#define slave_address_1 0x10

#define slave_address_2 0x20

int   keypress;

void hantar(int slave_address,int command);

void main()
{
        output_high(PIN_A2);

        while(1)
        {

         keypress = input_b();

         if(keypress==0b00000001)
         {
             hantar(slave_address_1, 0x00);


         }
         else if(keypress==0b00000010)
         {
             hantar(slave_address_1, 0x01);


         }
         else if(keypress==0b00000100)
         {
             hantar(slave_address_1, 0x02);


         }
         else if(keypress==0b00001000)
         {
```

```
                                              master
        hantar(slave_address_1, 0x03);

    }
    else if(keypress==0b00010000)
    {
        hantar(slave_address_2, 0x00);

    }
    else if(keypress==0b00100000)
    {
        hantar(slave_address_2, 0x01);

    }
    else if(keypress==0b01000000)
    {
        hantar(slave_address_2, 0x02);

    }
    else if(keypress==0b10000000)
    {
        hantar(slave_address_2, 0x03);

    }
    else
    {
    }
  }

'oid hantar(int slave_address,int command)

  i2c_start();
  delay_ms(1);
  i2c_write(slave_address);
  delay_ms(1);
  i2c_write(command);
  delay_ms(1);
  i2c_stop();
  delay_ms(1);
```

```
////////////////////////////////////////////////////////////////////////////
//  Program: Slave 1 (0x10) for robot movement control                    //
//                                                                        //
//  i2c Pins: pin_c4 (SDA), pin_c3 (SCL)                                  //
//                                                                        //
//  output: B4 = right motor (terminal 1)                                 //
//          B5 = right motor (terminal 2)                                 //
//          B6 = left motor (terminal 1)                                  //
//          B7 = left motor (terminal 2)                                  //
//                                                                        //
////////////////////////////////////////////////////////////////////////////


#include <16F877a.h>
#fuses XT,NOWDT,NOPROTECT,NOPUT
#use delay(clock=4000000)

#use I2C(SLAVE,SDA=PIN_C4,SCL=PIN_C3,ADDRESS=0x10,FORCE_HW)

#define slave_address 0x10

#INT_SSP

void robot_forward();
void robot_reverse();
void robot_turn_left();
void robot_turn_right();
void robot_stop();

void ssp_interrupt()
{
        setup_ccp1(CCP_PWM);
        setup_ccp2(CCP_PWM);
        setup_timer_2(T2_DIV_BY_4, 127, 1);

        int rcvd_address,rcvd_data;

        rcvd_address = i2c_read();

        rcvd_data = i2c_read();

        if(rcvd_address==slave_address && rcvd_data==0x00)
        {
                robot_forward();
        }
        else if(rcvd_address==slave_address && rcvd_data==0x01)
        {
           robot_reverse();
        }
        else if(rcvd_address==slave_address && rcvd_data==0x02)
        {
           robot_turn_left();
        }
        else if(rcvd_address==slave_address && rcvd_data==0x03)
        {
                robot_turn_right();
        }
   else
   {
      robot_stop();
   }
}

void main()
```

```
        //enable interrupts
        enable_interrupts(GLOBAL);

        enable_interrupts(INT_SSP);

        //loop until interrupts take place
        while(1)
        {
                output_high(PIN_B7);
        }

}

void robot_forward()
{
        output_high(PIN_B6);
        output_low(PIN_B7);
        set_pwm1_duty(255);

        output_high(PIN_B4);
        output_low(PIN_B5);
        set_pwm2_duty(255);
}

void robot_reverse()
{
        output_low(PIN_B6);
        output_high(PIN_B7);
        set_pwm1_duty(255);

        output_low(PIN_B4);
        output_high(PIN_B5);
        set_pwm2_duty(255);
}

void robot_turn_left()
{
        output_low(PIN_B6);
        output_high(PIN_B7);
        set_pwm1_duty(200);

        output_high(PIN_B4);
        output_low(PIN_B5);
        set_pwm2_duty(200);
}

void robot_turn_right()
{
        output_high(PIN_B6);
        output_low(PIN_B7);
        set_pwm1_duty(200);

        output_low(PIN_B4);
        output_high(PIN_B5);
        set_pwm2_duty(200);
}

void robot_stop()
{
        output_low(PIN_B6);
        output_low(PIN_B7);
        set_pwm1_duty(0);

        output_low(PIN_B4);
        output_low(PIN_B5);
        set_pwm2_duty(0);
}
```

```
/////////////////////////////////////////////////////////////////////////
//  Program: Slave 2 (0x20) for robot's mechanism control              //
//                                                                     //
//  i2c Pins: pin_c4 (SDA), pin_c3 (SCL)                               //
//                                                                     //
//  output: pin_b1 = mechanism for up/down (terminal 1)                //
//          pin_b2 = mechanism for up/down (terminal 2)                //
//          pin_b4 = mechanism for grip/release (terminal 1)           //
//          pin_b5 = mechanism for grip/release (terminal 2)           //
/////////////////////////////////////////////////////////////////////////

#include <16F877a.h>
#fuses XT,NOWDT,NOPROTECT,NOPUT
#use delay(clock=4000000)

#use I2C(SLAVE,SDA=PIN_C4,SCL=PIN_C3,ADDRESS=0x20,FORCE_HW)

#define slave_address 0x20

#INT_SSP

void mechanism_up();
void mechanism_down();
void mechanism_grip();
void mechanism_release();
void mechanism_stop();

void ssp_interrupt()
{
        int rcvd_address,rcvd_data;

        rcvd_address = i2c_read();

        rcvd_data = i2c_read();

        if(rcvd_address==slave_address && rcvd_data==0x00)
        {
                mechanism_up();
        }
        else if(rcvd_address==slave_address && rcvd_data==0x01)
        {
                mechanism_down();
        }
        else if(rcvd_address==slave_address && rcvd_data==0x02)
        {
                mechanism_grip();
        }
        else if(rcvd_address==slave_address && rcvd_data==0x03)
        {
                mechanism_release();
        }
        else
        {
                mechanism_stop();
        }

oid main()

        //enable interrupts
        enable_interrupts(GLOBAL);
```

```
        enable_interrupts(INT_SSP);

        //loop until interrupts take place
        while(1)
        {
                output_high(PIN_B7);
                output_low(PIN_B1);
                output_low(PIN_B2);
                output_low(PIN_B4);
                output_low(PIN_B5);
        }

}

void mechanism_up()
{
        output_high(PIN_B1);
        output_low(PIN_B2);

        output_low(PIN_B4);
        output_low(PIN_B5);
}

void mechanism_down()
{
        output_low(PIN_B1);
        output_high(PIN_B2);

        output_low(PIN_B4);
        output_low(PIN_B5);
}

void meshanism_grip()
{
        output_low(PIN_B1);
        output_low(PIN_B2);

        output_high(PIN_B4);
        output_low(PIN_B5);
}

void mechanism_release()
{
        output_low(PIN_B1);
        output_low(PIN_B2);

        output_low(PIN_B4);
        output_high(PIN_B5);
}

void mechanism_stop()
{
        output_low(PIN_B1);
        output_low(PIN_B2);

        output_low(PIN_B4);
        output_low(PIN_B5);
}
```

APPENDIX G

PROGRAMMING ALGORITHMS FOR REAL-TIME MASTER AND CONDITIONAL SLAVES

```
#include <16F84a.h>
#fuses XT,NOWDT,NOPROTECT,NOPUT
#use delay(clock=4000000)

#use I2C(MASTER,SDA=PIN_A0,SCL=PIN_A1,slow)

int data;

void main()
{
   while(1)
   {
   int data;

   port_b_pullups(TRUE);
   data = input_b();

   i2c_start();
   delay_ms(1);
   i2c_write(0x10);
   delay_ms(1);
   i2c_write(data);
   delay_ms(1);

   i2c_start();
   delay_ms(1);
   i2c_write(0x20);
   delay_ms(1);
   i2c_write(data);
   delay_ms(1);
   i2c_stop();
   delay_ms(1);
   }
}
```

```
#include <16F877a.h>
#fuses XT,NOWDT,NOPROTECT,NOPUT,NOLVP
#use delay(clock=4000000)

#use I2C(SLAVE,SDA=PIN_C4,SCL=PIN_C3, ADDRESS=0x10, FORCE_HW)

#define slave_address 0x10

#INT_SSP

void ssp_interrupt()
{
        int i=0;
        setup_ccp1(CCP_PWM);
        setup_ccp2(CCP_PWM);
        setup_timer_2(T2_DIV_BY_4, 127, 1);


        int rcvd_address, datain;

        rcvd_address = i2c_read();

   if (rcvd_address==slave_address)
   {
        datain = i2c_read();


                if (datain==0b00000001)
                {
                        motor_forward();
                }
                else if (datain==0b00000010)
                {
                        motor_reverse();
                }
                else if (datain==0b00000100)
                {
                        motor_left();
                }
                else if (datain==0b00001000)
                {
                        motor_right();
                }
                else
                {
                        motor_stop();
                }
  }
  else;

oid main()


       //enable interrupts
       enable_interrupts(GLOBAL);

  enable_interrupts(INT_SSP);

       //loop until interrupts take place
       while(1)
       {
               output_high(PIN_D2);
       }
```

```
}

void motor_forward()
{
        output_high(PIN_B0);
        output_low(PIN_B1);
        set_pwm1_duty(255);

        output_high(PIN_B2);
        output_low(PIN_B3);
        set_pwm2_duty(255);
}

void motor_reverse()
{
        output_low(PIN_B0);
        output_high(PIN_B1);
        set_pwm1_duty(255);

        output_low(PIN_B2);
        output_high(PIN_B3);
        set_pwm2_duty(255);
}

void motor_left()
{
        output_low(PIN_B0);
        output_high(PIN_B1);
        set_pwm1_duty(200);

        output_high(PIN_B2);
        output_low(PIN_B3);
        set_pwm2_duty(200);
}

void motor_right()
{
        output_high(PIN_B0);
        output_low(PIN_B1);
        set_pwm1_duty(200);

        output_low(PIN_B2);
        output_high(PIN_B3);
        set_pwm2_duty(200);
}

void motor_stop()
{
        output_low(PIN_B0);
        output_low(PIN_B1);
        set_pwm1_duty(0);

        output_low(PIN_B2);
        output_low(PIN_B3);
        set_pwm2_duty(0);
}
```

slave2[actvte at same time]

```
#include <16F877a.h>
#fuses XT,NOWDT,NOPROTECT,NOPUT,NOLVP
#use delay(clock=4000000)

#use I2C(SLAVE,SDA=PIN_C4,SCL=PIN_C3, ADDRESS=0x20, FORCE_HW)

#define slave_address 0x20

#INT_SSP

void ssp_interrupt()
{
        int rcvd_address, datain;

        rcvd_address = i2c_read();

   if (rcvd_address==slave_address)
   {
        datain = i2c_read();


                if (datain==0b00010000)
                {
                        mechanism_up();
                }
                else if (datain==0b00100000)
                {
                        mechanism_down();
                }
                else if (datain==0b01000000)
                {
                        mechanism_grip();
                }
                else if (datain==0b10000000)
                {
                        mechanism_release();
                }
                else
                {
                        mechanism_stop();
                }
   }
   else;
}

void main()
{

        //enable interrupts
        enable_interrupts(GLOBAL);

   enable_interrupts(INT_SSP);

        //loop until interrupts take place
        while(1)
        {
                output_high(PIN_D2);
        }


void mechanism_up()

        output_high(PIN_B0);
```

```
                                  slave2[actvte at same time]
        output_low(PIN_B1);

        output_low(PIN_B2);
        output_low(PIN_B3);
}

void mechanism_down()
{
        output_low(PIN_B0);
        output_high(PIN_B1);

        output_low(PIN_B2);
        output_low(PIN_B3);
}

void mechanism_grip()
{
        output_low(PIN_B0);
        output_low(PIN_B1);

        output_high(PIN_B2);
        output_low(PIN_B3);
}

void mechanism_release()
{
        output_low(PIN_B0);
        output_low(PIN_B1);

        output_low(PIN_B2);
        output_high(PIN_B3);
}

void mechanism_stop()
{
        output_low(PIN_B0);
        output_low(PIN_B1);

        output_low(PIN_B2);
        output_low(PIN_B3);
}
```

APPENDIX H

PROJECT'S GANTT CHART

No	Work Description	Week																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	Developing Programming Algorithm																	
	- Master PIC																	
	- Slave PIC																	
	- Test and Troubleshoot																	
2	Desigining the electronic circuitry																	
	- Remote circuitry																	
	- Robot circuitry																	
	- Test and Troubleshoot																	
3	Construction of Robot's Base																	
	- Design structure and mechanisms																	
	- Identify materials to be used																	
	- Construct the base																	
	- Install mechanisms																	
4	Integrate Structure with Circuitries																	
	- Test and Troubleshoot																	
5	Reports Submission																	
	- Weekly Logbook																	
	- Progress Report 1																	
	- Progress Report 2																	
	- Draft Report																	
	- Final Report (soft cover)																	
	- Technical Report																	
	- Final Report (hard cover)																	
6	Oral Presentation																	