INTELLIGENT MOBILE ROBOT

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF
BACHELOR OF ENGINEERING –
COMPUTER SCIENCE AND ENGINEERING
BHARATHIAR UNIIVERSITY, COIMBATORE.

SUBMITTED BY

Mr. M.Karthikeyan

Mr. M.B.Karthikeyan

Ms. R.Nithya Nandhini

Ms. M.Sangeetha

Ms. Thangavelu, Subadhra

GUIDED BY

External Guidance
Mr.J.Chandran,
Jenesis Systems & Controls
Coimbatore.

Mr.M.A.Mahalingam,
Micro Magnetic Technologies,
Coimbatore.

Internal Guidance
Ms.S.Rajini, B.E.,
Senior Lecturer,
Dept of Computer Science,
Kumaraguru College of Technology,
Coimbatore.

Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore - 641 006

MARCH 2002.

CERTIFICATE

Department of Computer Science and Engineering

Kumaraguru College of Technology

Coimbatore - 641006.

This is to certify that the project work entitled

INTELLIGENT MOBILE ROBOT

Done by

Mr. M.Karthikeyan (9827k0180)
Mr. M.B.Karthikeyan (9827k0181)
Ms. R.Nithya Nandhini (9827k0191)
Ms. M.Sangeetha(9827k0210)
Ms. Thangavelu, Subadhra(9827k0220)

In partial fulfillment for the award of the degree of
Bachelor of Engineering in Computer Science and Engineering
of Bharathiar University, Coimbatore
during the academic year 2001 - 2002

S. Thangasamy 18/3/02
Professor and Head
Dr. S.Thangasamy, B.E, Ph.D.,

Rajini
Guide
Ms. S.Rajini, B.E.,

Certified that the candidate was examined by us in the Project Work Viva Voce Examination held on ....15/3..... and the University Register Number was .....

S. Murugan 15/3
Internal Examiner

External Examiner

ACKNOWLEDGEMENT

First and foremost, we thank the Almighty for His grace to take up this project. We thank our Principal Dr.K.K.Padmanabhan (B.Sc.)Engg., M.Tech, and Ph.D. for providing us with excellent facilities for doing the project work.

We owe our sincere thanks to the Head of our Department Prof.K.Thangasamy, B.E(Hons),Ph.D, for providing us with lab facilities and material.

Ms.S.Rajini, B.E, MISTE, our project guide has ushered us throughout our work and we thank her for inciting us with expeditious motifs and efficiently helping us in the design and implementation of our project work.

Kudos to Mr.J.Chandran, Chief Technical Advisor of Jenesis Systems and Controls, for steering us with a tactical maneuver till the end of our fabrication work. He has been a lodestar by providing us with laudable advice and efficient guidance at the time of need and of course, his matchless dedication, determination and confidence in us has led us all the way towards achieving the target.

We would like to express our sincere gratitude to the backbones of our project - Mr.M.A.Mahalingam, Micro Magnetic Technologies, CBE, Mr.J.PremKumar, M.D, Jenesis Systems and Controls, Coimbatore, Mr.K.BalaSubramanian,M.D, Precision Equipments Pvt. Ltd.,Chennai, Mr. Raghunathan,Hydroquip Hydraulics Pvt. Ltd.,Chennai, for helping us with everything they can from behind the scenes.

We take pride in saying that our unquenched spirit and search has

cannot forget to express our gratitude to our friends and staff members of the Department of Computer Science and Engineering, Department of Electronics and Communication, Department of Electrical and Electronics who have provided us incentives and helped us directly and indirectly to complete our project work.

SYNOPSIS

Robotic Engineering is emerging as one of the most important areas of Technology of this millennium. A project in robotics needs a sound knowledge in various field of engineering like Mechanical, Electronics, Electrical components, Sensor technology and Computer systems. The future of Robotics which is dependent on the sensor technology makes it to be the cutting edge in technology. The mobility factor also facilitates the robot to be used in a range of fields.

In our project, we have proposed a robot that can pick up an object from the source and place it in the destination in a given work area. On its path from the source to the destination, any obstacle present will be detected by the Optical Proximity sensors. Thus, it bypasses the obstacle and reaches the destination.

The robot in its work area is interfaced to the computer. The programming part of our project is done entirely in C language. The wheel rollers and a pick and place mechanism with an electromagnet controlled by the DC Stepper motors are used in the robot. The robot can thus be used in industrial areas as an automated guided vehicle to lift the objects and reach the target by itself, as programmed.

Our project, a small shell in the ocean of Robotics is an epitome of integration of various branches of Engineering. Aided by the Department Of Technical Education, it has led us to take initiative to explore the petty details of Electronics and Mechanics.

CONTENTS

INTRODUCTION

When we talk about ***ROBOTS****, many people will think of those machines in the science fiction films. They have man-like shape, behave like a human being and speak understandable languages.*

The word robot was first devised by Czech author ***Karel Capek*** *and simply meant work or servitude. In manufacturing industries, robots are used to perform repeated and tedious jobs. They work with high accuracy and without complaints. In exploration, robots are sent to hazardous areas, to deep sea, or even other planets to perform jobs, which no human could easily do.*

What is a Robot?

- *A Robot is a re-programmable, multifunctional manipulator designed to move materials, parts, tools or special devices through variable programmed motions for the performance of a variety of tasks.*
- *Robots can do things we humans just don't want to do, and usually do it cheaper.*
- *Robots can do things more precise than humans and allow progress in all scientific fields*

What are Robots made of?

Robots have 3 main components:

- *Brain –usually a computer.*
- *Actuator and mechanical parts – motors, pistons, grippers, wheels, gears*

➢ *Sensors – vision, sound, temperature, motion, light, touch, etc.*

With these three components, robots can interact and affect their environment to become useful.

Laws of Robotics:

Popular science fiction writer ***Isaac Asimov*** *created the Three Laws of Robotics:*

Law 1*: A Robot must not injure a human being or, through inaction, allow a human being to come to harm.*

Law 2: *A Robot must always obey the orders given to it by a human being, except where it would conflict with the first law.*

Law 3: *A Robot must protect it's own existence, except for cases in which it would conflict with the first or second law.*

Later, Asimov added this "Zeroth Law"

Law 0: *A Robot must not injure humanity or, through inaction, allow humanity to come to harm.*

Drive mechanism:

Commercially available industrial robots are powered but one of the three types of drive systems.

(a) ***Hydraulic and Pneumatic****: They are associated with large models with high speed and strength. The disadvantages are typical requirement of large floor space and systems, inclination to nil leakage. These drives are fluid driven power to operate their manipulators, where the hydraulic drive uses fluids and pneumatic drive uses gases. Moreover, pneumatic drives are reserved for robots that possess fewer degrees of freedom.*

(b) ***Electric:*** *These are not only cheaper but also accurate and highly repeatable too. Electrically driven robots may be actuated by D.C. Stepping motors or D.C. Servo motors*

Classification of robots

The robots can be classified according to the types of control, capabilities, configuration, and mobility.

Based on types of control:

- *Point-to-point robots move from one point to another but cannot stop at arbitrary intermediate points.*
- *Continuous point robots can move to a prescribed number of points along a path and can stop at arbitrary intermediate points.*
- *Computed trajectory robots can move along a path specified algebraically.*
- *Servo-controlled robots have some means of sensing the current position and feeding this sensed position back so that they can follow a prescribed path.*

Based on capability:

- *Sequence-controlled robots are machines, which go through a fixed sequence of actions according to the instructions.*
- *Computer trajectory robots follow a specified path between the starting and finishing points. Some numerically controlled machines are of this type.*

- *Adaptive robots can react to their environment using their sensors. The performance is optimized by adjusting controls to changing parameters.*
- *Intelligent robots are provided with sensors to study and model the environment, thereby generating a knowledge base. They perform with the help of an expert system provided along with the robot. The knowledge base is constantly updated so as to better the performance with the passage of time.*

Based on configuration:

- *A robot may articulate its joints in the Cartesian co-ordinate system, in which case, it is known as Cartesian robot.*
- *A Cylindrical robot works with the cylindrical co-ordinate system.*

Based on mobility:

- *Robots may be fixed in the workspace or mounted on a short track. These are called as robots of Fixed type.*
- *Mobile robots are mounted on wheels, powered by batteries or power supply and guided by non-contact type tracks. Sometimes, they are guided by personnel using a Tele-operator.*
- *Walking robots have legs, which can move about, in difficult terrain. Some have the ability to climb up and down the slopes and staircases. These robots are powered by the batteries or engines, and are operated remotely via a radio link by the computer.*

System Requirements

PRODUCT DEFINITION

In this section, the requirements of the robot pertaining to its functions are defined. The main functions of the robot are

- *Mobile*
- *Intelligent*
- *Pick and place mechanism*

Mobile

The robot is designed to be mobile in nature. The mobility is implemented by means of wheels. There are 2 wheels in the front of the robot for direction and 2 wheels at the back for driving. Two stepper motors, one for each of the above functions have been connected to the wheels. The motor used for driving is of higher torque specification due to its necessity to bear the weight of the robot.

Intelligent

Mobile robot is intelligent by its capacity to sense obstacles. Sensing is implemented by optical proximity sensors. The sensor is placed in front of the robot so that the robot can respond to any object that comes in the path of the robot.

Pick and Place

Pick and place operation is the prime function of the intelligent mobile robot. The robot is designed to pick only small iron particles. An electromagnet connected to a pulley arrangement is driven by a stepper motor. This electromagnet is made to pick the object in the source and place it in the destination by appropriate control from the computer.

PROJECT PLAN

The success of a project depends on channelising the plans towards target in tandem with the well-organized utilization of the pre-determined time for the various tasks. Thus, we initially set off with our project plan.

The key assignment of the project lay in the determining the tasks and the order in which they have to be sequenced.

Task 1: Data collection

The prime stage of the project is data collection. Data has to be collected from the various fields of engineering relating to the project design, fabrication, and documentation. Though it is stated as first, it is our task necessary to be carried out throughout the project.

Task 2: Data Analysis

The collected data is analyzed illuminating the functions of the robot complying with the aim of the project. The pros and cons of the various methodologies are explored. A time span of 2 weeks is placed as our target for initial data collection and its analysis.

Task 3: Elements selection

The elements of the project have to be selected taking into consideration all of the following factors:

- *Fitness for the purpose*
- *Cost*
- *Feasibility for implementation*

Task 4: Design

The design phase becomes easier after selecting the optimal elements. The fabrication and interfacing design are decided with one main constraint: Compactness. The selection and design are given an optimum time of 2-3 weeks.

Task 5: Fabrication and interface circuits

The robot fabrication along with its circuits has to be completed in accordance with the respective design. This critical stage of the project will take a span of 3-4 weeks.

Task 6: Coding

The brain part has to be done from the scratch of the project. The basic modules can be completed and tested to fit into the main module after fabrication is done. The coding will continue till the end of our project.

Task 7: Testing

The testing phase is not a detached phase. As and when modules or components are done, testing is conducted to ensure the proper and intended working of the robot.

By carrying out the project plan in a meticulous manner, our intelligent mobile robot will be completed in the stipulated time.

Design Document

STEPPER MOTOR

The Stepper motor is often, considered as a digital device which converts electric pulses into proportionate mechanical movement.

The stepper motor has been used for robot control. These motors are selected keeping the consideration of cost and accuracy of control. The motors work in steps of 1.8 degrees, which makes them easier to control and hence gives way for accurate positioning.

FEATURES OF STEPPER MOTORS

Small step angles

A stepper motor rotates through a fixed angle for each pulse. The rated value of this angle is called the step angle and is expressed in degrees. The smaller the step angle, the higher the resolution of positioning can be. One feature of stepper motor is that they can be made to realize a small step angle. Engineers are interested in the number of steps in the revolution, which is conventionally the step number. The relation between the step angle θS and the step number S is

$$S=360/\theta S$$

A standard four-phase motor has a step number of 200. Some precision motors are designed to attain one revolution with 500 or 1000 steps. However, the step angles in some simple motors are large as 90°, 45°,or 15°.

High positioning accuracy

Accuracy in positioning is important factor, which determines the quality of a stepper motor. Stepper motors are designed so that they rotate through a predetermined step angle in response to a pulse signal and come to rest at a precise position.

Since the accuracy at no load depends on the physical accuracy of the rotor and stator, stepping motors are manufactured very carefully. Moreover, stepping motors are designed so that a highly restoring torque is produced when a displacement from a rest position occurs due to load torque. The air gap between the rotor and stator teeth is designed to be as small as possible to this end. Thus the positioning accuracy depends only on the machine characteristics and the driving circuit, while other electronic parameters have no effect on positioning accuracy.

Let us here consider some of the terminology involved in discussing the maximum static torque.

Maximum static torque

- ***Holding torque:*** *It is defined as the maximum static torque that can be applied to the shaft of an excited motor without causing continuous rotation.*
- ***Detent torque****: It is defined as the maximum static torque that can be applied to the shaft of an unexcited motor without causing continuous rotation. In general the larger the holding torque, the smaller the position error due to load. The detent torque appears only in motors having a permanent magnet.*

Positions at which the rotor stops moving

- ***Rest position:*** *It is also known as equilibrium position. It is defined as the positions at which an excited motor comes to rest at no-load.*
- ***Detents position:*** *It is defined as the position at which a motor having a permanent magnet in its rotor or stator comes to rest without excitation at no-load.*

Positioning accuracy

- ***Step position error:*** *It is defined as the largest positive or negative static angular position error (compared with the rated step angle), which can occur when the rotor moves from one rest position to the next.*
- ***Positional accuracy:*** *It is defined as the largest angular position error of a rest position related to the whole multiple of the rated step angle, which can occur during a full revolution of the rotor when moving from a reference rest position.*
- ***High torque-to inertia ratio****: It is desirable that a stepping motor moves as fast as possible in response to an input pulse or pulse train. No only a quick start but also a quick stop is required for a stepping motor. If the pulse train is interrupted while the motor is running at a uniform speed, the motor should be capable of stopping at the position specified by the last pulse. The above indicates that the ratio of the torque to rotor inertia must be large in stepping motors as compared with conventional electrical motors.*
- ***Stepping rate and pulse frequency:*** *The speed of rotation of a stepping motor is given in terms of the number of steps per*

second and the term 'stepping rate' is often used to indicate speed.

- *Since, in most stepping motors, the number of pulses applied to the logic sequencer equals the number of steps, the speed may be expressed in terms of pulse frequency. Hertz (Hz) is the input of stepping rate.*
- *It should be noted, however, that the stepping rate does not specify the absolute speed. This speed is termed 'rotational speed' for conventional electrical machines and expressed in terms of revolutions per minute. The relation between the rotational speed and the stepping rate is given by*

$$N=60\,f/S$$

Where, N *– rotational speed (r.p.m)*

f *–stepping rate (Hz)*

S *–step number*

PARAMETERS OF STEPPER MOTORS

Certain parameters of stepper motors are defined below.

- ***Step angle:*** *This refers to the angle through which the motor moves for one step.*
- ***Stepping rate****: The number of steps executed per second is known as the stepping rate. If the stepping pulses were given at a rate greater than the maximum allowable rate, then the motor would miss many steps.*
- ***Holding torque:*** *This is equal to the external torque which must be applied to break away from its equilibrium position, when a stator winding remains excited.*

- ***Dynamic torque:*** *The torque produced by the stepper motor under a constant stepping rate is known as the dynamic torque.*
- ***Detent torque:*** *This is the torque required to overcome the residual magnetism and reluctance torque under the unexcited condition. This torque enables the motor to hold a load even when the stator windings are de-energized.*
- ***Holding torque/inertia ratio:*** *High performance motors have a large holding torque/inertia ratio. Such motors have a length greater than the diameter.*
- ***Step response:*** *When the motor executes a single step, the rotor exhibits a decaying oscillatory response.*
- ***Resonance****: The inertia of the motor together with the magnetic stiffness forms an oscillatory system. At a certain stepping rates, this phenomenon causes the rotor to oscillate and lose the step integrity. Addition of inertia and viscous damping would improve the stability. Special inertia cum viscous dampers called lanchester dampers are commonly employed. In the unipolar mode, Darlington transistor pairs drive the windings.*

TYPES OF STEPPER MOTORS:

There are 3 main types of stepper motor:

1. *Variable reluctance stepper motors*
2. *Permanent magnet stepper motors*
3. *Hybrid stepper motors*

We have used variable reluctance stepper motor which is described below:

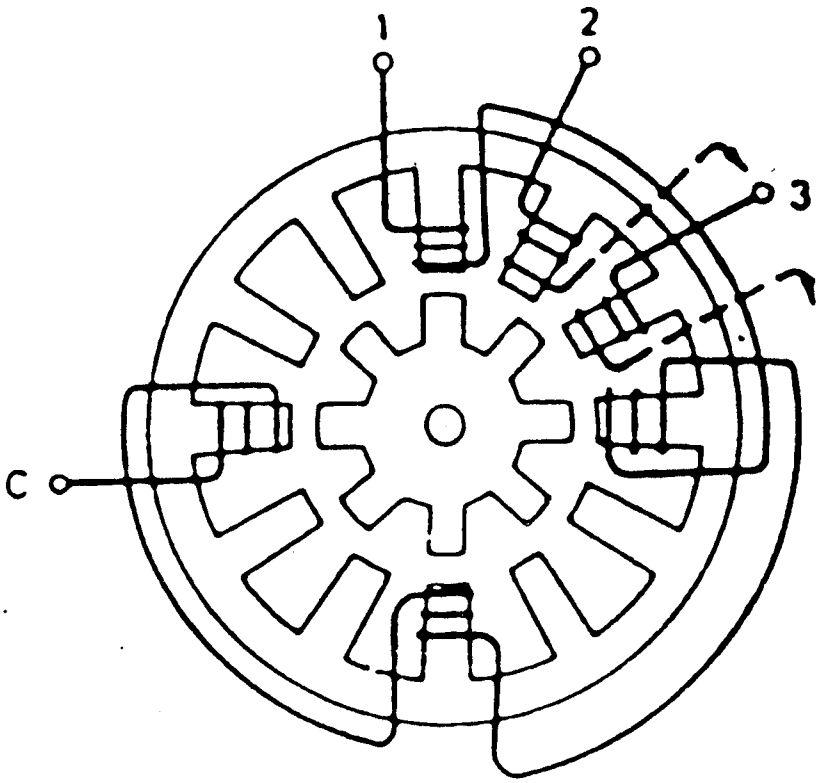
Variable reluctance stepper motors

VR stepper motor may be considered to be the most basic type of stepper motor. This is a three-phase motor having six stator teeth. Any 2 opposing stator teeth which are at 180° from each other belong to the same phase; i.e., coils on each opposing tooth are connected in series or parallel. The rotor has 4 teeth. The stator and the rotor core are normally made of laminated silicon steel, but solid silicon steel rotors are extensively employed. Both the stator and the rotor materials must have high permeability and be capable of allowing a high magnetic flux to pass through them even a low magneto motive force is applied. Current to each phase is controlled in the on/off mode by the respective switches. If a current is applied to the coils of ph1 or in other words if ph1 is excited, the magnetic flux will occur. The rotor will then be positioned so that the stator teeth 1 and 1' and any two of the rotor teeth is aligned.

Thus when the rotor teeth and the stator teeth are in alignment, the magnetic reluctance is minimized, and this state provides a rest or equilibrium position due to some external torque applied to the rotor shaft, a restoring torque will be generated. This will result in the curving of magnetic flux line at the edges of the teeth of both the stator and rotor. Magnetic lines of intensity have a strong tension, or in other words, magnetic lines have a tendency to become as short and straight as possible. This is known as the Maxwell stress.

When the rotor and stator teeth are out of alignment in the excited phase the magnetic reluctance is large. The VR motor works in such a way that the magnetic reluctance becomes minimum. This motion through a step angle at each switching of the excitation is called a step. After completing a rotor teeth pitch rotation in 3 steps

the rotor will apparently return to its original position. Air gap should be as small as possible. The air gap between the rotor and stator teeth must be minimum to produce high torque from a small volume and to attain accuracy in high positioning.



For smaller step angles: the possibility of realizing a small step angle is one of the unique features of the stepper motors. The relationship between step angle θs, number of phases m, rotor teeth, Nr, and step number S is given by,

$$S = 360/\theta s = mNr$$

In order to reduce the step angle, the number r or the rotor teeth Nr should be increased. It can be taken from the above explanation that the number of stator teeth can be increased as well as the rotor teeth. But, the number of stator teeth is not specified in the above equation.

The large salient position around which windings are located conventionally are called poles. A pole has 2 or more stator teeth, and all the teeth in a pole have the same magnetic polarity at any time.

ANTICLOCKWISE						CLOCKWISE					
STEP	A1	A2	B1	B2	HEX	STEP	A1	A2	B1	B2	HEX
1	0	1	1	0	06	1	0	1	0	1	05
2	1	0	1	0	0A	2	1	0	0	1	09
3	1	0	0	1	09	3	1	0	1	0	0A
4	0	1	0	1	05	4	0	1	1	0	06

A total of twelve steps are required to move the rotor by 360 degrees(mechanical). The switching sequence is given in the table above.

The main advantages of stepper motors are that they:

- *Are directly compatible with digital control techniques, the only interface they need is a power switch.*
- *Can be driven open loop with good position and speed control.*
- *Have no accumulative positional error or drift.*
- *Provide bi-directional rotation with no additional complexity of the driver.*
- *Are mechanically simple, rugged and durable (no commutator, no brake, no clutch).*
- *Require little or no maintenance.*
- *Are reliable and can be repeatedly stalled without damage.*

Some disadvantages limit their application:

- *Low efficiency.*
- *Only a limited power output range is available (upwards), expecting for electro-hydraulic stepper motors.*
- *Fixed step angle with an ordinary driver.*
- *Relatively high overshoot and oscillation in step response.*
- *Friction loads increase position error, although it is non-cumulative.*

Listed below are various ways a stepper motor may be applied as:

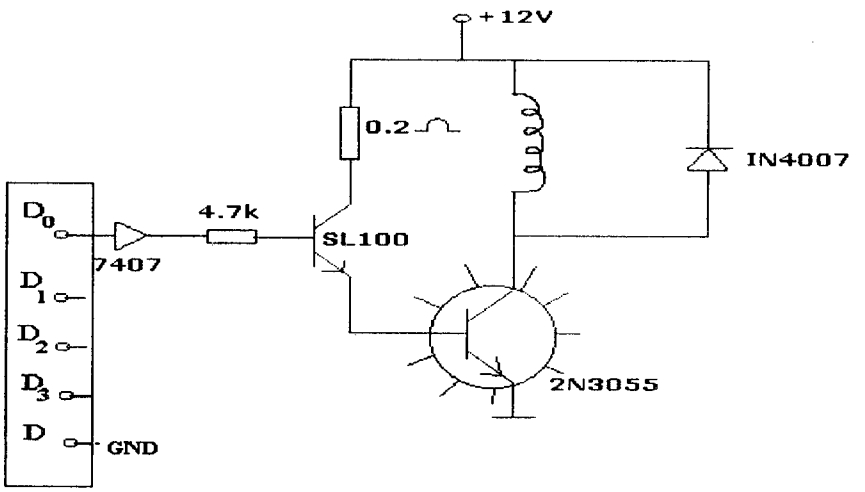
- *Synchronous motor.*
- *Variable speed motor.*
- *Open loop servo motors (no control feed back required).*

This very wide range of capabilities offers a broad field of applications. The following list represents some of the typical present day stepper motor applications:

- *Printers*
- *Punched-tape readers and punchers*
- *Disk memory drives*
- *X-Y plotters*
- *Incremental tape recorders*
- *Remote indicating devices*
- *Robots*

STEPPER MOTOR DRIVE CARD

In stepper motors as the input of rotor rotates in equal increments in response to a chain of input pulses it requires a drive card for driving the input pulses.



In this project, 3 stepper motors are employed. D0 to D3 lines of the parallel port are used for feeding the data to the stepper card. The lines are connected in parallel to each of the stepper card. The stepper card selection is done by means of a switching circuit driven by D4, D5 and D6 lines for each of the stepper motor respectively.

The buffer 7407 is used to isolate the system from adverse circuit effects. The bits are given as inputs to the base of a Darlington pair transistor. Since large current is needed to drive the stepper motor the Darlington transistor is used because of its high current gain. The stepper motor is connected at the collector terminal. The diode is connected in parallel with the stepper motor. When there is an input to the base of the transistor 1 the collector-base gets open-

circuited as a result of which current flows. through the windings of the stepper motor from the supply.

When transistor 1 is low, it is driven into saturation. All the current passes through transistor 1. The presence of diode 1N4007 across the winding is to avoid flyback. Flyback is the phenomenon, which occurs when high current levels are switched in inductors. The diode provides a path for the flyback current to drain.

PARALLEL PORT

A PC printer port is an inexpensive and yet powerful platform for implementing projects dealing with the control of real world peripherals. The printer port provides eight TTL outputs, five inputs and four bidirectional leads and it provides a very simple means to use the PC interrupt structure.

Port Assignments

Each printer port consists of three port addresses; data, status and control port. These addresses are in sequential order. That is, if the data port is at address 0x0378, the corresponding status port is at 0x0379 and the control port is at 0x037a.
The following is typical.

Printer	***Data Port***	***Status***	***Control***
LPT1	*0x03bc*	*0x03bd*	*0x03be*
LPT2	*0x0378*	*0x0379*	*0x037a*
LPT3	*0x0278*	*0x0279*	*0x027a*

To definitively identify the assignments for a particular machine, use the DOS debug program to display memory locations 0040:0008. For example:

```
>debug
-d 0040:0008 L8
0040:0008      78 03 78 02 00 00 00 00
```

Note in the example that LPT1 is at 0x0378, LPT2 at 0x0278 and LPT3 and LPT4 are not assigned. Thus, for this hypothetical machine;

Printer	***Data Port***	***Status***	***Control***
LPT1	*0x0378*	*0x0379*	*0x037a*
LPT2	*0x0278*	*0x0279*	*0x027a*
LPT3	*NONE*		
LPT4	*NONE*		

Port Outputs

Please refer to the figures titled Figure #1 - Pin Assignments and Figure #2 - Port Assignments. These two figures illustrate the pin assignments on the 25 pin connector and the bit assignments on the three ports.

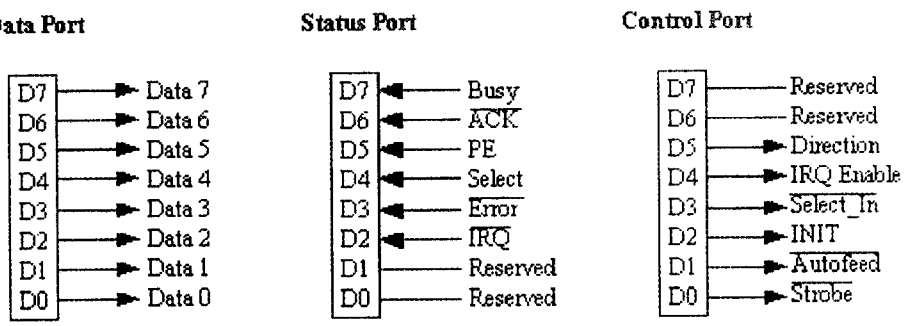
Fig 1. Pin Assignments

Signal Name	*Register Bit*	*DB-25 Pin*	*I/O Direction*
-Strobe	*¬C0*	*1*	*Output*
+Data Bit 0	*D0*	*2*	*Output*
+Data Bit 1	*D1*	*3*	*Output*
+Data Bit 2	*D2*	*4*	*Output*
+Data Bit 3	*D3*	*5*	*Output*
+Data Bit 4	*D4*	*6*	*Output*
+Data Bit 5	*D5*	*7*	*Output*
+Data Bit 6	*D6*	*8*	*Output*
+Data Bit 7	*D7*	*9*	*Output*

-Acknowledge	S6	10	Input
Busy	S7	11	Input
Paper End	S5	12	Input
Select In	S4	13	Input
-Auto Feed	C1	14	Output
-Error	S3	15	Input
-Initialize	C2	16	Output
-Select	C3	17	Output
Ground	-	18-25	-

(Note again that the S7, C0, C1 & C3 signals are inverted)

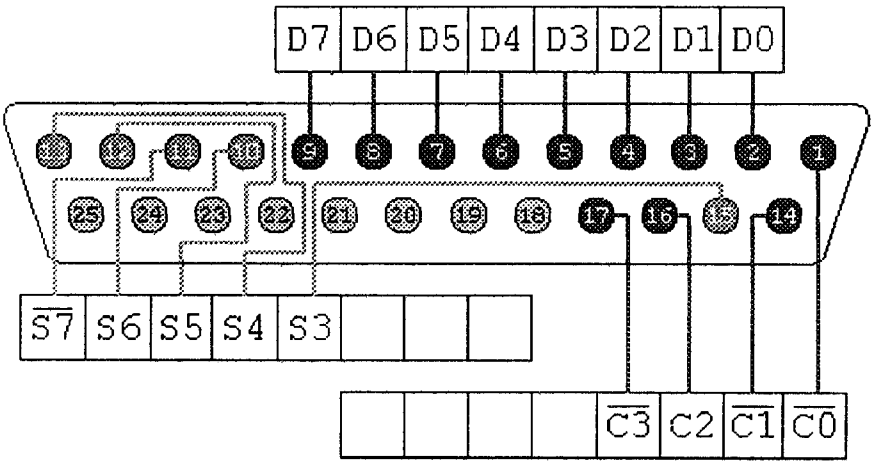
Fig 2. Port Assignments



Note that there are eight outputs on the Data Port (Data 7(msb) - Data 0) and four additional outputs on the low nibble of the Control Port. /SELECT_IN, INIT, /AUTO FEED and /STROBE.

All outputs on the Data Port are true logic. That is, writing a logic one to a bit causes the corresponding output to go high. However, the /SELECT_IN, /AUTOFEED and /STROBE outputs on the Control Port have inverted logic. That is, outputting a logic one to a bit causes a logic zero on the corresponding output. This adds some

complexity in using the printer port, but the fix is to simply invert those bits using the exclusive OR function prior to outputting.



Port Inputs

Note that in the diagram showing the Status Port there are five status leads from the printer. (BSY, /ACK, PE (paper empty), SELECT, /ERROR).

The following fragment illustrates the reading the five most significant bits in "true" logic.

```
#define DATA 0x03bc
#define STATUS DATA+1
...
unsigned int in_val;
...
in_val = ((inportb(STATUS)^0x80) >> 3);
```

Note that the Status Port is read and the most significant bit, corresponding to the BSY lead is inverted using the exclusive-or function. The result is then shifted such that the upper five bits are in the lower five bit positions.

0 0 0 BUSY /ACK PE SELECT /ERROR

Another input, IRQ on the Status Port is not brought to a terminal on the DB-25 printer port connector.

At this point, at a minimum, there are 12 outputs; eight on the Data Port and four on the lower nibble of the Control Port. There are five inputs, on the highest five bits of the Status Port. Three output bits on the Control Port and one input on the Status Port are inverted by the hardware, but this is easily handled by using the exclusive-or function to selectively invert bits.

Port Addressing

The printer adapter responds to five I/O instructions: two output and three input. The output instructions transfer data into two latches whose outputs are presented on the pins of a 25-pin D-type female connector.

Two of the three input instructions allow the processor to read back the contents of the two latches. The third allows the processor to read the realtime status of a group of pins on the connector.

A description of each instruction follows

Output to address 278/378/3BC Hex

Bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Pin	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*

The instruction captures data from the data bus and is present on the respective pins. These pins are each capable of sourcing 2.6 mA and sinking 24 mA. It is essential that the external device not try to pull these lines to ground.

Output to address 27A/37A/3BE Hex

Bit	*7*	*6*	*5*	*4*	*~3*	*2*	*~1*	*~0*
Pin	*-*	*-*	*-*	*IRQ enable*	*17*	*16*	*14*	*1*

This instruction causes the latch to capture the least significant bits of the data bus. The four least significant bits present their outputs, or inverted versions of their outputs, to the respective pins shown above. If bit 4 is written as 1, the card will interrupt the processor on the condition that pin 10 transitions high to low.

These pins are driven by open collector drivers pulled to +5 Vdc through 4.7 k-ohm resistors. They can each sink approximately 7 mA and maintain 0.8 volts down-level.

Input from address 278/378/3BC Hex

This command presents the processor with data present on the pins associated with the corresponding output address. This should normally reflect the exact value that was last written. If an external device should be driving data on these pins (in violation of usage ground rules) at the time of an input, this data will be ORed with the latch contents.

Input from address 279/379/3BD Hex

This command presents real time status to the processor from the pins as follows.

Bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Pin	*11*	*10*	*12*	*13*	*15*	-	-	-

Input from address 27A/37A/3BE Hex

This instruction causes the data present on pins 1, 14, 16, 17 and the IRQ bit to be read by the processor. In the absence of external drive applied to these pins, data read by the processor will exactly match data last written to the corresponding output address in the same bit positions. Note that data bits 0-2 are not included. If external drivers are dotted to these pins, that data will be ORed with data applied to the pins by the output latch.

Bit	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
Pin	-	-	-	*IRQ enable*	*~17*	*16*	*~14*	*~1*

SENSORS

Introduction

Sensors are used as peripheral devices in robotics include both simple types such as limit switches and sophisticated types such as machine vision systems. Of course, sensors are also used as integral components of the robot's position feedback control system. Their function as peripheral devices in a robotic work cell is to permit the robot activities to be coordinated with other activities in the cell. The sensors used in robotics include the following general categories:

1. ***Tactile sensors:*** *these are sensors, which respond to contact forces with another object. Some of these devices are capable of measuring the level of force involved.*
2. ***Proximity and range sensors:*** *a proximity sensor is a device that indicates when an object is close to another object but before contact has been made. When the distance between the objects can be sense, the device is called a range sensor.*
3. ***Miscellaneous types:*** *the miscellaneous category includes the remaining kinds of sensors that are used in robotics. These include sensors for temperature, pressure, and other variables.*
4. ***Machine vision:*** *a machine vision system is capable of viewing the workspace and interpreting what it sees. These systems are used in robotics to perform inspection, parts recognition and other similar tasks.*

Desirable features of sensors:

1. ***Accuracy:*** *accuracy is interpreted to mean that the true value of the variable can be sensed with no systematic positive or negative errors in the measurement.*
2. ***Precision:*** *precision means that there is little or no random variability in the measured variable.*
3. ***Operating range:*** *the sensor should possess a wide operating range and should be accurate and precise over the entire range.*
4. ***Speed of response:*** *the transducer should be capable of responding to changes in the sensed variable in a minimum time. Ideally, the response would be instantaneous.*
5. ***Reliability:*** *it should not be subject to frequent failures during operation.*
6. ***Cost and ease of operation:*** *the cost to produce, install and operate the sensor should be as low as possible. Further the ideal circumstance would be that the installation and operation of the device would not require a specially trained, highly skilled operator.*
7. ***Calibration:*** *the time and trouble required to accomplish the calibration procedure should be minimum. Further, the sensor should not require frequent recalibration*

Why optical proximity?

There are two alternatives used in sensing; one can be done by sonar and the next is infra red proximity sensors. The serious caveats Sonar has are:

Cost: *even a sonar setup for small robot use costs around $40 per*

module transducer pair.

Sensor Noise: *Sonar are fairly noisy sensors, readings will jitter and oscillate slightly. There's also the problem of specularity, where the angle between the object and the transducer or the material the object is made of create either false or non-existent readings.*

Minimum Range Limits: *An important concern, sonar have a minimum range at which they can detect objects. For the most popular setup this is around 4-6 inches.*

We have employed optical Proximity sensors, which emits infra red light with range of about 10cms .The basic principle behind this is to illuminate an area with (infrared) light and measure how much returns. As an object comes closer to the sensor, more light will be returned.

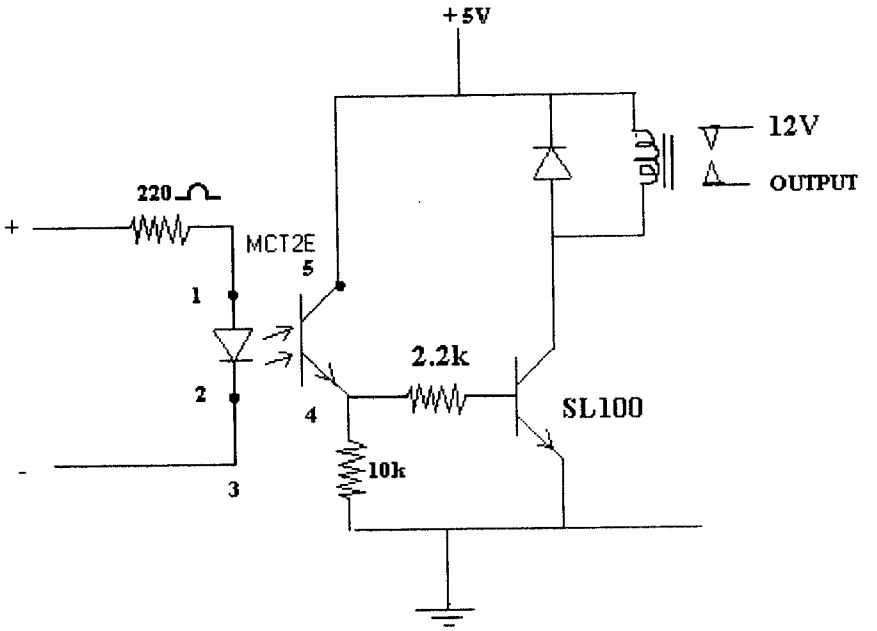
Optical Proximity Sensors

Proximity-controls sense the presence of objects by bouncing light off of the object and detecting the diffuse reflected light. They are used for color detection and material-distinction if there is enough contrast.

Optical proximity sensors use a wide beam source and a selector mounted in the same unit. This works best for large or nearby objects. No alignment of multiple units is necessary, as for through-beam, but it may be desirable to adjust the target so that it has a surface that is more perpendicular than a nearby background surface.

Optical proximity is best suited to situations where the object or material to be detected is brighter (more reflective) or much closer

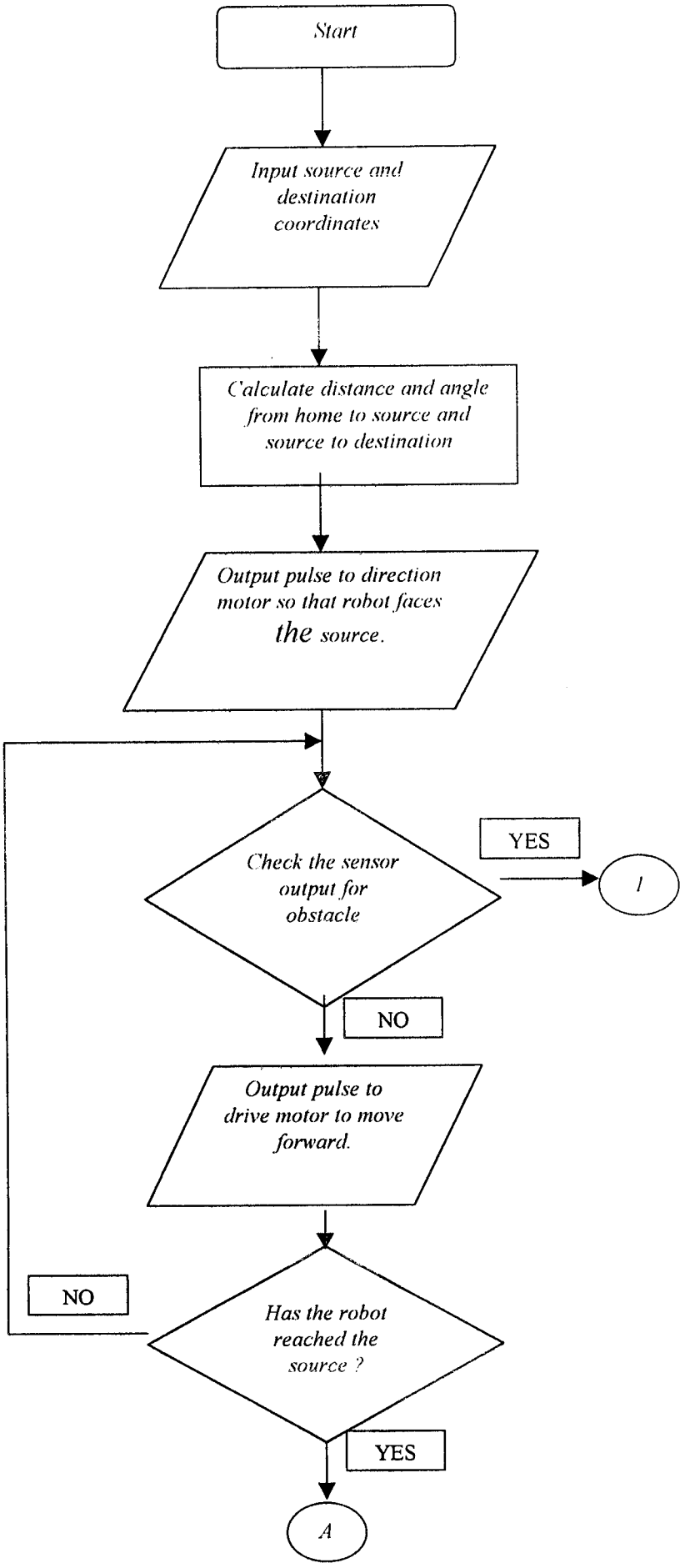
SWITCHING CIRCUIT:

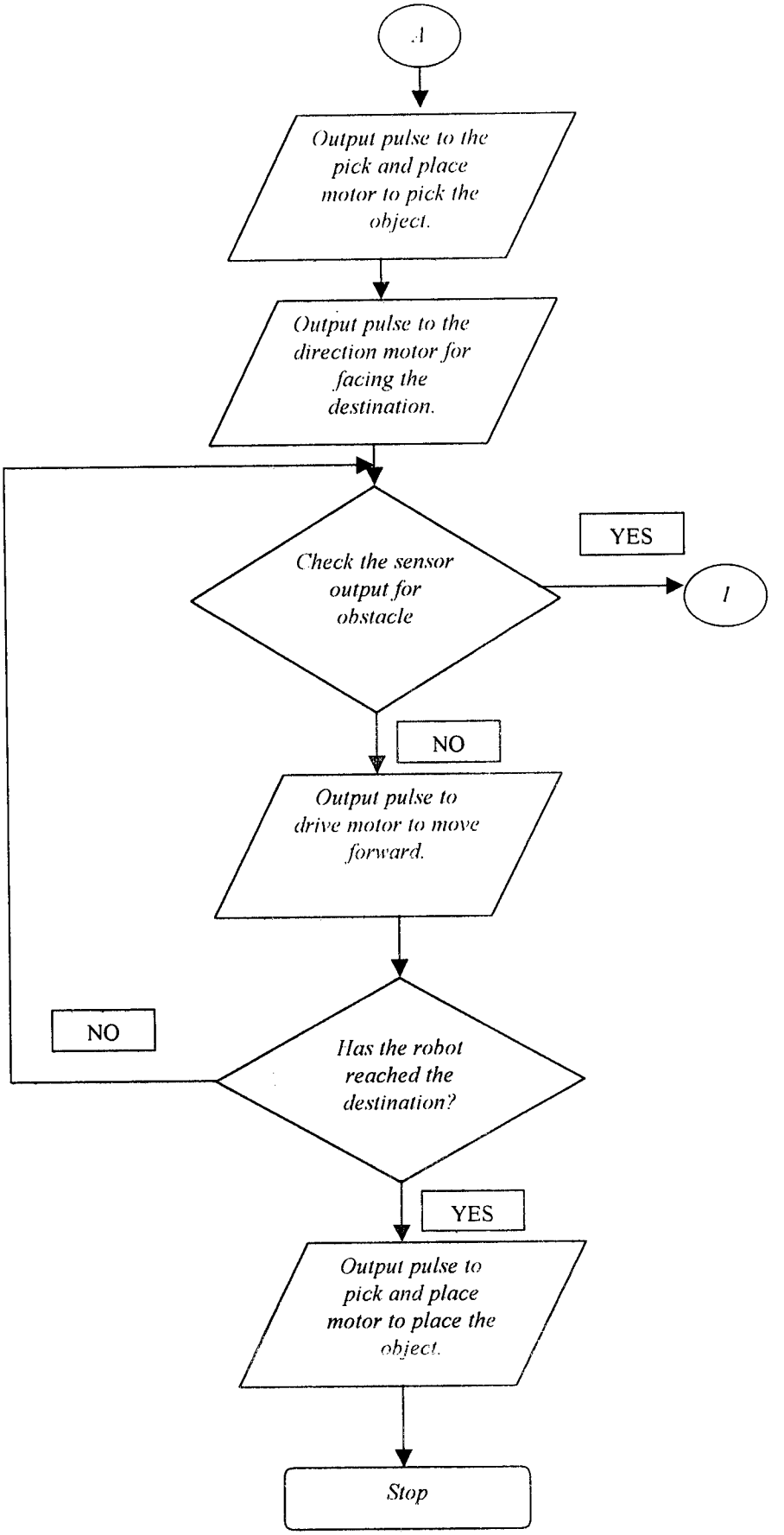


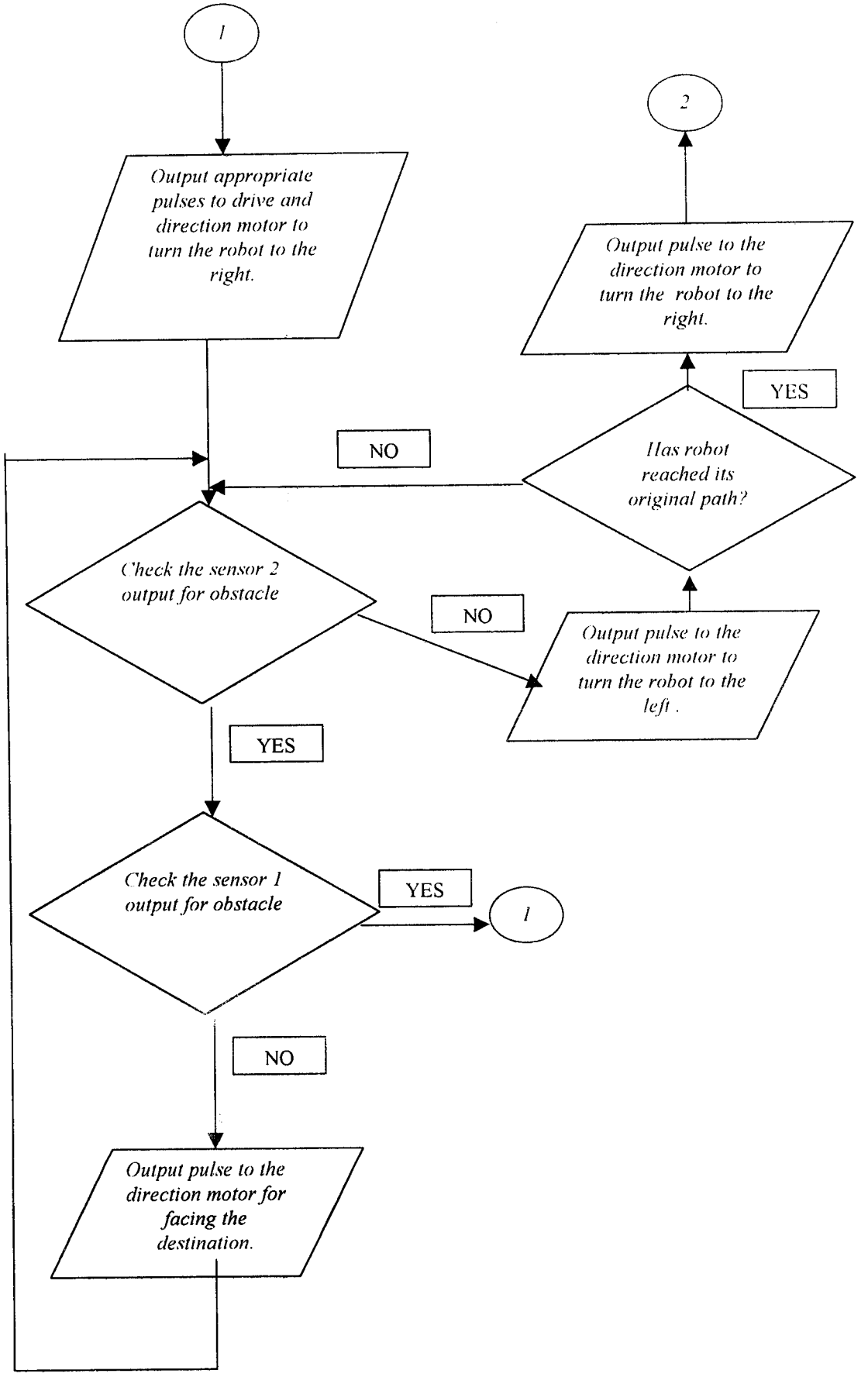
In our project, three stepper motors have been used for the operation of the robot. To activate the appropriate motor at the respective time through the parallel port a switching circuit is used. Control signal are sent to the relay circuit from D4 to D6 lines of the port. Control signal from D7 is used to activate and de-activate the electromagnet. Each of these lines sends a pulse given to it through the program. The appropriate motor drive card is then enabled.

Thus, the respective motor will run according to the data sent through the data lines.

Flowchart


Start
Input source and destination coordinates
Calculate distance and angle from home to source and source to destination
Output pulse to direction motor so that robot faces the source.
Check the sensor output for obstacle
YES
I
NO
Output pulse to drive motor to move forward.
Has the robot reached the source ?
NO
YES
A


A
Output pulse to the pick and place motor to pick the object.
Output pulse to the direction motor for facing the destination.
Check the sensor output for obstacle
YES
1
NO
Output pulse to drive motor to move forward.
Has the robot reached the destination?
NO
YES
Output pulse to pick and place motor to place the object.
Stop


1
Output appropriate pulses to drive and direction motor to turn the robot to the right.
2
Output pulse to the direction motor to turn the robot to the right.
YES
NO
Has robot reached its original path?
Check the sensor 2 output for obstacle
NO
Output pulse to the direction motor to turn the robot to the left .
YES
Check the sensor 1 output for obstacle
YES
1
NO
Output pulse to the direction motor for facing the destination.

Working of the Robot

WORKING OF THE ROBOT

The robot has been designed with its total working in mind. It is its functions that determine the usefulness of a robot. The primary working of the robot is explained in this section.

The robot is designed basically for pick and place. This is a normal operation of the robot but much advancement has been added to it in this project.

Initially, the user is confronted with the coordinates for the source, where picking of the object has to be done, and the destination, where the object has to be placed. These coordinates should lie in the defined area of the robot environment. The robot then calculates the distance from the current coordinate to the source appropriately and starts moving.

A sensor is present in the front of the robot. This sensor is used for detecting obstacle during the moving of the robot from either home to source or from source to destination. The sensor sends its signal to the computer for appropriate control of the robot. When the sensor senses an obstacle, the driving motor of the robot is stopped to stop the robot. The direction motor is then appropriately given pulses to turn the robot and sense for obstacle again. When the obstacle is not in the path of the robot, the robot travels in the new path to reach its destination thus bypassing the obstacle. Only stationary obstacles are considered in our project.

On reaching the source, the robot is given the commands to pick up the object. We have implemented the pick and place mechanism with the help of an electromagnet. Thus only iron objects can be used here for demonstration. Any further extension to the type of object can be implemented with appropriate modification in the pick and place mechanism.

The robot then travels from the source to destination sensing for obstacles. On reaching the destination, the object is placed in the appropriate place and the robot stops its entire functioning as its assigned task is successfully complete.

TESTING PHASE

In a project, the testing phase is essential for its successful completion. Each of the hardware components had to be tested and verified along with corresponding circuits for their efficient working.

Let us examine the test conducted in each of the following hardware components :

- ✓ *Stepper motor*
- ✓ *Stepper motor drive card*
- ✓ *Sensor*
- ✓ *Parallel port*

STEPPER MOTOR

The stepper motors used in this project are:

- *2 * 3kg stepper motor*
- *1* 10kg stepper motor*

The stepper motors are connected to the drive cards according to the color codes of the motor. If the color code is correct from the motor to its drive, the motor shaft will rotate when the program is run in the system. If there is an error in the connection, the motor ceases to rotate. Hence, the various lines of the motor are checked for continuity with the help of multimeter to ensure the error-free status of the motor.

STEPPER MOTOR DRIVE CARD

The drive card serves as the interface between the computer and the motor. The data lines from the computer are connected to the drive card for input. The circuit in connection with the motor is checked for completion when the motor lines are connected.

LED serves as a main testing tool in the drive card. They indicate the energized pole and the power supply status during the running of the program. Also, any voltage drop during work is indicated by an immediate dimming of LED. Thus, the continuous working of the motor implies the proper and accurate working of drive card.

SENSOR

The sensor testing phase proved to be very useful. The sensor is connected to 12V power supply and its output is fed as input to the system for status reporting.

The output of the system is not directly fed to the system due to possibilities of varying high voltage. Hence, an optimizing circuit is connected between the computer and the sensor. This isolation proves to be efficient than connecting the sensor directly to the computer.

PARALLEL PORT

The main interface component between the computer and the robot is the parallel line printer port. The robot is connected to the parallel port through a D 25 pin type connector.

Initially the port was checked for its address by the following DOS command:

```
C:\>debug
-d 0040:0008 L8
```

The output of this as per our computer was 78 03 00 00 corresponding to an address of 0x378.

The output and input pins of the parallel port are checked for their stated operation with the general C statements:

Outportb(port address, value); Inportb(port address);

During tracing of the voltage, multimeter is used for testing by keeping the positive in the respective pin and the negative in the ground pins of the port.

The above tests were the primary tests with respect to the project. A few other testing had to be performed after the total fabrication of the robot.

The main tests were:

- *The wheel alignment correction by running the robot with the entire load.*
- *Pulley arrangement with respect to the rope length and the motor running*

All these tests were done considering the requirements of the project. Many other minor tests were performed during the course of completing the project.

Applications

APPLICATIONS

It is true that "man create machines". But, the machines nowadays excel us, their creators by making the impossible possible. Robots designed for a specific purpose, can be used for multiple applications. Our intelligent mobile robot designed for pick and place can help as:

- *Automated guided vehicle*
- *Material handler in industries*
- *Movement of toxic radioactive materials*
- *Handling of explosives in residential and working environment*

Future Enhancements

FUTURE ENHANCEMENTS

Robotics promises new opportunities for manufacturing and transferring materials to become fully and flexibly automated at every level from an individual work cell to work center to complete departments and factories, all of them using labour-saving technologies.

The Intelligent mobile robot can be improvised on by implementing sophisticated range sensors, hi-tech pick and place mechanisms or grippers. The obstacles can be detected in larger vicinity and its distance in precision can be found by IR range finders and even cameras can be employed for providing robot vision: - an automatic pattern recognition system that would build a digital model of the external world in the brain of a robot.

The mobile robot can be further enhanced by operating from remote hosts in a network. By the accretion of still advanced technologies, it can be made wireless also or it can be made to respond to vocal commands. The mobile robot can also be given life by making it respond by voice.

Robots can carry out the auxiliary operation of loading, unloading and transferring parts and also the assembly operations properly by going from cell to cell both vertically and horizontally in a predetermined pattern and automatic control. Accuracy can be enhanced far by high technological elements.

Conclusion

CONCLUSION

We started this project with petty details in this thought-provoking world of Robotics. Now with the completion of the project, we are a part of this world with a great satisfaction of the practical knowledge we have gained.

The mobile robot interfaced to the computer has been designed to implement the pick and place mechanism. The use of the sensor made the project a highly efficient one. Various tests were conducted and the desired working of the robot with its components was ensured.

Our project forms the basic foundation for many advanced research projects. Further improvements can be accommodated. Incorporation of advanced object identification is one such example. As we have seen there are tremendous implications of our project and it would be quite promising.

In its relative infancy, the state of the art of robotic applications is some ways paralleling development of digital computers. It is to be assumed that over the next few years, non-traditional robotic applications will begin to appear which will in parts contribute to the development of the fully automated factory of the future.

REFERENCES

1. *Mikell P. Groover, Mitchell Weiss, Roger N. Nagel, Nicholas G. Odrey - "INDUSTRIAL ROBOTICS :Technology, Programming and Applications", International Edition 1986.*
2. *Theodore Wildi- "ELECTRICAL MACHINES, DRIVES AND POWER SYSTEMS", Fourth Edition.*
3. *P.A. Janaki Raman – "ROBOTICS AND IMAGE PROCESSING" – International Edition 1995.*
4. *www.dir.yahoo.com/science/Engineering/mechanical-engineering/robotics.*
5. *www.ljkamm.com/robots.html.*
6. *www.robotbooks.com/electronics.html.*
7. *www.mrobot.com/robotbooks.html.*

Appendix

```
//                    Intelligent Mobile Robot //

# include <math.h>
# include <conio.h>
# include <stdio.h>
# include <ctype.h>
# include <graphics.h>

# define data 0x378

struct poin
    {
      int x;
      int y;
    };

//                    Function Declarations //
void robo_forward();
void robo_reverse();
void steer_left();
void steer_right();
void pulley_low();
void pulley_raise();
void grid1();
void travel(struct poin,struct poin);
struct poin randth(double,double,double,double);
struct poin inpu();
```

```
int sensor1();

//                    Type Declaration  //
int x[11],y[11];
int i;
double x1,y1,x2,y2,r,r1,r2,t,nx1,ny1,nx2,ny2;
int j,k=0;
struct poin p1,p2,ap1,ap2, ang1,ang2 ,nang;
float rotang,actang,rotre,rotde,i1,dis1,dis2,dis3,hori,vert;
int motar[4],ctr;


//                    Main Program  //
void main(void)
{
  int gdriver=DETECT,gmode,errorcode;
  clrscr();
  initgraph(&gdriver,&gmode,"c:\\tc\\bgi");
  grid1();
  getch();
  cleardevice();
  printf("Source coordinates of object\n");
  p1=inpu();
  printf("Destination coordinates of object\n");
  p2=inpu();
  printf("%d,%d is the given source point\n",p1.x,p1.y);
```

```
ap1=actpoin(p1);
ap2=actpoin(p2);
printf("The r and angle between home and source:\n");
ang1=randth(0,0,p1.x,p1.y);
printf("The r and angle between source and destination:\n");
ang2=randth(p1.x,p1.y,p2.x,p2.y);

getch();

cleardevice();
grid1();
setcolor(RED);
line(40,440,ap1.x,ap1.y);
line(ap1.x,ap1.y,ap2.x,ap2.y);
printf("%d,%d is source\n%d,%d is destination",p1.x,p1.y,p2.x,p2.y);

getch();

cleardevice();

//                                  initialize array
motar[0]=0x16;
motar[1]=0x1A;
motar[2]=0x19;
motar[3]=0x15;

travel(p1,ang1);          //from home to source
pulley_low();
```

```
 outportb(data,0x80);
 pulley_raise();
 travel(p2,ang2);          // from src to destination
 pulley_low();
 outportb(data,0x00);
 pulley_raise();
}                          // end of main

                    //definitions of sub programs

void travel(struct poin p1,struct poin ang1)
 {
   actang=90-ang1.y;
   i=0;
   while(i!=7)
    {
       steer_right();
       i++;
    }
   i=0;
   i1=3.85*actang;
   rotre=floor(i1);
   rotde=i1-rotre;

//                        rotate robot
   for(i=0;i<rotre;i++)
     robo_forward();
   for(i1=0.0,k=0;i1<rotde;k++)
```

```
 {
    outportb(data,motar[k]);
    i1 = -0.25;
 }
i=0;

while(i!=7)
 {
    steer_left();
    i++;
 }

i=0;
dis1=ang1.x*30;
dis2=floor(dis1);
dis2=dis2/10*50;

l1:
ctr=0;
hori=vert=r1=0;
//      initialize counter to determine no. of loop executions
while(1)
  {
    while(ctr!=dis2)
      {
        if(sensor1()!=0x40)
          {
```

```
    robo_forward();
  }
else
  {
    outportb(data,0x00);
    delay(2000);
    r1=ctr*0.2;
    hori=r1*sin(actang);
    vert=r1*cos(actang);
    i=0;
    while(i!=7)
     {
       steer_right();
       i++;
     }
    i=0;
    i1=3.85*rotang;
    rotre=floor(i1);
    rotde=i1-rotre;
    for(i=0;i<rotre;i++)
     {
       robo_forward();
     }
    for(i1=0.0,k=0;i1<rotde;k++)
     {
       outportb(data,motar[k]);
       i1+=0.25;
     }
```

```
        i=0;
      while(i!=7)
        {
         steer_left();
         i++;
        }
      i=0;
      ctr=0;
      while(sensor2()!=0x08)
        {
         robo_forward();
         ctr++;
        }
      for(i=0;i<25;i++)
        {
         ctr++;
         robo_forward();
        }
r2=ctr*0.2;
nx1=hori+r2;
ny1=vert;
nx2=p1.x*30;
ny2=p1.y*30;
nang=randth(nx1,ny1,nx2,ny2);
i=0;

while(i!=7)
```

```
steer_left();
i++;
}
 i=0;
 i1=3.85*nang.y;
 rotre=floor(i1);
 rotde=i1-rotre;

for(i=0;i<rotre;i++)
{
    robo_forward();
 }
 for(i1=0.0,k=0;i1<rotde;k++)
 {
  outportb(data,motar[k]);
  i1+=0.25;
 }
  i=0;

  while(i!=7)
  {
     steer_right();
     i++;
  }

  i=0;
  dis1=nang.x;
  dis2=floor(dis1);
```

```
    actang=nang.y;
    rotang=90-actang;
    goto 11;

    } //end of else
   } //end of ctr while
  }   //end of infinite while
}  // end of travel function



void robo_forward()
{
 printf("\ninside forward");
 outportb(data,0x15);
 delay(50);
 outportb(data,0x19);
 delay(50);
 outportb(data,0x1A);
 delay(50);
 outportb(data,0x16);
 delay(50);
}

 void robo_reverse()
 {
 printf("\ninside reverse");
```

```
{
 outportb(data,0x16);
 delay(100);
 outportb(data,0x1A);
 delay(100);
 outportb(data,0x19);
 delay(100);
 outportb(data,0x15);
 delay(100);
 i++;
}
i=0;
}

void steer_left()
{
 printf("\ninside left");
  outportb(data,0x25);
  delay(100);
  outportb(data,0x29);
  delay(100);
  outportb(data,0x2A);
  delay(100);
  outportb(data,0x26);
  delay(100);
}
```

```
void steer  right()
{
 printf("\ninside right");
  outportb(data,0x26);
  delay(100);
  outportb(data,0x2A);
  delay(100);
  outportb(data,0x29);
  delay(100);
  outportb(data,0x25);
  delay(100);
}

 void pulley_low()
 {
  while(i!=8)
  {
   outportb(data,0x46);
   delay(100);
   outportb(data,0x4A);
   delay(100);
   outportb(data,0x49);
   delay(100);
   outportb(data,0x45);
   delay(100);
   i++;
  }
```

```
i=0;
}

void pulley  raise()
{
 while(i!=8)
 {
  outportb(data,0x45);
  delay(100);
  outportb(data,0x49);
  delay(100);
  outportb(data,0x4A);
  delay(100);
  outportb(data,0x46);
  delay(100);
  i++;
 }
 i=0;
}

int sensor1()
{
 int in;
 in=inportb(0x379);
 printf("\n%x",in);
 in=in&0x40;
 return in;
}
```

```
int sensor2()
{
 int in;
 in=inportb(0x379);
 printf("\n%x",in);
 in=in&0x08;
 return in;
}

struct poin inpu()
  {
  struct poin sp1;
  printf("enter x value\n");
  scanf("%d",&sp1.x);
  printf("enter y value\n");
  scanf("%d",&sp1.y);

  //                evaluate if x and y value within specified range
  if (sp1.x<=10&sp1.y<=10)
  return sp1;
  else
    {
     printf("invalid coordinates\n enter the values once again\n");
     inpu();
     }
  }
```

```
void grid1()
{
int i,j,k=0;
for(i=40;i<=440;)
 {
    line(40,i,440,i);
    i+=40;
 }
for(i=40;i<=440;)
 {

    line(i,40,i,440);
    i+=40;
 }
for(i=40,j=440,k=0;i<=440,j>=40,k<11;k++)
 {
    x[k]=i; y[k]=j;
       i+=40; j-=40;
       //printf("%d,%d\n",x[k],y[k]);


 }
//for(i=0;i<=10;i++)
  outtextxy(40,450,"0");
  outtextxy(80,450,"1");
  outtextxy(120,450,"2");
  outtextxy(160,450,"3");
  outtextxy(200,450,"4");
```

```
 outtextxy(240,450,"5");
 outtextxy(280,450,"6");
 outtextxy(320,450,"7");
 outtextxy(360,450,"8");
 outtextxy(400,450,"9");
 outtextxy(440,450,"10");

 outtextxy(26,80,"9");
 outtextxy(26,120,"8");
 outtextxy(26,160,"7");
 outtextxy(26,200,"6");
 outtextxy(26,240,"5");
 outtextxy(26,280,"4");
 outtextxy(26,320,"3");
 outtextxy(26,360,"2");
 outtextxy(26,400,"1");
 outtextxy(26,40,"10");
}
struct poin actpoin(struct poin acp)
{
 struct poin acp1;
 acp1.x=x[acp.x];
 acp1.y=y[acp.y];
 return acp1;
 }

 struct poin randth(double x1,double y1,double x2,double y2)
```

```
  struct poin ang;
 double r,r1,t;
printf("x1=%lf,y1=%lf",x1,y1);
printf("\nx1=%lf,y1=%lf",x2,y2);
r=((x2-x1)*(x2-x1))+((y2-y1)*(y2-y1));
r1=(double)sqrt(r);
ang.x=r1;
printf("r1=%lf",r1);
t=asin((y2-y1)/r1);
ang.y=t*180/3.14;
printf("\nt=%lf",t);
 return ang;
 }
```