1. INTRODUCTION

1.1 PROJECT DETAILS

1.1.1 PROJECT DEFINITION:

Design a mobile robot with gripper based on ATMEGA series microcontrollers. The following are the major specifications of the robot:-

1. Four wheel DC motor drive with gear.
2. Horizontal gripper with pick and place.
3. Array of optical sensors to track path.
4. Proximity sensors to avoid obstacles.
5. Interface with remote control.

Optional specifications:-

1. Interface 4GB SDRAM for path data acquisition and history storage.
2. Automatic charging of its own battery using docking station.

The system has 3 different modules – obstacle detector module, the line tracker module and IR remote interfacing module, along with motors and a motor driver. These two modules integrated together form the entire system performing the different functionalities needed.

The obstacle detector module senses an obstacle, and either follows it or changes its path according to the different modes set.

The line tracker module detects a path which it has to follow and then follows the path. If in case it senses an obstacle then it'll try to change its path or wait at that position for a while and then follow the same path again after making sure that it has no threat ahead. These functions also depend on the modes set.

The IR remote interfacing module helps the user to control the robot using an RC5 remote control. The different modes of the robot can be set as per the commands issued by the remote control.

All these functionalities can be performed at different speeding grades thus enabling the user to use the robot according to varied requirements of situation which includes environments in which it will operate.

Apart from all this if the time permits interfacing the SDRAM card with microcontroller is to be performed which will help to store the location where robot is. Thus location will be stored in form of coordinates to remember where it is and to easily go to the place where it has already gone through which is stored as history.

It will also include the automatic charging station for robot to charge itself when the battery level is below the optimum level to move robot appropriately, thus making it completely mobile and less user dependent and after charging is completed it automatically moves from charging station to its normal activity.

1.2 PURPOSE

To design the best possible mobile robot from given resources which can provide a multi-processing and a multitasking system which can do a certain number of important activities necessary in daily life like sense an object nearby, follow and guide through a path, lift and place objects, etc. All these functionalities can be achieved at different speeds and can be controlled from distant region through remote. It is designed in such a way that according to different situations it is placed in, it can take actions appropriate to get solution of that situation. These all functions can be performed by CPU which is thus designed to perform multiple tasks at a time utilizing maximum CPU efficiency. Also the system would be multiprocessing in that there'll be more than one CPUs working at a given time which will together process and give the final result.

1.3 SCOPE

The robot can perform multiple actions such as:

1. Detect obstacles and take appropriate action.
2. Find its own path and follow it, i.e. a line tracker.
3. It can work as a gripper, i.e. it can be used for picking and placing things.
4. It can be operated by IR-remote

It also has certain drawbacks like:

1. The gripper cannot function properly for all sizes, shapes and weights of an object.
2. The distance from which the robot can be operated through remote is limited.
3. The system is battery operated hence it has a greater chance of getting damaged.

1.4 OBJECTIVE

The objective to build this kind of a system was to make a multi-purpose auto-mobile robot which if enhanced can be made as powerful medium for security and other basic services at various places such as offices, high security regions like Railway station and Airport. It can also be enhanced to be used for general house hold services and a guide to elderly persons in society. The key objective in designing a robot over manpower is that it can be trusted along with greater efficiency.

Another important objective is to achieve the task of networking between CPU thus making it possible to perform more than one function at a time and according to the result obtained by one CPU to be analyzed and action taken by another CPU, thus obtaining master-slave like functionality which again is a gateway to perform multiple activities.

1.5 TECHNOLOGY AND LITERATURE OVERVIEW

1.5.1 TECHNOLOGY

1. ATMEL ATMEGA series microcontroller - ATMEGA8L.

   The ATmega8L is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8L achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

   This microcontroller was the best option in terms of cost also in addition to providing the necessary functionalities.
2. Proximity (TSOP) sensors and optical sensors.

   The proximity sensors detect a distant object from the place it is located, which is necessary for object and obstacle detection. TSOP sensors are used for sensing and IR LEDs are used here because they have a better range as compared to most other light emitters and also their light sensitivity is also more.

   The optical sensors are one of the easiest and best ways to detect colours and hence follow a path. Hence optical sensors have been used for path tracking.
3. Timer IC - 555.
4. 100 rpm Motors and Motor driver IC L293D.
5. Lithium ion battery.
6. USB Programmer for programming.

1.5.2 LITERATURE

1. Datasheet ATMEGA 8L and ATMEGA32.
2. Robokits USB programmer user manual.

   The Robokits USB programmer is used to program the microcontroller chips through a USB interface. The programmer transfers the Hex code from the computer to the microcontroller system.
3. Different C compilers’ manual which can produce a Hex code from the C code.

   C compilers like WINAVR, AVR Studio4 and CodeVision AVR have been used for compiling the C code and producing the Hex code.

4. Gripper study.
5. Remote (RC5) interfacing.

    Most audio/video systems use an infrared RC5 remote control. Hence they are easily available and cost efficient. The RC5 remote receiver, required for these remote controls and a similar one we have used, is easy to implement, code, and also low cost, along with all these it is compatible with the microcontroller used.

6. Schematic circuit diagram making tools.
7. Study of software tools to make PCB.

2. PROJECT MANAGEMENT

2.1 FEASIBILITY STUDY

Various types of Feasibility study are identified as under:

2.1.1 TECHNICAL FEASIBILITY

The technical Feasibility test involves questions like

- Is the current Computer's configuration and system technology adequate for Usage?
- Is the selected technique sufficient for future enhancements?
- Are the manpower and resources available sufficient for development and maintenance?

All these feasibility constraints are studied and satisfied.

2.1.2 OPERATIONAL FEASIBILITY

The users of the system should be able to operate it easily. This demands good user interface. The system we developed can be operated using any audio/video remote control widely in use these days by any common man. A reference to the User Manual is required for the user to learn about the modes of operation. Hence there is a very little possibility that the user would resist using the system.

2.1.3 ECONOMIC FEASIBILITY

The questions put forward in economic feasibility are:

- Are there sufficient cost benefits in creating the system?
- Are the costs of implementation of current system so great that the task of project development is required?

This feasibility study measures the cost effectiveness of project. It takes into consideration cost and benefit. Our project satisfies this criterion also.

- Hardware-Software Cost

This feasibility is of paramount importance in development of any system. The system requires C compiler softwares which are easily available on the internet for free. The software which converts C file to hex format which could be programmed into the microcontroller are also available on internet. The only cost under concern is the hardware cost which includes cost to construct entire system. Apart from the system the things required to program the system should be considered and cost in utilizing this hardware was to be considered.

- Maintenance Cost

Any system should be built such that each and every module of the system is easily and independently accessible. This is a major concern for maintenance. In case the modules are not easily accessible they increase the cost of maintenance in that all the other modules on which the present one is dependent on will have to be disturbed. Also this cost includes the hardware maintenance cost of the system which is not an issue in our system unless there is any damage. Apart from it the maintenance cost includes the charging adapter through which robot is charged and made usable and electricity supply to charge it.

2.1.4 TIME SCHEDULE FEASIBILITY

To build any system the factor of time should be of prime importance. The project should be such that it could complete its specification in allotted time with complete layout. The study on time feasibility showed us that our project with major specification can be completed in allocated period of time.

2.2 PROJECT PLANNING

2.2.1 PROJECT DEVELOPMENT APPROACH AND JUSTIFICATION

The requirements of the system keeping the user in mind are first analyzed, revised, validated and approved by the users. The development commences after the approval phase. The system designer and developer will try to adhere to the requirements mentioned above in order to develop the required system.

The system requires development of both hardware and software. Thus the reversible waterfall model is used for developing our system. The development of system will include study of requirement analysis, feasibility study, planning entire design. The testing will include the unit testing after every phase is completed and then integration testing and System testing. If any change in prior stage, this model will enable us to review the system.

2.2.2 PROJECT PLAN

The Project planning includes understanding of the requirements completely and abstract designing of entire system. The entire design cannot be made at once so the system is divided into subsystems and modules and each module is to be designed and implemented one by one.

The first task is the feasibility study in planning of any project according to the requirement specification. If the project is feasible then we must move on to the designing and developing system.

The premiere task would be study of requirement to make a robot like structure which would include two motor drives and basic hardware construction to make robot move.

Then robot should be programmed for performing different moves to test the efficient movement of robot. Then our project would be divided into Four major parts:-

1. Design, Construction of robot having obstacle detecting capacity and then the appropriate programming to work according to condition i.e. we can program it to avoid the obstacle or also to follow the moving obstacle.
2. Design, Construction of a line tracking robot and programming it accordingly.
3. Design of gripper that would be able to perform many tasks efficiently. Such that it can grip any kind of objects easily and can take heavy loads to transport from one place to another.

   The construction of gripper requires study of grippers available in market and also the cost should be considered. If cost is an issue the most easy and cost efficient way should be made to perform the design.
4. The other task would be to interface Remote control with robot in such a fashion that robot moves according to remote controls.

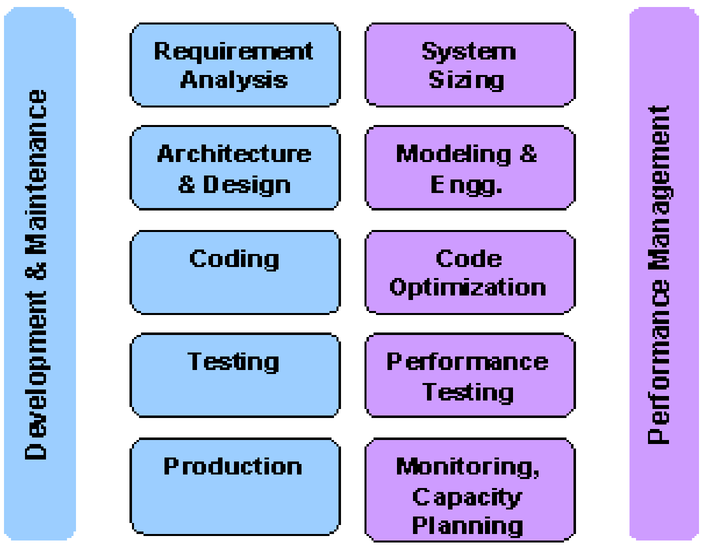


Fig 2.1 Project Planning and Management Approach

2.2.3 MILESTONES AND DELIVERABLES

Milestones are identified in order to complete the entire project in the time duration. Milestones are identified for every module of our system.

PHASE	DELIVERABLES	PURPOSE
System Requirement and Analysis	• Requirement Gathering and analysis. • Functional Specifications • Non Functional Specifications	It gives exact understanding of the user’s requirements.
System Design.	• Abstract design • Requirement for design • Final Construction • Designing flow of operation • Discussing and designing the working	It gives the logical as well as physical structure that describes the system.
Implementation	All the requirement are one by one implemented and worked upon	It gives the output.
Testing	The output obtained for the required functionality after implementing and doing various types of testing	It gives the required module

Table 2.1 Milestones and Deliverables

2.2.4 ROLES AND RESPONSIBILITIES

As a system designer and developer we must ensure that the system should fulfil its specification. It should be efficient and consistent. There must not be any anomalies and redundancies in it. Apart from all this it must ensure the security and safety concerns.

Project Team Structure

The Robot project had following team structure:-

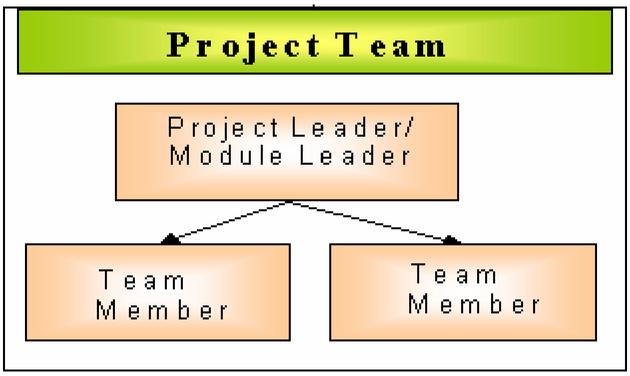


Fig 2.2 Project Team Structure

- **Project Leader**

The responsibilities of the Project Leader are:

- Planning the application customization/development work in the Project.
- Monitoring the day-to-day status of application, system development/customization activities.
- Planning and scheduling of enhancement of system in co-ordination with Team members.
- Ensuring the quality of the deliverables.
- Discussing and giving clarifications necessary to the team members regarding Functionality, system construction etc.
- Manage the flow of work within his/her team and allocate work to team members.
- Monitor the Quality of the deliverables being created.
- Maintain a clarification document and ensure that any change or clarification is reflected in the design documents.

- **Team member**

The responsibilities of a team member include:

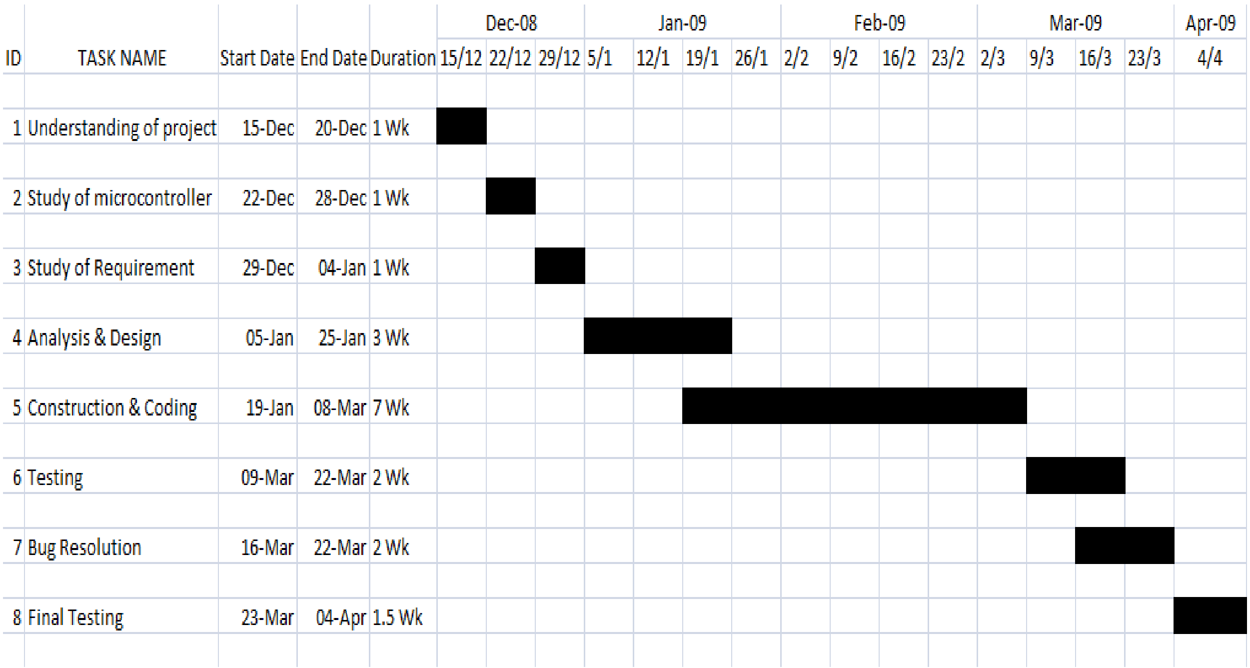
- Carrying out the work allocated
- Report on a weekly basis to Project leader.

2.2.5 GROUP DEPENDENCIES

As the entire project was developed by one group there were no group dependencies present.

2.3 PROJECT SCHEDULING

2.3.1 Time Allotted to Various Activities



3. SYSTEM REQUIREMENTS STUDY

After studying the functionalities, a requirement study was done. All the requirements necessary for developing the system were considered keeping in mind the cost and the time constraints.

3.1 STUDY OF CURRENT SYSTEM

As per the requirement study done, the system needs:-

- 2 ATMEGA8 microcontrollers.
- TSOP sensors (proximity) and optical sensors.
- Timer IC LM555
- 100 rpm motors and a motor driver IC L293D.
- Other components like resistors, capacitors, regulators, condensers etc. for current control, cooling and power regulation.

The system can perform the following functions:-

- It can sense an obstacle in its way and avoid or follow it, using the proximity sensors.
- It can find and follow a given path.
- It can grip, pick and place objects from one place to another.
- It can perform all these different actions as per the user commands using a remote control.
- The system also makes sure that power is not wasted, i.e. when a module is not being used then it makes sure that only those modules which are working use the power. This is done using a transistor B548 B by switching the power supply on and off as and when needed. For example, if the robot is in the obstacle detection mode, it doesn’t provide supply to the line tracker module, hence not wasting the power but utilizing it.

3.2 PROBLEMS AND WEAKNESSES OF CURRENT SYSTEM

- The line tracker in the current system is not capable of following a path having steep turns of less than 30^0 bend from the horizontal.
- The proximity sensors are insensitive to black coloured objects; hence they cannot detect black coloured objects.
- The system has to be charged frequently and no automatic charging facility is there.
- The gripper is not strong enough to lift objects of all shapes, sizes and weights.
- The system is battery powered; hence there are chances of it getting burnt due to high short-circuit currents in it.

3.3 USER CHARACTERISTICS

Any of the functionality performed by the staff in the organization involves a flow of information along predefined hierarchy of users. The flow consists of information along with hierarchy or outside the system. It involves reviewing, commenting and approving along various officers in the hierarchy. In our case, there is a project guide and we, the trainees.

3.4 HARDWARE AND SOFTWARE REQUIREMENTS

3.4.1 HARDWARE REQUIREMENTS

- The system needs a 12 V battery supply for the system to be alive.
- USB Programmer board to program the system.
- USB cable for interfacing the board with the computer.

3.4.2 SOFTWARE REQUIREMENTS

- Programmers' Notepad.
- WINAVR Compiler.
- USB Programmer software – Robokits USB Programmer or AVR Studio or Code Vision AVR.

3.5 CONSTRAINTS

3.5.1 HARDWARE LIMITATIONS

- The system uses battery power supply, hence causing the need to frequently charge the system.
- Care has to be taken always that the supply is proper, otherwise there are chances of system getting damaged.
- Overcharging the system may lead to an explosion in the system.

3.5.2 INTERFACES TO OTHER APPLICATIONS

- The system as a whole can be interfaced to any hardware application depending on the number of ports, etc. needed and other hardware requirements

3.5.3 HIGHER ORDER LANGUAGE REQUIREMENTS

- The entire coding is done in C language. WINAVR programming environment is needed for this.

3.5.4 RELIABILITY REQUIREMENTS

- For the reliability of the system in its working, the user has to make sure that it is completely charged and there is no internal damage in circuitry.

3.5.5 CRITICALITY OF THE APPLICATION

The project has mainly four modules, namely – the IR remote interfacing module, the line tracker module, the obstacle detector module and the gripper module. Of these some of the critical issues were as follows:-

- A major problem was faced with IR remote interfacing module, because we couldn't get the input read properly, solving which took a long time.
- Only a dummy gripper module had been designed, which could not be turned into a working module because of the lack of mechanical knowledge needed for its construction. Later, after a thorough study, the task was accomplished.

3.5.6 SAFETY AND SECURITY CONSIDERATIONS

- For the system to be safe, it should not be overcharged, i.e. the maximum its battery can be charged is up to 12V.
- Because the battery used is Lithium ion battery, care should be taken while it is being charged like it should be kept in some isolated place so that in case of any blast etc. it doesn't affect the surroundings.

3.6 ASSUMPTIONS AND DEPEDENCIES

- The system assumes that chances of internal damage are negligible until any external damage is done to the system.

- The system is completely dependent on the power supply it gets for it to work properly.

4. SYSTEM ANALYSIS

4.1 REQUIREMENTS OF NEW SYSTEM

4.1.1 SYSTEM REQUIREMENTS

Study has been conducted to understand the requirements of the system.

Requirement gathering of the system will enable us to analyze and propose a new system. This will be the basis for the development and implementation of the system for better efficiency, effective utilization of manpower and hence the desired system.

R1: Sense the obstacle.

Precondition: - The robot set in the obstacle avoider mode.

Constraint: - The obstacle should be in the height range of the sensor.

Input: - Mode command through the remote control.

Output: - Robot sensing the obstacle if there is one in its way.

R2: Avoid an obstacle.

Precondition: - The robot is set in obstacle avoider mode.

Constraint: - The obstacle should be in the height range of the sensor.

Input: - Mode command through the remote control.

Output: - In case of an obstacle in the robot's way, the robot changes its direction.

R3: Follow the path.

Precondition: - The robot should be in line tracker mode

Constraint: - The path should be black.

Input: - Mode command through the remote control.

Output: - Robot follows the path.

R4: Change speed of the robot.

Precondition: - The robot should be in any particular in which the speed of the robot matters.

Constraint: - Robot is not halted.

Input: - Speed grade command through the remote control.

Output: - Robot travelling in the said speed.

R5: Pick and Place an object through gripper

Precondition: - There should be an object to pick and place and robot should be set in gripper mode.

Constraint: - The object should be of appropriate size and shape so that robot gripper can be able to pick and place that object where ever required.

Input: - Mode command for gripper in action.

Output: -The robot performs the actions like picking an object and placing it.

R6: Follow an obstacle.

Precondition: - The robot is in obstacle follower mode.

Constraint: - The obstacle should be in the height range of the sensor.

Input: - Mode command through the remote control.

Output: - In case of obstacle in its way robot will follow that obstacle.

Apart from all this, the conditions mandatory to system working are

R7: Charge the battery to drive system.

Precondition: -The battery used to drive the system should be having charge less than required threshold level.

Constraint: - The system should work with battery.

Input: - Plug-in charger.

Output: - System should be charged.

R8: Switch System on.

Precondition: -System is off.

Constraint: - The system should be charged enough.

Input: - Switch on the power supply.

Output: - System ready to work.

4.2 FEATURES OF NEW SYSTEM

The new system has to be designed such that it performs different functions and it can be autonomous with enhancement. The present system is the prototype of the original system and its features are as follows:-

- Obstacle Sensor
  As the name suggests robot senses obstacles in its range/dormitory and convey it to the controller. For this purpose TSOP sensors are used. The Controller will in turn take appropriate steps.

- Line tracker
  The system can sense the path and then will follow that path. This would work as a dedicated path for certain important activities such as charging of battery and disposing of any unwanted object.

- Gripper
  Designing an efficient gripper which is also simple, economical and light in weight is a difficult task, and this is what we have concentrated on. A completely 'Do it yourself' concept of constructing a robot gripper is designed, fabricated and presented.

  The gripper functions include:-

  1. Opening and closing of the gripper levers (jaws).

  2. The 'lifting up and down' motion of the gripper.

  The gripper performs these actions in a sequence i.e. closing its arms and gripping, lifting up, placing down and then releasing.

- IR remote Interface
  The remote interface will help any user to be able to use the robot from some distance easily. The robot can then be controlled in above mentioned modes using remote.

- Distributed/multiprocessing CPU: - The system will use Distributed CPU architecture i.e. more than one CPU will be used in the system and the CPUs together will do the processing and decision making and thus control the robot, thus enhancing the functionality of robot.

- Apart from all the above functionality the new system is designed and programmed in such a way that waste of current can be avoided and system can work efficiently till longer time duration. Thus making of the new system had large amount of time spent on its design which could be avoid any kind of waste in current and thus power.

5. SYSTEM ARCHITECTURE

This chapter describes the system architecture with the available components, how it is the best in its way, its pros and cons and why the other alternatives available have not been considered.

5.1 SYSTEM ARCHITECTURE AND DESIGN

This section describes the logical model of the system and its architecture as to what components are to be used, where and how.

5.1.1 TYPE OF COMPONENTS REQUIRED AND USED

- Microcontroller

A microcontroller is needed to program the robot as per the requirements. A microcontroller is chosen instead of a microprocessor because the former has a microprocessor in it, in addition to which it has other important features like timers, ram, rom, interrupts, etc. built in on a single chip. When finalizing a microcontroller, along with its features, the cost constraints should also be taken into consideration, to make sure that all the constraints are met. The microcontroller we have chosen for our project is ATMEL ATMEGA8L. This was finalized because it was the best option satisfying the cost constraints and fulfilling all the needs. In addition it has certain important features like In System Programming, independent Flash and EEPROM memories, etc.

- Distributed CPU v/s Single CPU

In a distributed CPU architecture, as the name suggests, CPUs can be interfaced on a large scale (2-10) and communication between them can be made possible, thus leading to an increase in the through-put of the system and enabling it to perform a lot many functions than it could do if it were a single system. Also many external devices like sensors, timers, motors, etc. can be interfaced. There are certain drawbacks of using a distributed system like increased complexity, need for more hardware, etc. but these are negligible in comparison to the advantage it provides. This approach can be used in situations where we need a multitasking system performing a large number of functions.

In a single CPU architecture there is just one CPU which controls the entire system, hence reducing the complexity and hardware, system overhead, etc. in comparison to distributed CPU architecture. This approach has certain disadvantages also like the CPU is limited to perform only certain number of actions. This architecture can be used when the system is very simple and has not to perform many functions.

We chose to use the distributed CPU for multiprocessing in architecture because of number of functions the system performs at a time. Also future enhancements are very much possible in the system and using this architecture increases the flexibility of the system for the same.

The system performs operation like:-

1. Sensing the path
2. Sensing the Front Obstacle.
3. Sensing the Left Obstacle.
4. Communication between two CPUs
5. Performing Motor operations according to decision taken
6. Gripper Functioning

- Sensors

Sensors are needed in our system for two purposes:–

1. Detecting an obstacle.
2. Tracking the path.

For detecting any objects in the range set according to potentiometer, the sensor used should convey to the system that there is an obstacle in its region. The sensor used for such task should be sensitive enough to detect any obstructions in its dormitory which can be done with an IR led that radiates light rays and the sensors detect it. In case of obstruction the rays reflected back are different and thus obstacle is detected. This kind of mechanism is required to detect obstacle. A problem with such a sensor is that it cannot detect any black coloured objects because they don’t reflect any light, in contrary they’ll absorb all the rays thrown by the sensor.

A similar mechanism is used by the sensors used to track path, where the optical sensors are used to track the path. These are placed near the light emitting source. Thus when rays are not reflected means the rays are been absorbed by the surface. The rays can be absorbed by only black surface but reflected by other surface thus black path can be detected.

We have used TSOP sensors for obstacle detection because of its high sensitivity and IR sensors as path detectors which can easily track path.

- Battery used

The battery used in our system is a Lithium ion battery. Li-ion batteries are now-a-days becoming a practical alternative for robot power. They are smartly packed to avoid any accidents and are long lasting and can be charged very quickly, i.e. it takes a maximum of two hours to get charged from dead to full. It has certain drawbacks like the cost of its charger is high and hard to find compared to other batteries in use. It also has chances of damage or explosion. Also the voltage is not constant, i.e. in that the battery operates over a range of 12V – 10.8V when fully charged, which may be a problem in certain cases, not in ours.

- Motors

The motors used are reversible, geared, 100 rpm, requiring 12V supply. Reversible motors are capacitor-run, single phase motors. These are suited for applications requiring a frequent change in the direction of the motors, like in our case. Geared motors have a mechanical advantage over the other motor types, that gears of unequal sizes can be combined to attain different rotational speeds and torques.

Different options like stepper motors, pneumatic motors, etc were also available but they have a number of limitations. Stepper motors have disadvantages like, precision and speed limitations, inability to spin freely like other DC motors, more vibration in comparison to other motor types which may prove to be bad at very high speeds and also leads the motor to lose a significant amount of torque, etc. The pneumatic motors, in spite of all the good they provide, prove to be less efficient in comparison to the other motor types. Moreover none of these provide the reversibility needed in the system.

In total, three motors are used in the system, two of them for driving the wheels of the robot and one for the gripper.

- Gripper

The gripper is used for pick and place of objects. Many gripper designs have been studied which are available on the internet. Almost all of them use two or more motors for its working, i.e. the gripper drive, then its pick and place actions. The use of more motors increases the load on the system. To reduce this, a design using just a single motor was to be implemented which seemed to be necessary and sufficient requirement for the gripper to work properly.

5.1.2 SYSTEM ARCHITECTURE

After a thorough study of the components to be used and how, the system is designed such that it uses all possible resources in the best possible way.

The Figure 5.1 shows a block diagram of the system. As it can be seen in the diagram, two CPUs are interfaced with each other. The first CPU is used for remote interface, obstacle detector and gripper and the second CPU is used for the line tracker module. The microcontroller ATMEGA8L has 3 I/O ports of which two are 8-bit and one is 7-bit. The ports used for each module are also shown.

- Obstacle avoider uses PORTC0, PORTC5 & PORTD2
- Line tracker uses PORTC1, PORTC2, PORTC4 in CPU 1 and PORT C 2 -5 PORTB 1 in CPU2
- Gripper uses
- Remote interface uses PORTD6

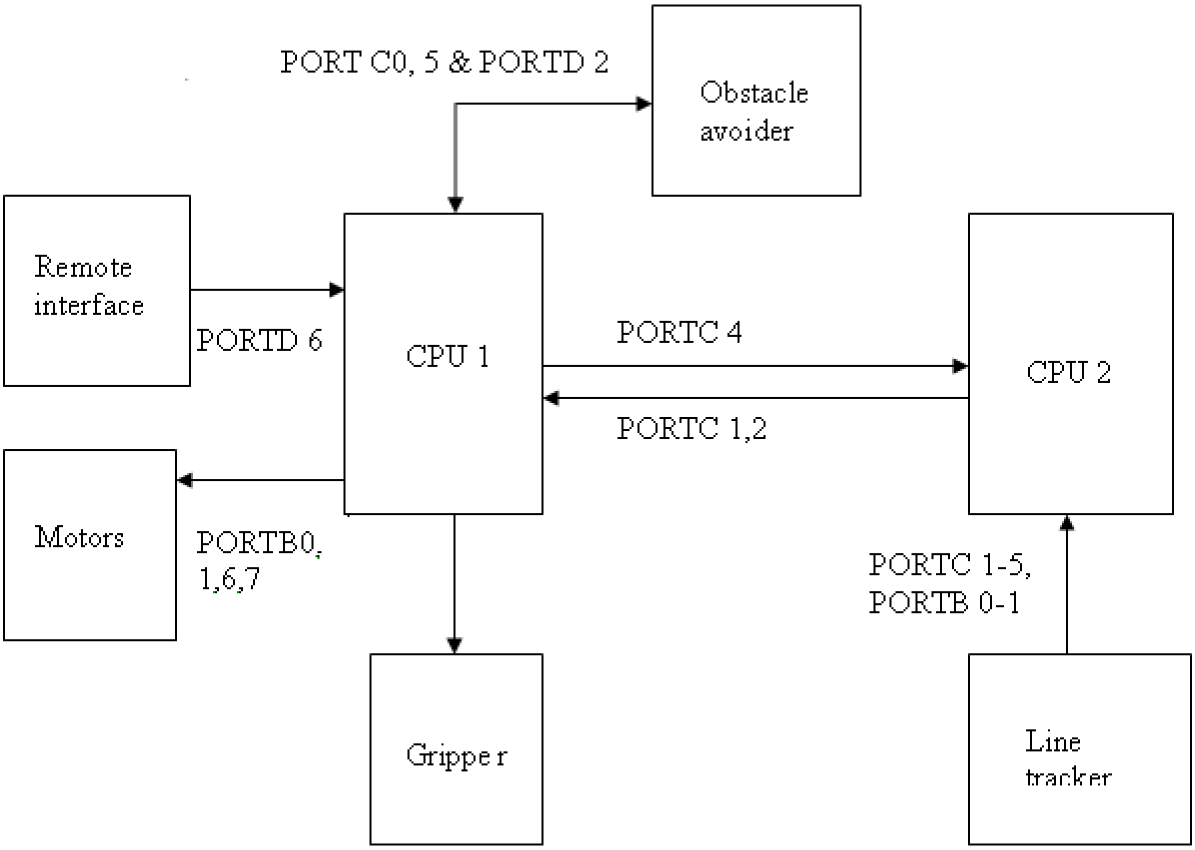


Figure 5.1 Block diagram of the system

5.1.2.1 GRIPPER ARCHITECTURE

The gripper we designed for our system includes a single motor which drives the gripper, and also does the pick and place. The proposed design uses thread links to perform both the gripper functions as stated above, using a ‘reversible’ 100rpm motor. The motor shaft winds and unwinds two strings at a time which in turn opens, closes, lifts up and down the gripper jaws, as desired is shown in the Figure 5.2 below.

Figure 5.2 Gripper in action

The proposed gripper has four Degrees of Freedom (DOF).

- What is Degree of Freedom?
  In mechanics, DOF of a system is the number of independent displacements and/or rotations that specify completely the displaced or deformed position and orientation of the body or system. The number of DOF of a system is the number of parameters of the system which can be independently varied.

As stated above, our gripper has four DOF as below: -

1. The gripper arms can open and close freely in space.
2. The lifter platform can move up and down freely.
3. The entire system to which the gripper is connected can move forward and backward.
4. Also the system can move leftwards and rightwards.

The Figure 5.3 shows the construction of the gripper. It consists only of three parts connected to the robot base using a common door hinge. The gripper arms are connected to the lifter platform using bolted joints and the pull points of the gripper arms are connected to the motor using strong threads which respond to the motor motion. The gripper arms are easy to make and provide up to four Degrees of Freedom. The disadvantages with this gripper-arm construction are that they are stationary unless mounted on a mobile platform and the cost to build is proportional to the lifting capabilities. The lever action used to open, close, lift and place is done with a counter spring to avoid backlash. The construction time with the help of minimal tools like hex saw and drill is less than one hour.

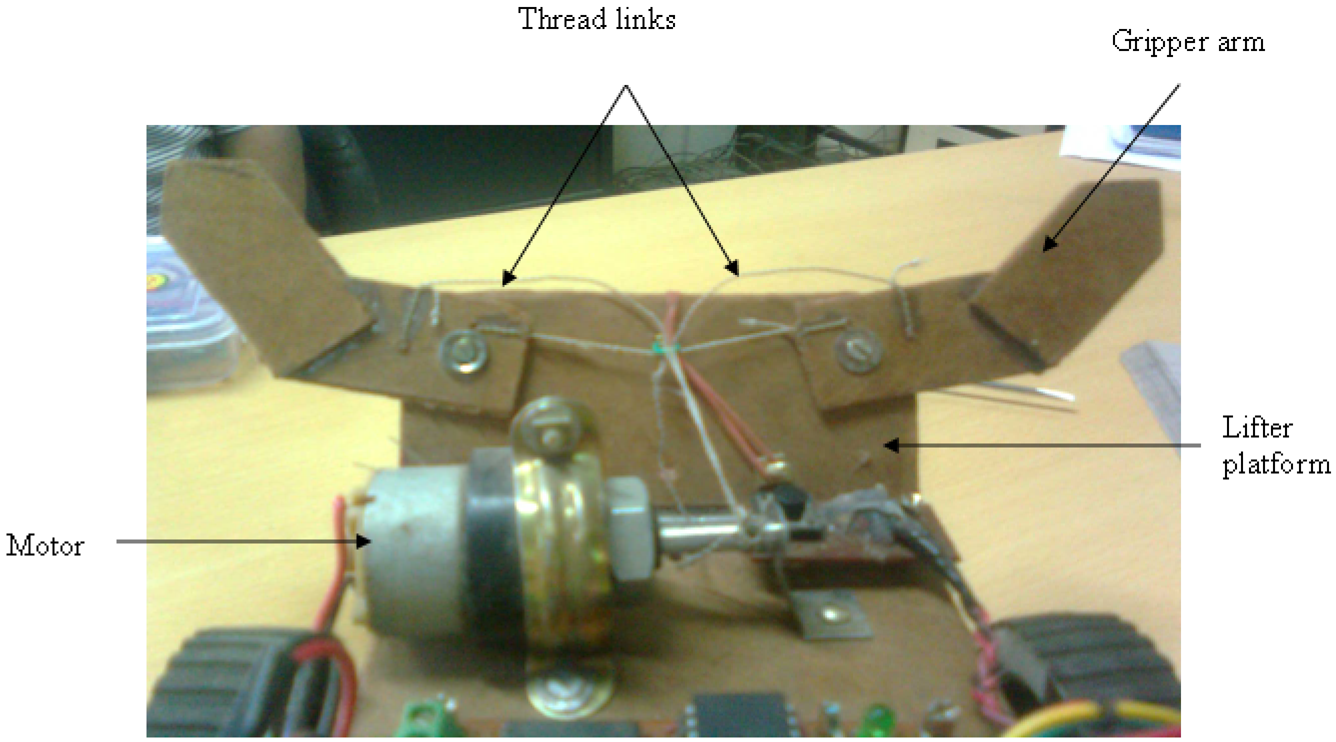


Figure 5.3 Construction of gripper

5.1.2.2 SENSOR BLOCK

The figure below shows the sensor block of the system.

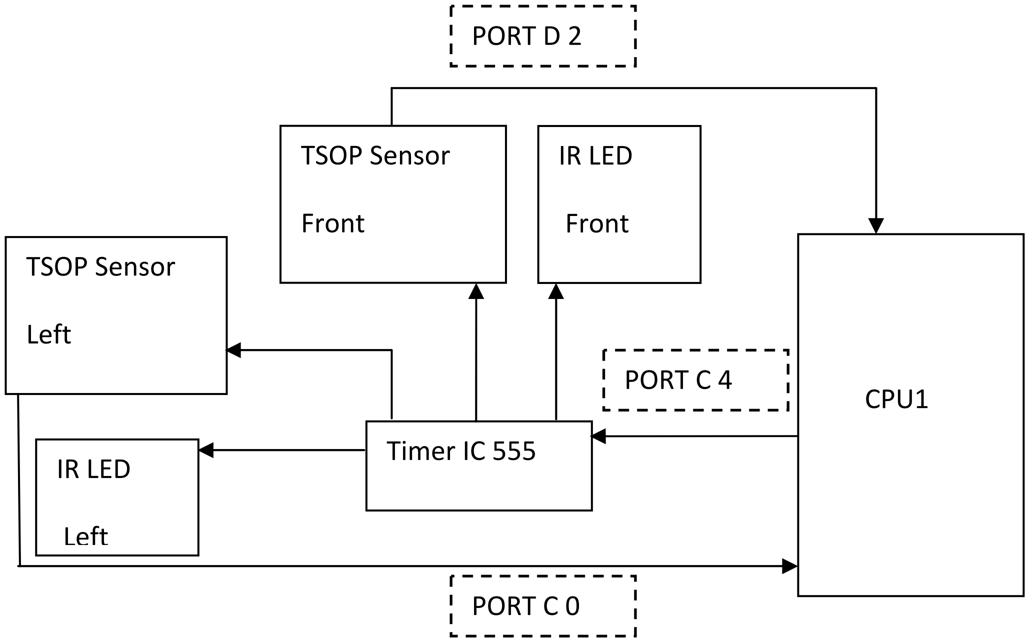


Figure 5.4 Sensor block diagram

The sensor block as can be seen in the block diagram above has two sensors and two IR LEDs, a timer IC LM555. The sensors are mounted such that they can tell the robot that there is an obstacle in the front or in the left or both. The IR LEDs keep radiating 36 KHz IR rays continuously. In case of an object in the robot's way, the object will reflect the rays radiated by the IR LED and the sensor will detect these rays through which it learns about the object in its way. This is conveyed to the microcontroller i.e. the CPU 1 and appropriate action is taken according to the mode the robot is in. The obstacle if sensed in the front the robot moves left and the left sensor also does its work. So the need of a right sensor is lessened significantly. The same could be done with the sensor in the right and the robot moving accordingly.

5.1.2.3 LINE TRACKER BLOCK

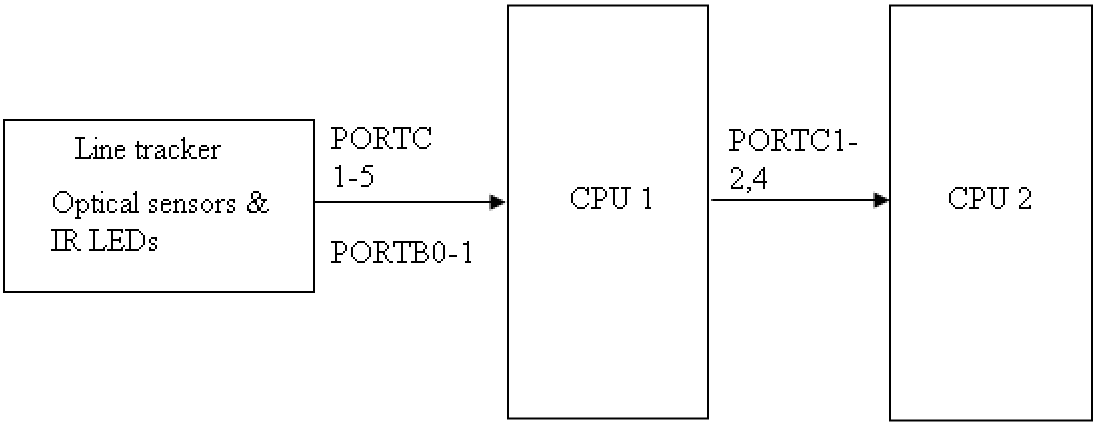


Figure 5.5 Line Tracker Block Diagram

Above shown figure 5.5 is the line tracker block of the robot. It has as shown an array of optical sensors and IR LEDs. The IR LEDs radiates IR rays when they get power supply and the optical sensors detect the light thrown by the objects they are exposed to. Also this block has indicators to show if there is an object in front or not. They glow in case of no object present or if a black object is present and they don’t if any other coloured object is present. Thus the sensors learn about the black path and this knowledge is conveyed to the CPU 2 of where the black path is. This in is sent to the CPU 1 which accordingly decides what to do and what not to do.

5.1.2.4 IR RREMOTE INTERFACE BLOCK

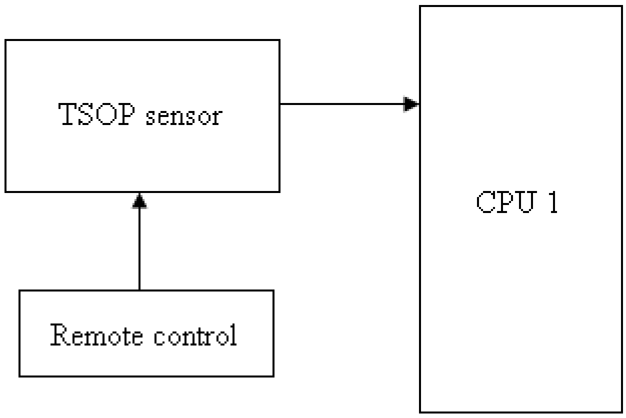


Figure 5.6 Remote control block

Shown in the figure 5.6 above is the remote control block of the system. The remote can be any audio/video TV remote used in daily life. The remote gives the commands to the robot as per the instructions in the user manual. These commands are caught by the sensor, which are then transferred to CPU 1. This then decodes the command and understands takes appropriate action.

5.1.2.5 DESIGN CRITERIA OF THE ROBOT

While designing the robot various things are to be kept in mind. Designing a robot requires balance between size (mostly weight), motor power and battery power. These three elements are connected with each other (more battery power increases the weight of the robot and requires stronger motors) and finding the "perfect" balance requires a lot of tweaking and experimenting. Using light materials brings down the weight significantly. Building the system frame out of a light and strong material would be a lot lighter than using metal plates, etc.

The electronics of a robot generally fall in 6 categories:

- Motor control: controls the movement of the motors, servos and such.
- Sensor reading: reads the sensors and provides this information to the controller.

- Communication: Provides a link between controller and an external PC, another robot, or a remote control.
- Controller: microcontroller board, processor board or logic board. This part makes decisions based on sensor input and the robot's program.
- Power management: parts that provide a fixed 5VDC, 12VDC or any other level coming from the batteries. Circuits that monitor the status of the batteries.
- Glue logic: Additional electronics that allow all the parts to be connected with each other. An example is a CMOS to TTL level converter.

Not all the robots use all the six categories mentioned above but generally a robot is a mixture of some or all of these.

THE ROBOT STRUCTURE

The robot is structured such that it can sustain any accidents which it is likely to meet with.

A wheeled robot was the best for our specifications and constraints. The robot in our case is four-wheeled. Of these four, two are powered wheels and two are unpowered. The powered wheels are used to drive the robot and the unpowered wheels are used for the balancing the robot by providing a point of contact with the ground. The balancing is achieved in a way that the center of mass will be concentrated between the wheels; also the battery, motors, etc. can be kept between the wheel spaces.

Wheels of a robot provide many advantages like:-

- Low cost
- Simple design and construction
- Near infinite different dimensions cater to your specific project
- Diameter, width, material, weight, tread, etc. can all be custom to our needs
- An excellent choice for beginners

It also has some disadvantages like:-

- Wheels may lose traction or may slip easily
- They have very small contact surface area

However the wheel width can be chosen keeping in mind the track of the robot. Wider wheels offer better stability but they experience more friction and hence decrease speed significantly. On the other side, thinner wheels are intended for better speeds, but they could be susceptible to slipping off the track. A rubber belt around the wheels can be used to decrease the friction and also to give a good grip to the wheels. This also makes the wheels heavy and is susceptible to slipping on wet surfaces. An alternate solution to rubber/leather grip is sponge grip which is very light in weight, give a good grip and is also better suited for rainy seasons. But the disadvantage it provides is that it gets dirty easily and also tends to exhaust the motor very soon. Hence, keeping in mind the track and certain other factors like speed, grip, friction, etc. the wheel diameter and width can be decided.

The robot uses belt drives to transfer the rotary motion between the two wheels. The motion of the driving wheel is, generally, transferred to the driven wheel via the friction between the belt and the wheel. Belt drives however have relatively high inspection and maintenance demands. The advantages of belt drives include:-

- Easy, flexible equipment design, as tolerances are not important.
- Isolation from shock and vibration between driver and driven system.
- Belt drives require no lubrication.
- Maintenance is relatively convenient

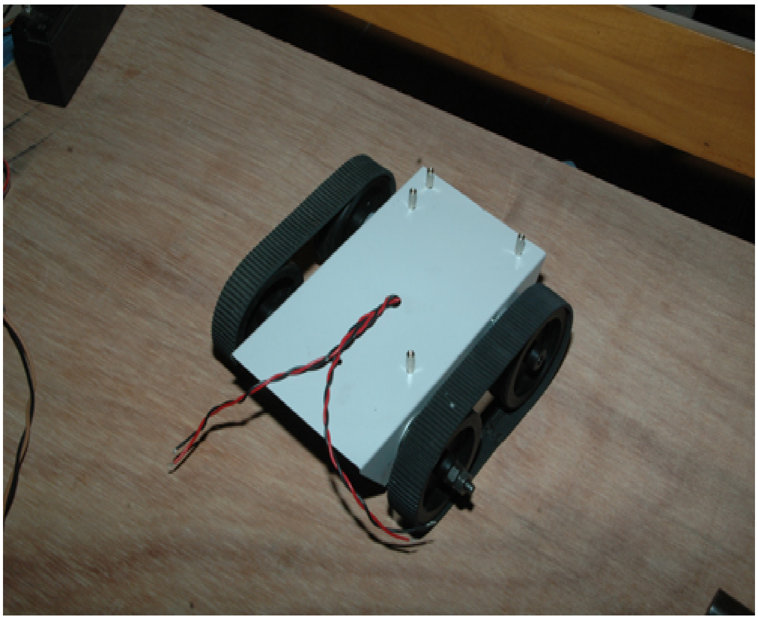


Figure 5.7 Robot Structure (insert the pic showing chassis n wheels)

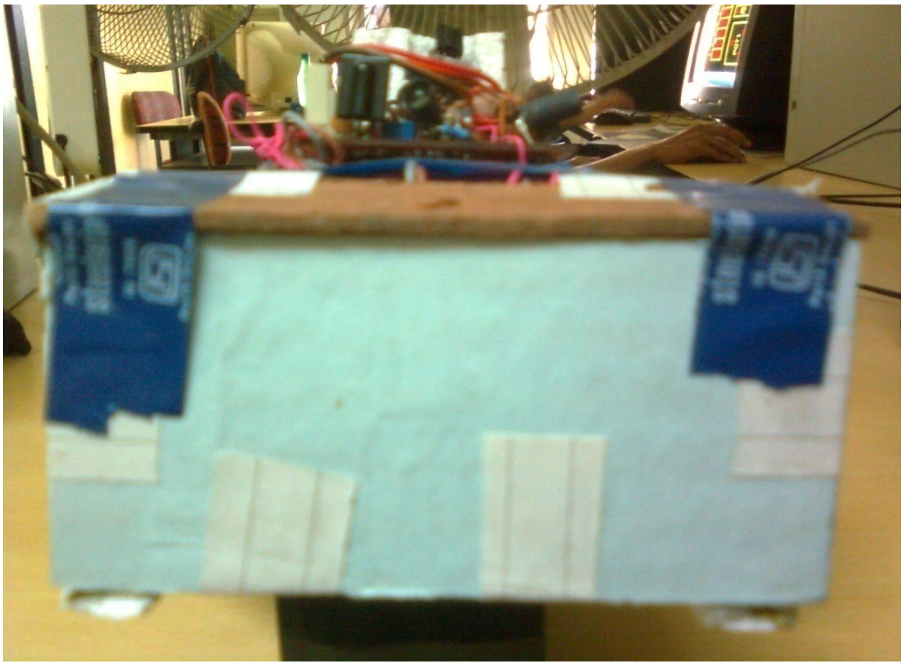


Figure 5.8 Box covering Line tracker sensors

Figure 5.9 Array of optical sensors of line tracker from box.

The chassis of the robot, as shown in the Figure 5.7 above, is made of mild steel and is strong enough to bear minor accidents that the robot might face. Also the hardware is mounted on the chassis with a hardboard base supporting it. All the blocks are mounted such that no other block or hardware is required to be disturbed to access the other. Also the entire hardware board can be separated from the chassis removing just two screws from the board. The line tracker module is packed in to a strong cardboard box like structure as shown in figure 5.8 and figure 5.9. This packing is mainly done so that external light rays don't affect the working of the sensors. In absence of the box, the external light affects the working in a way that the rays are so strong that the sensors instead of sensing the black path sense the external light and hence fail to detect the path. The packing is kept such that hardboard platform attached to the chassis protects it from the top and the line tracker box being a little inside the chassis and below the hard board. Thus it is made independent and also safe. These factors help making the design to be good in terms of accessibility and organization.

HARDWARE CONSIDERATIONS

- The system uses normal LEDs which use at least 15mA of current to glow. This effects the current consumption of the circuit as a whole to a significant extent. Instead, low-power (or simply dimmer) LEDs can be used. This drastically reduces how much current your circuits consume.
- The system uses CMOS ICs and not TTL. Using CMOS instead of TTL provides many advantages. CMOS ICs are voltage controlled transistors, unlike TTL which uses BJTs for logic operations whose working is current dependent, hence the over all power consumption is greater in TTL. CMOS, using voltage based transistors for logic operations, consumes much less power. CMOS ICs with reducing current usage, allow a more relaxed power supply. A variable power supply n a specific range will also work for CMOS ICs. CMOS are also smaller and cheaper but slower compared to TTL.
- The microcontroller here needs to work on its maximum speed grade to make it faster. The disadvantage is that more the clock of the microcontroller more is its current consumption.
- The IR LEDs used are 5mm diameter, with 45nm bandwidth. Infrared LED is commonly used in a wide variety of remote control and communications projects. Its wide 60 degree beam width and high output power make it a great infrared transmitter.
- Motors: - The motors used in the system are 100 rpm, reversible, 12 V and single phase. The specifications are as follows:

1. 100 rpm12V DC motor with Gearbox
2. 6mm shaft diameter with internal hole
3. 125 gm weight
4. 2 Kg cm torque
5. No-load current = 60mA, Load current = 300mA

The power required to drive the motor is called the motor load. No-load current is the current flowing through the motor under no load, i.e. the wheels are turning but no work is being done. And in case of work being done, the current flowing through motor a is called load current.

5.1.3 CONTROL THOERY

This decides the control to be used in the system. Techniques from control theory convert the tasks into commands that the system could understand and hence drive the actuators. There are different control techniques like PID control, Non-linear control, Fuzzy control, etc.

PID controller takes three values on which the processing is done. It first takes the Process value, estimates the error relative to the desired condition, and then processes it with the information it has and produces the output. Thus the controller always checks for error and responds to it in case of any.

Non-linear control means that in systems working with this type of controller, the output doesn't depend on input. Also the output in such systems depends on time, i.e. with time the output changes.

Fuzzy control is a control which takes a large range of values and then derives a logical representation of them in 0s and 1s.

Fuzzy control has three stages for processing input through output:-

1. Input stage – maps the sensor or other inputs such as switches, etc. to appropriate truth values and membership functions.
2. Process stage – invokes each appropriate rule and generates a result for each of them, then combines the result.
3. Output stage – converts combined result back to specific control output value.

Fuzzy logic has the advantage that the solution to the problem can be easily understood by anybody. It is well suited to low-cost implementations based on sensors, etc. This control theory will be used in our system in the future to implant Artificial intelligence in the robot. For example, there may be situations when the robot has to take a decision based on the previous track followed i.e. suppose the robot comes to a dead end while traversing a path, then it can use the path history to take a decision as to where to go next. If the robot has moved left for maximum number of times, it'll then decide to go left otherwise right and so on.

6. IMPLEMENTATION PLANNING

The right strategy for implementation of a system like Robot is very critical to make it successful over a period of time. The strategy should cover the people, processes, and the technology with broad view of the current status on these aspects in mind.

6.1 IMPLEMENTATION ENVIRONMENT

6.1.1 MICROCONTROLLER

The microcontroller used in our system is ATMEL ATMEGA8L. It has been used for all the jobs of the robot. Making the use of the interrupts and timers made the implementation easy in a way that priorities could be set easily and the system was made faster than normal because of its different speed grades. The microcontroller is programmed in such a way that it works as an obstacle avoider, obstacle follower, line tracker, along with all these an interface with the remote control.

The implementation of all these modules uses mainly timers and interrupts of the microcontroller.

Timers: - What are timers? Timers are a type of clock or timing devices through which we can control the sequence of events happening and which are going to occur. Timers are important because they help in maintaining sequence of events as mentioned before, also they can be used to time certain events so that an event can be serviced at a particular time and for a particular interval only and then the normal operation can be resumed.

Interrupts: - What are interrupts? Interrupt is a signal from either the hardware or software to indicate that a more important request has to be serviced. In case of an arrival of an interrupt, the normal execution of the process is stopped and the interrupt is serviced, after which the normal execution is resumed. Interrupts is a very important feature because these help the CPU to know about more important and urgent procedures to be serviced.

6.1.2 DISTRIBUTED CPU V/S SINGLE CPU ARCHITECTURE

We have used distributed CPU architecture in our system hence enabling multiprocessing. Two CPUs have been used to make the entire functioning of the robot possible. CPU1 works as an obstacle avoider and follower, also as an interface to the remote control. CPU2 works as a line tracker. This is kept in a different CPU so as to enable future enhancements and also that it can be used as an entire package which would make interfacing with other applications easier. The line tracker system sends its output to the main system i.e. CPU1 and this CPU then processes the input it gets and controls the system.

6.1.3 LANGUAGE USED AND WHY?

The language used in the system is C. A higher level language like this is used because it can interface hardware with software. Also in industrial applications like this using a higher level language, over assembly languages, has an advantage that it makes the code easier to understand and also the line of code and the effort needed in development decreases to a large extent.

6.1.4 SENSORS

There are two types of sensors used. For obstacle detection TSOP sensors, i.e. proximity sensors and for path tracking, optical sensors are used. Both these sensors are used along with IR LEDs so that the rays emitted by the LEDs are reflected by the objects or path and thus the sensors learn that there is something in its way which it has to be aware of. The proximity sensors have been used with a 555-timer IC. And the optical sensors are used with LEDs which will indicate an object sensed by glowing on and off. This LED reading is then recorded which will help in decision making as to what action is to be taken, when and how.

6.1.5 MOTORS

The system uses 100 rpm reversible motors. These motors have been used considering the cost and its rpm ratings both. But a motor with a higher rpm rating will certainly result in better working of the system. These motors are reversible, i.e. the motor direction can be changed as per needs. This mechanism proves helpful in the wheel drive of the robot hence making the controlling of the motion of the robot easy.

6.1.6 BATTERY

The battery used in the system is Lithium ion battery. This battery has a high capacity than the nickel-cadmium batteries in terms of weight and volume. Also it has a lower self-discharge and a faster charge. Also the battery packs are smart comparatively hence enabling a control on charger settings.

6.2 PROGRAM/MODULES SPECIFICATION

The system is divided into a number of modules making the system distributed and multiprocessing also. The different module into which the system is divided is:-

- Obstacle avoider: - In this module we have programmed in such a way that when obstacle is detected it must avoid the collision with obstacle .Thus it will sense the obstacle avoid it and travel safely. Thus, maintaining the safety of hardware.

  To make this software, the things needed were: -

  1. Sensor should firstly detect an obstacle.

  2. This detection should then be reported to the CPU.

  3. The CPU should take appropriate action.

  - Problems faced and how they were overcome: -

    A major issue faced is that the sensors used are not capable of detecting black objects. Hence the algorithm designed for obstacle detection assumes that there can be no black obstacle. To detect a black obstacle ultrasonic sensors can be used which will be done in the future enhancements. Ultrasonic sensors emit sound waves and decide the presence of an object and its distance based on the echo received back by it.

    Initially the algorithm was implemented in the mentioned order and the action the CPU would take was to move left in case of an obstacle in the front. And full supply was given to the 555 timer all the time the obstacle avoider was on. But this reduced the range of detection i.e. the distance from which the sensor could detect the obstacle was very less. To improve this range, we implemented the same strategy but with switching on and off the 555 timer at regular but short time intervals. This considerably improved the range of obstacle detection. Also because of switching the timer on and off alternately, there is a reduction in the current consumption, hence controlling the current consumption.

    Another problem faced with this module when it is integrated with the remote control module is that because the IR LEDs are on all the time, the rays emitted by these and the remote control signal both get mixed up and the obstacle is detected even when there is none. Hence the obstacle detector doesn't work as expected in that case.

- Line tracker: - The algorithm for this module is designed in such a way that when the robot is on path and in path follower mode it would detect the path and move on path until the end of it.

  The sub-goals needed to achieve this were: -

  1. Detect a black path, i.e. check if the rays emitted by the IR LEDs are reflected back or not. If they are not reflected then it is a black surface.

  2. Send this detection to the CPU 1.

3. Now decide if the robot is completely on the path or slightly off the track and hence which direction should the robot move on.

4. Send this decision to the CPU 2 which will in turn control the motor as per the direction needed.

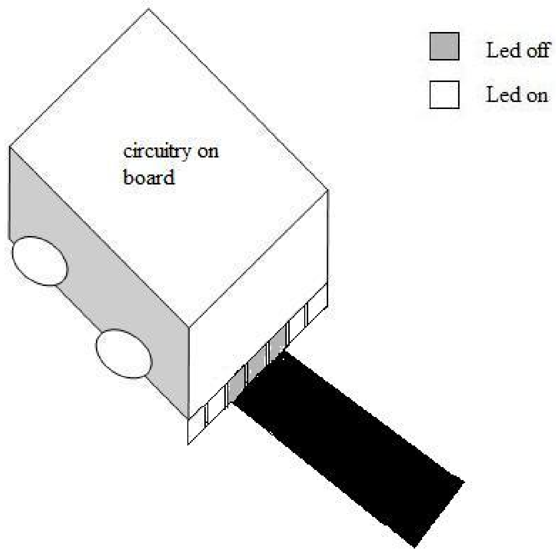


Figure 6.1 The robot moving forward

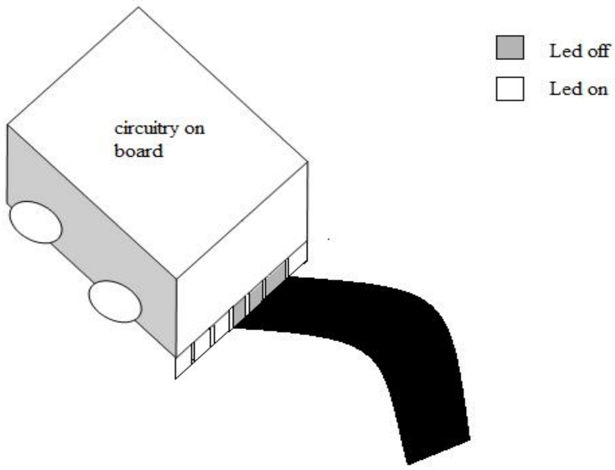


Figure 6.2 The robot moving left

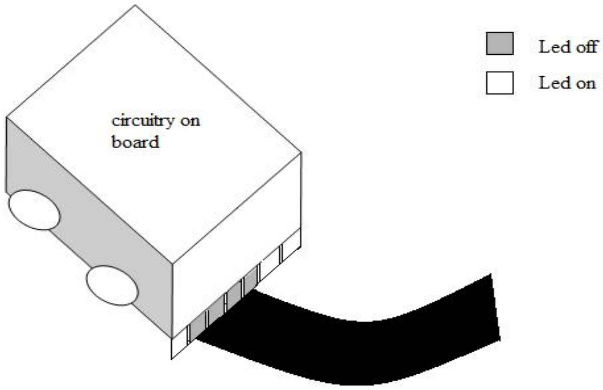


Figure 6.3 The robot taking a right turn

The figures shown above depict the reaction of the robot when it is on track and it goes out of track (left or right). The 7 LEDs who's on and off condition helps the line tracker in determining its position are shown by the boxes in the robot. In figure 6.1 the robot moves in the forward direction and the 3 LEDs in the centre are off through which the tracker gets to know its position and hence moves forward. Similarly for the robot to move right the extreme left led has to be on i.e. the fifth LED has to be on and vice versa for the left turn. These two conditions are depicted in the figures 6.2 and 6.3.

- Problems faced and their solution: -

  An important issue which the algorithm doesn't take into consideration is that the grooves on the floor also tend to absorb all the rays emitted by the LEDs and hence do not reflect any light and the sensors are sensitive to this also. The algorithm designed assumes the absence of such grooves on the floor.

Also the algorithm uses at least 2 sensors' input for the detection of every direction. Hence the error which can occur due to any one sensor not working properly is reduced and any such faulty detection will also not affect the decision.

The algorithm also can detect steep turns up to 30^0 from the horizontal. When there is a steep turn the robot does the following to know where it has to go as shown in the figure: -

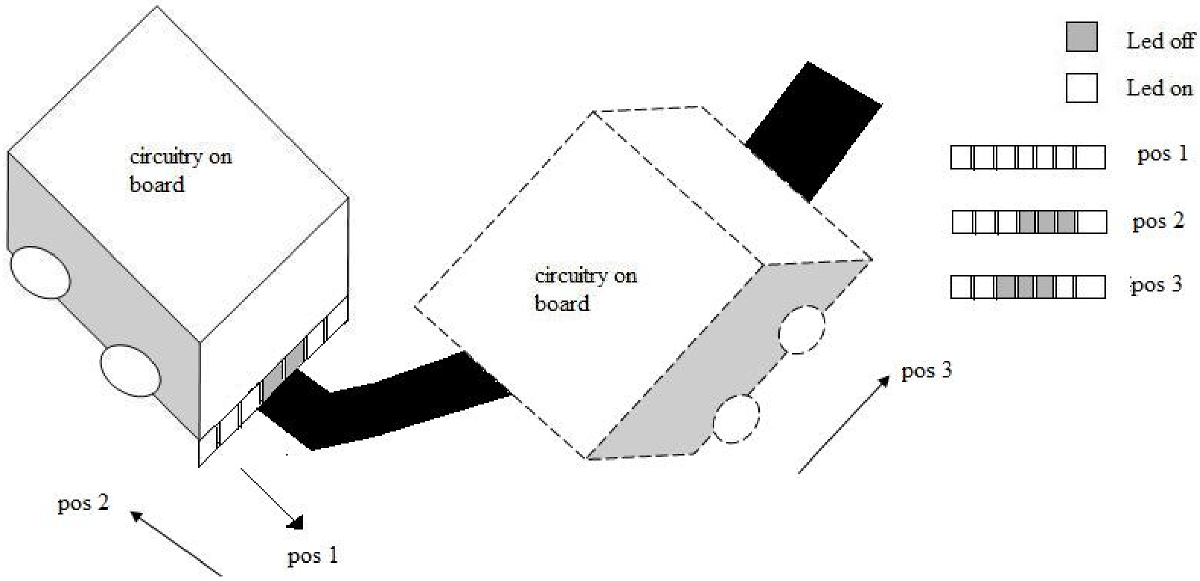


Figure 6.4 The robot taking a steep turn

- Stop at the beginning of the turn, thinking that it might be the end of the path.
- Go back a little and check if there is a path in any direction other than forward.
- If there is no path then stop. If there is then turn to that direction and repeat the entire procedure until it can't go forward or it can't find a path.

The above figure 6.4 is a pictorial representation of the algorithm for a steep turn of the line tracker. The robot is initially moving on its path. Pos 1, pos 2 and pos 3 indicate the different LED states of the line tracker during the algorithm. Once it encounters a steep turn, it goes to pos 1 first i.e. stops and then pos 2 i.e. goes back a bit. It then checks for a turn in any direction but forward. It finds a path in left direction hence the tracker's new state becomes pos 3 and it turns to the left as shown in figure and follows the path.

However this doesn't work for turns steeper than 30^0 from horizontal because the alignment of the sensors is such that they cannot detect such turns. A different logic will have to be implemented for detecting such turns which will be done in the future.

Also the robot can travel in varied speeds in this mode. It increases its speed when it has to take a turn so that it can try taking more steeper turns if necessary with more accuracy and when it has to move forward only, it travels in lesser speed comparatively so that it doesn't miss out any detection.

- Gripper: - This module is programmed to pick an object satisfying all its constraints and place it where ever required. The algorithm is designed such that the gripper first opens its arms and then tries to hold the object and pick it. Then it goes to the location where the object is to be placed and then places the object there and releases its arms. All this is done in hardware just by winding and unwinding of the thread links tied to the motor from the gripper.

When the gripper has to grip and pick the object the motor moves in one direction (winds up) and when it has to place and release, the motor moves in the opposite direction (unwinds itself).

- Problems faced and their solution: -

  A major issue here was that if the motor turns in a direction more than needed, then there is every chance that the thread links break and the gripper stops working. How does the robot come to know that it has lifted the object enough and the motor should stop running? To make this possible, a switch mechanism is used such that when the object is lifted and the motor has winded itself enough, a screw on the lifter platform presses the switch to tell that the motor should stop winding itself now and come to rest. Once the switch is released, the motor starts unwinding to place the object and then release it.

- IR Remote control: -This module is programmed to operate the robot from a distance. Thus providing more mobility to robot.

  The algorithm here takes the signals from the remote, decodes them and then according to the command of the remote control, takes appropriate action resulting in the robot following the commands by the remote control.

  - Problems faced and their solution: -

    A major problem in this designing this module is that there are a lot of proximity sensors mounted on the board. Which sensor is the remote sending signal to? And which sensor should be interpreting these signals? When integrated with the obstacle sensor mode, the remote sensor receives signals which are to be received by the obstacle sensors only and hence behaves in an unexpected manner.

    To overcome this situation, the IR LEDs which are emitting rays and because of which the remote and obstacle rays are getting mixed up are switched off by the software until the remote signal is coming in. This helps to avoid the unnecessary noise created by the mixing up of signals. Also the remote sensor is placed at more height from ground level as compared to the proximity sensors for obstacles to reduce some of the noise.

    This problem is mainly because both the signals i.e. the remote and obstacle signals are of 36 KHz. The frequency of remote signals can’t be changed because the remote control will send 36 KHz signal only but that of the rays emitted by the IR LEDs can be changed. So if their frequency is set to 58 KHz then the problem of mixing up of these signals can be easily overcome.

As the above specified module does not have any internal dependencies amongst each other so any kind of approach during programming can be followed.

- Speed grades: - An algorithm for different speed grades of the robot has also been designed. Hence the robot can travel in 4 different speed levels. This algorithm is based on the logic of switching the motor on and off. Doing this alternately for different time intervals increases and decreases the speed of the motor, i.e. the motors are switched on for a few microseconds and then switched off for (255-on time) microseconds. If the motor is kept on for all 255 counts and off for 0 counts then the motor runs with full speed. If the on time is reduced by certain amount then the motor will run a bit slower than the full speed. Similar strategy is used to achieve different speed levels.

6.3 CODING STANDARDS

Comments and description of functions, etc

All code -should be well commented (Developer etc). All procedures and functions should begin with a comment to explain what the function/procedure performs. A brief synopsis is sufficient.

The below is the Comment template for procedure and function:

```
***************DESCRIPTION OF FUNCTION******************************
Developer:
Entire Description
**********************************FUNCTION******************************
Purpose     : This function performs so on operation.
Input   : Input if any
Output : Output if any
****************************************************************************
```

All the functionality is also described in the program commented in such manner:-

```
**********************************FUNCTION******************************
TIMSK=_BV(TOIE0);     //Enabling Timer 0 interrupt
****************************************************************************
```

All the variables used in the program are given the meaningful name. The global variables are declared in the global region while all local variables used in particular function not required in another function are used locally.

Variable type: e.g.: volatile static int describes usage of the variable in interrupts.

The names of function are also meaningful according to task it does.

7. TESTING

Testing is the process carried out on system to detect the differences between its behavior and the desired behavior as stipulated by the requirements specifications. Testing is advantageous in several ways. Firstly, the defects found help in the process of making the system reliable. Secondly, even if the defects found are not corrected, testing gives an idea as to how reliable the system is. Thirdly, over time, the record of defects found reveals the most common kinds of defects, which can be used for developing appropriate preventive measures such as training, proper design and reviewing.

7.1 TESTING PLAN

The testing sub-process includes the following activities in a phase dependent manner:

a) Create Test Plans.
b) Create Test Specifications.
c) Review Test Plans and Test Specifications.
d) Conduct tests according to the Test Specifications, and log the defects.
e) Fix defects, if any.
f) When defects are fixed continue from activity.

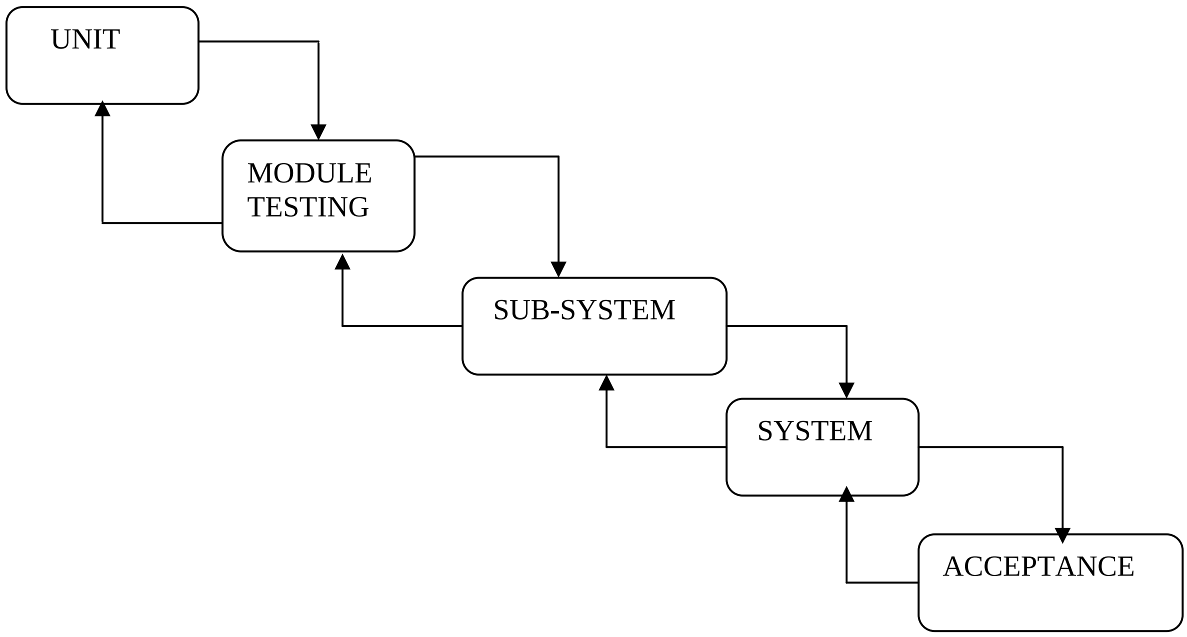


Fig 7.1.1 Testing Plan

7.2 TESTING STRATEGY

The development process repeats this testing sub-process a number of times for the following phases.

a) Unit Testing.
b) Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed in case of software and working of all hardware after it is used. Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct in software part and in hardware part how the hardware is integrated and how do they respond after integration to each other. System Testing ensures that the system meets its stated design specifications. Acceptance Testing is testing by the users to ascertain whether the system developed is a correct implementation of the Software Requirements Specification.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

We have performed both Unit Testing and System Testing to detect and fix errors. A brief description of both is given below.

Unit Testing

Objective

The objective of Unit Testing is to test a unit of code (program or set of programs) using the Unit Test Specifications, after coding is completed in case of software and working of all hardware after it is used. Since the testing will depend on the completeness and correctness of test specifications, it is important to subject these to quality and verification reviews.

Input: Unit Test Specifications

<u>Testing Process</u>

- Checking for availability of Code Walk-through reports which have documented the existence of and conformance to coding standards.
- Review of Unit Test Specifications

  Verify the Unit Test Specifications conform to the program specifications.

  Verify that all boundary and null data conditions are included.

7.3 TESTING METHODS

Black-box and White-box Testing

In black-box testing the system is viewed as a black box, without knowledge of its internal structure or behavior. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In white-box testing knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the system are traced. Hence more defects than black-box testing are likely to be found.

The disadvantages are that exhaustive path testing is infeasible and the logic might not conform to specification. Instrumentation techniques can be used to determine the structural system coverage in white box testing. For this purpose tools or compilers that can insert test probes into the programs can be used.

Code Coverage

The way to make sure that you have got all the control flow covered is to cover all the paths in the program during the testing (via white-box testing). This implies that both branches are exercised for an 'if' statement, all branches are exercised for a case statement, the loop is taken once or multiple times as well as ignored for a while statement, and all components of complicated logical expressions are exercised. This is called Path Testing. Branch Testing reports whether entire Boolean expression tested in control structures evaluated to both true and false.

Additionally it includes coverage of switch statement cases, exception handlers and interrupts handlers. Path testing includes branch testing as it considers all possible combination of individual branch conditions. A simpler version is Statement Testing which determines if each statement in the program has been executed at least once. The coverage via Path Testing includes the coverage via Statement Testing. Since Path Testing is extremely comprehensive it is costly, hence a viable minimum should be measuring Statement Testing coverage.

7.4 TEST CASES

Purpose	**Pre-Condition or Required input**	**Expected Result**	**Actual Result**	Pass/Fail
Check the working of the proximity sensors	An obstacle in front of the sensor.	The LED mounted on the board should glow as soon as the object is sensed.	The LED glows when the object is sensed.	Pass
Check the working of the robot.	Code the robot just to blink the LED mounted.	As soon as the robot is coded, the LED should start blinking.	The LED blinks.	Pass
Check the motors of the system.	Input to run the motors.	The motors should move as soon as they get the input.	The motors move.	Pass
Check the forward motion of the motors.	Program the robot with forward motion command.	The robot should move in the forward direction.	The robot moves forward.	Pass
Check the left of the motors.	Program the robot with left command.	The robot should move to the left.	The robot moves forward.	Pass
Check the right motion of the motors.	Program the robot with right command.	The robot should move to the right.	The robot moves right.	Pass

Check the ‘stop’ of the motors.	Program the robot with stop command.	The robot should stop.	The robot stops.	Pass
Check working of the forward proximity sensor.	Start the motors keeping them in the forward direction and keep an obstacle in the robot’s way.	The robot should change its direction as soon as the obstacle is sensed.	The robot moves left.	Pass
Check the working of the left proximity sensor.	Start the motors in the forward direction and keep an obstacle to the left of the robot.	The robot should change its direction as soon as the obstacle is sensed.	The robot moves right.	Pass
Check if the robot follows an obstacle	Set the motors moving in forward direction and the proximity sensors on.	Once an obstacle is sensed the robot should follow it.	The robot follows the obstacle as soon as it is sensed.	Pass
Check the working of the line tracker sensors.	Give supply to the sensors, and keep an object in front of it	The LEDs mounted with the sensors glow, and they respond by glowing in case of any object in front of it except a black colored one. In case of a black colored object, it doesn’t glow.	The LEDs glow when an object is placed in front of it.	Pass
Check if the	Integrate the sensor	The robot should start	The robot moves on the	

line tracker follows a black path.	with the motors and put the robot on a black path.	moving on the black path.	black path.	Pass
Check if the line tracker works for turns steeper than 30º from the horizontal.	There should be a steep turn in the path being traversed.	The robot should take the turn.	The robot stops at the steep turn.	Fail
Check working of the gripper (dummy model).	Winding and unwinding of the thread links tied to it.	When the thread is winding, the gripper arms should close and get lifted up. When the thread is unwounded, the gripper arms should come down and open.	The gripper arms close and lift up when winded, and come down and open up when unwounded.	Pass
Check working of the gripper (actual model).	Winding and unwinding of the thread links tied to it through the motor.	When the thread is winding, i.e. motor rotates in a direction, the gripper arms should close and get lifted up. When the thread is unwounded i.e. motor moves in the opposite direction, the	The gripper arms close and lift up when winded, and come down and open up when unwounded.	Pass

		gripper arms should come down and open.		
Check if the robot responds to the remote control.	Any button should be pressed through the remote.	The LED mounted on the robot should glow.	The LED glows.	Pass
Check if the robot interprets the command through remote control.	Any button pressed of the remote (0-9).	Four LEDs on the test board glow as per the binary conversion of the number pressed on the remote.	The LEDs glow as per the binary conversion of the number pressed.	Pass
Check if the robot listens to the commands given by the robot.	Any button pressed for a particular functionality.	The robot should perform the said functionality.	The robot does what the remote says.	Pass

Table 7.1 Test Cases

8. USER MANUALS

The figure of the end structure of robot is shown below. This structure is used to do operations like obstacle avoider, Line tracker, and IR remote.

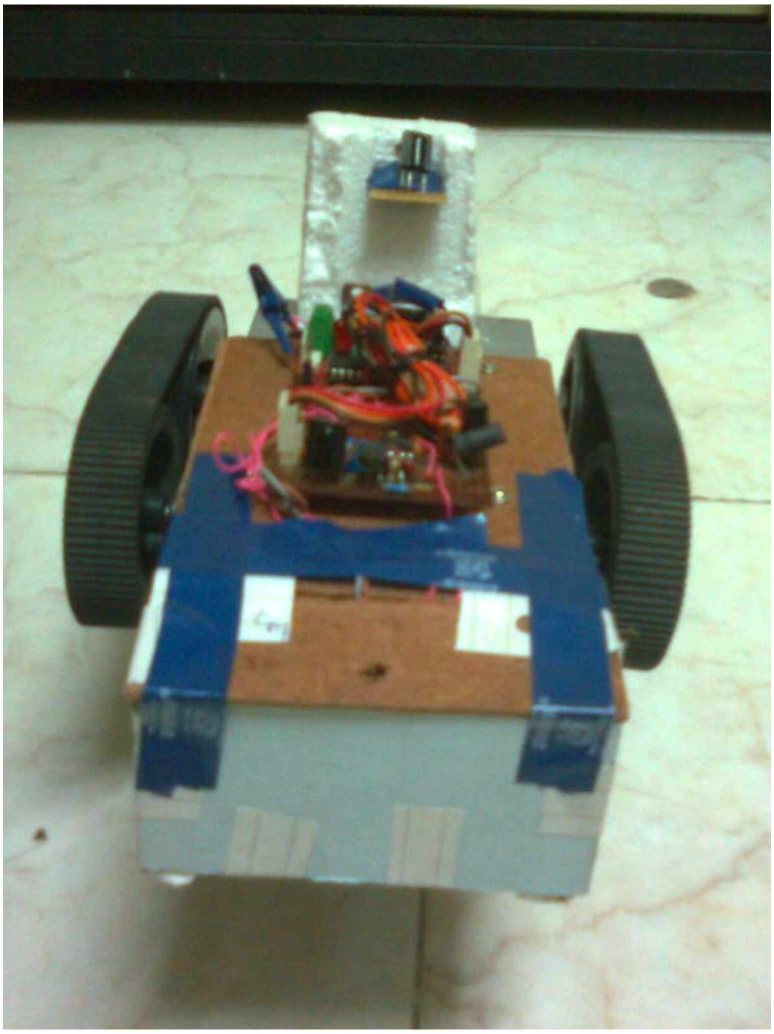


Figure 8.1 The Robot

The various other functionality performed by it is shown in below figures and the complete manual to operate it is shown below.

Figure 8.2 Remote control to operate robot.

The RC5 remote control is used to control the robot. We have used the above remote to control it. Any RC5 remote will be able to control the robot.

Here the different buttons of remote control are used to follow different operations.

- Button 1 to 4 controls the speeding of robot where 1 is the highest speed and 4 is the lowest speed.
- Button 5 is used to navigate with IR remote which is set as default mode.
- Button 6 is used to navigate on black track
- Button 7 is used to enable obstacle avoider mode.
- Button 0 is used to enable robot as in grabber i.e. to enable gripper for pick and place mode.

- Button PR+, PR-, Vol+ and Vol- are used to navigate robot in IR remote mode in forward, backward, right and left directions respectively.

  To enter any mode the following task are to be performed:

  For e.g. Press 7 to set obstacle avoider mode and then Press 5 to return to IR-navigate.

The below figures gives the idea of robot working in various modes:-

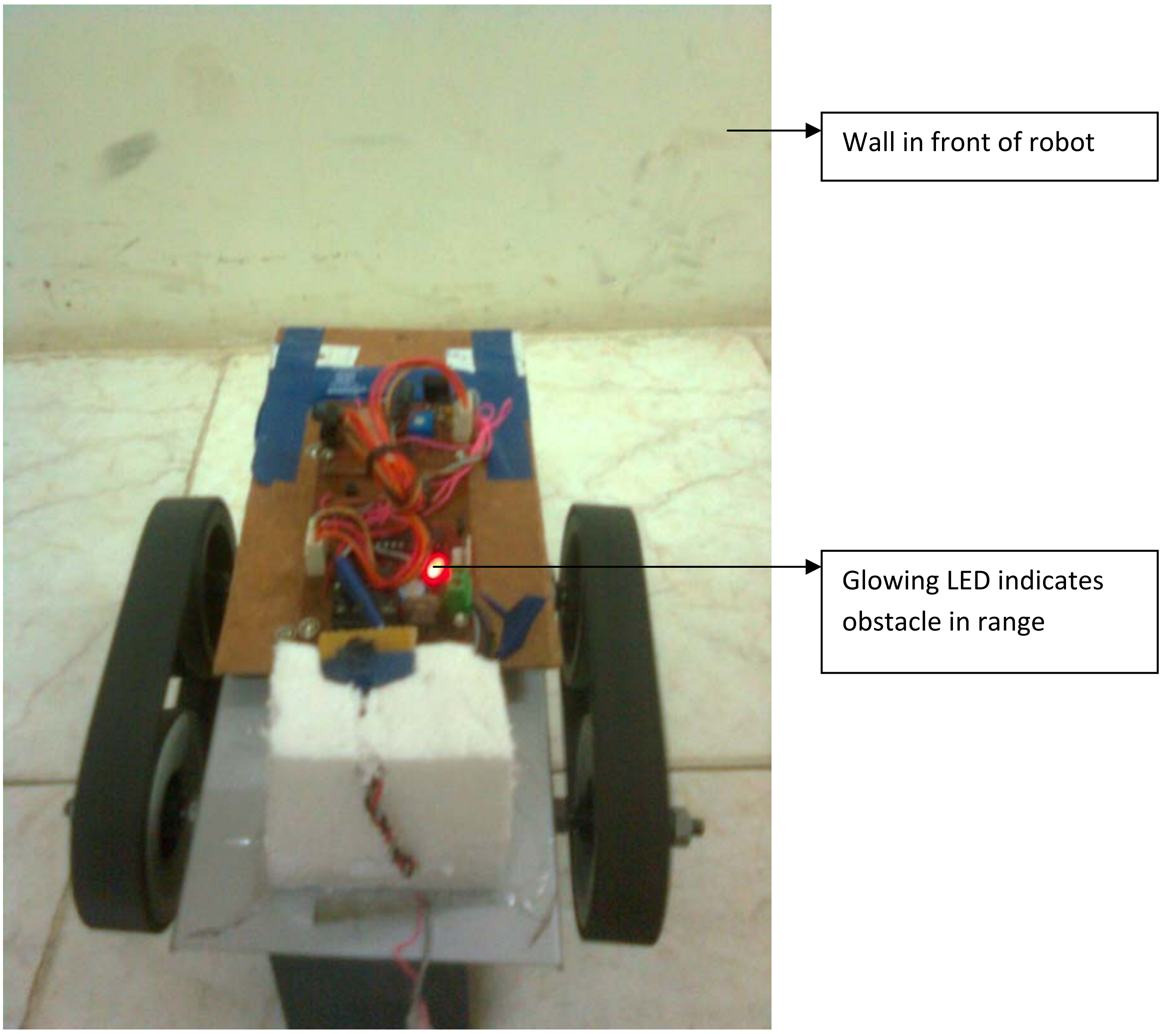


Figure 8.3 Obstacle avoider mode.

Figure 8.4 Black Path on which robot moves

When pressed 6 Robot moves on line tracker mode.

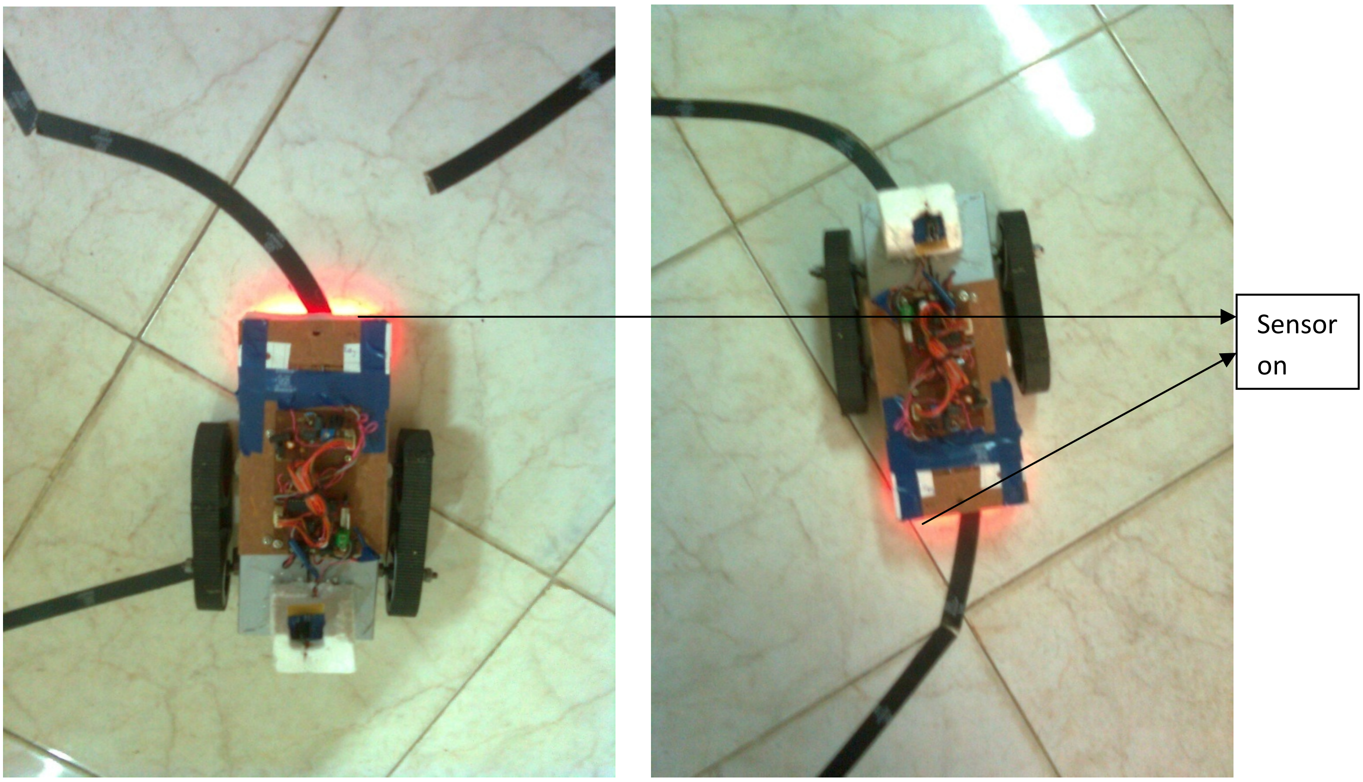


Figure 8.5 Robot following the black track.

Figure 8.6: Robot working in Grabber i.e. gripper for pick and place

When pressed 0 near the object to be grabbed robot moves in grabber mode where only gripper arms will work with PR+ and PR- to grip object lift up and down.

9. LIMITATIONS AND ENHANCEMENTS

9.1 LIMITATIONS

The System is designed in such a way to give the best possible throughput in the given time and cost. Even though there are some limitations which can be improved in end product.

The system we have built is prototype model of the final system to be built and so has the limitation in the strength and material used to build it. The system could have the better output if more advanced CPU could be used with high speed grades so that working could be made more speedily and thus increasing overall throughput of system. For e.g.:- We have used ATMEGA 8L CPU in our system which has maximum speed grade of 8 MHz but if we use the ATMEGA 8 CPU it has maximum speed grade of 16 MHz which would almost double the internal process speed and throughput can be increased.

The CPU we have used is 28 pin IC so if we can use the CPU with more pins which can be used by developer than more functionality could have been incorporated in more efficient way.

Now talking of strength of the system it is made to survive normal collisions but it can be made of material stronger if budget amount can be adjusted. The battery used can also be of improved quality which can last longer so that system can work for longer time and equally safe. But in the given budget this is the best possible battery which can be used to drive such a multifunctional system with keeping safety issue in mind.

The system mechanics is currently accessible and can be disturbed. As this is the prototype of the original system to be build so enough steps are not taken to ensure that the system does not get disturbed.

The system software is designed to get out of most of the erroneous situation but if the time spend on software side could be increased this could get better.

This system can be operated by a novice user but only after he is familiar with the complete user manual. The novice user will take some time to adjust to system but after getting adjusted the user can operate it easily.

Thus, this all limitation can be overcome upon in future end product design and implementation.

9.2 ENHANCEMENTS

- Bluetooth instead IR

We have used IR remote to operate the robot from distant place. But due to usage of IR remote the distance from the place it can be operated becomes limited. So in future we would use the Bluetooth device to interface it with distance device so that the mobility range of robot can increase.

- Gripper arms to carry heavy load and its future implementation

As a prototype we have used gripper arms made of hardboard which can also carry much load but if other stronger material is used, even more load can be carried. The material used in designing of entire gripper is done to check the functioning. Now that the functionality is performed, more time would be spent on making it stronger.

The arms of the gripper are made of hardboard and hence pose chances of getting bent or broken in case the gripper lifts more load than its capacity. Instead, a material like fiberglass would be used next, in the enhanced version, to make the entire structure more powerful than it is. Steel is also an option but it increases the load on the system along with the strength it provides, resulting in the reduced efficiency of the motors, etc. In comparison, fibreglass is light in weight and has a very high weight to surface area ratio making it one of the best and strongest materials which can be used.

Applications where the gripper can be used:-

The concept of the proposed gripper is applicable to more professional applications like:-

1. Pick and place of household articles.
2. A demonstration robot is created using a gripper to assist disabled people to navigate the robot and fetch distant objects and carry them when and where needed, using a remote control.
3. Gripper can also be used as a baggage carrier at places like airports, railway stations, etc. to carry any kind of luggage from place to place. Using a robotic gripper makes it easier and more secure to do the job.
4. Today, when terror is at its peak, the use of man power to dispose any kind of explosives is a threat to the society. Instead a device like a robotic gripper can be used to locate, pick and dispose the explosives causing no harm to the society making it a safer, more powerful and hence definitely a better option.
5. In view of the simplicity and low cost, it is possible to use multiple robots with gripper to perform a single complex job, under integral control of AI control software.

Thus, it has a huge scope as it is economical, safer and also doesn’t disturb the social environment.

- Increased power motors and battery
  The motors used cannot take much load and slows down its speed so motors which can take more loads will be used and battery too of better quality would be used.

- Better CPU
  The CPU used in current system has much limitation such as internal speed grades and port pins for usage. Better CPU would enhance the throughput of the entire system.

- Security
  The major concern was the systems internal as well as external safety and security it provides. So, appropriate steps should to be taken to enhance its safety and security issues, because the end product would be used by different kinds of people, hence increasing the need for the system to be more safe and secure.

  The devices, ICs used in the current system are used keeping security issues in mind. But the external security can be improved as well as safety measures can also be improved so that external factors do not affect it.

  Also the system would use ultrasonic sensors to detect black colored objects, which is a major drawback in the present system.

- Less user dependency
  The system should be as mobile as possible. The robot should be programmed in such a manner that its dependency can be reduced and thus the efficiency is increased.

- Automatic charging station
  The charging station is to be build so that when battery power goes low it automatically can detect it and take useful measures to ensure that robot does not breakdown and automatically finds the charging station and docks itself there. After complete charging it gets back to its work.

  In figures 9.1 and 9.2 shown here is a design of the automatic charging station which will be implemented in the future. The robot enters the charging region when its charge goes below the necessity level. Once the robot enters the charging station, it has various paths leading to the docking point where it can dock itself to get charged. After it is completely charged, it goes back on the path from which it entered gradually opening its arms.

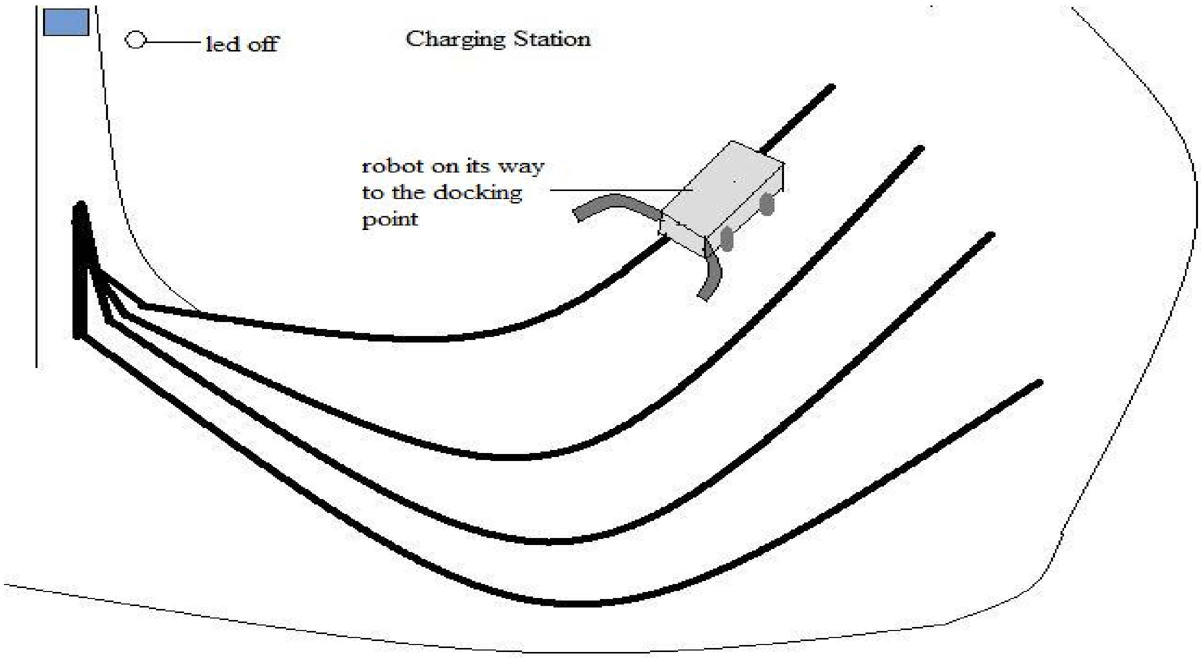


Figure 9.1 Automatic charging – The robot on the tracking to the docking point

Figure 9.1 shows the robot after it has entered into the charging station. As it can be seen in the figure, the charging station is a large area with various paths leading to the docking point. The paths are made in such a way that the robot can easily find one as soon as possible. These paths are made to converge once the robot nears its destination. Also the gripper's arms help detect the docking point once it has entered the zone. Initially the gripper arms are completely open. But once the robot enters the charging region, which is a path narrowing gradually near the docking point, the arms start closing themselves and when they completely get closed the robot realizes that it has reached its destination and now it's time to dock itself and get charged up.

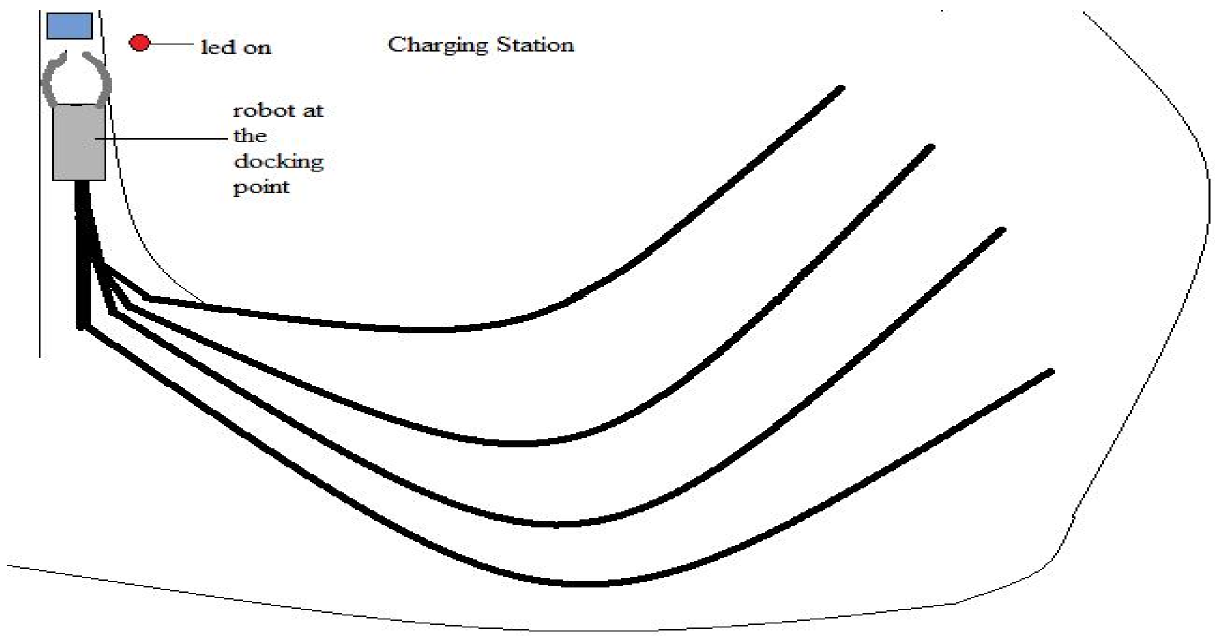


Figure 9.2 Automatic charging – The robot in the docking point

The figure 9.2 shows the robot travelling to the docking point which is a small box at the top-left corner of the figure. The led shown beside it indicates if the robot has started charging or not. Here we can see that the robot has reached the docking point and started charging itself which is indicated by the gripper’s closed arms and the led glowing. This is how the automatic charging will be achieved in the enhanced system.

- Flash card interface
  The flash card is to be interfaced with the robot working so that it can save the location of each and every point where it goes and thus also perform history storage which will guide robot where to go and where it is.

10. CONCLUSION AND DISCUSSION

10.1 CONCLUSION

- This model is the prototype model of the original system.
- This system requires development of both hardware and software efficiently.
- **Gripper** - The gripper would enable pick and place of objects which when professionally developed can be advantageous in different fields which would have business scope as well.
- **Interfacing with Remote Control** – This would allow user to interface with system from distant places and can operate from good distance.
- **Obstacle detecting sensors** – This sensors are economical and very useful these can be used in various areas such as to avoid an obstacle, to show that there is obstacle or to follow the dynamic objects. Thus has wider business application.
- **Line Tracker –** This technology can work as a dedicated path to various functionality it performs.

10.2 DISCUSSION

The project if enhanced for more professional usage it can be used in various concerning regions like security services, as a house hold help,

10.2.1 SELF ANALYSIS OF PROJECT VIABILITIES

and in professional institutes for various other works. This can be done at very low cost using this project with greater efficiency. Thus, making this project a great help for society.

10.2.2 PROBLEM ENCOUNTERED AND POSSIBLE SOLUTIONS

- First problem we faced in understanding project work was the hardware which included microcontroller, various ICs which we were unaware of. During the period of training we studied different hardware and their working. We conducted various experiments to become familiar with their working.
- The second problem encountered was designing of gripper which was mechanical task and to make it cost efficient we needed to use the material which was not that strong. But finally we were able to make it with appropriate guidance.

10.2.3 SUMMARY OF THE PROJECT

Throughout the project, we enjoyed working on the project because it was innovative and great learning experience. This project is the prototype of the final end product to be built and keeping in mind the vision and scope of the final project it was worth working for.

When we worked upon this project there were many difficulties faced by us but under the appropriate guidance we were able to complete it in allotted time. The objective of project was to achieve the task such as gripper for pick and place, obstacle avoider, path finder etc which were successfully accomplished.

APPENDIX C

Name of external Guide	**Dr. H.S. Mazumdar**
Designation	**Head, R & D Center**
Email Id of External Guide	**hsmazumdar@hotmail.com**
Contact Number of external Guide (mobile)	**097255-34521**
Name of company	**R & D Center, DDU**
Contact Number of Company	**0268-2522792**
Address of Company	**R & D Center, DDU, Nadiad**

REFERENCES

Websites Referred

1. www.wikipedia.org
2. www.solarbotic.net
3. www.wikianswers.com
4. www.vegarobokit.com
5. www.robokits.co.in
6. www.atmel.com
7. www.protoss.ru

EXPERIENCE

SUMMARY OF EXPERIENCE DURING THE PROJECT

Throughout the process, we enjoyed working with the team and it was a great learning experience. We had the wonderful experience which would enable us to understand hardware more easily and efficiently.

Working with a globally renowned person, was a great learning experience to work according to his standards and application areas. This experience has widened our horizons of knowledge to a very large extent. Till now we were exposed only to the software side of learning but with this project we have attained hardware learning from very basic level, we got aware of the current powerful technology available in the market and their behaviour. With this project we have attained a lot of understanding from the practical point of view also.

This immeasurable knowledge gaining experience for us has already sown its seeds into our career and we hope to carry it with us throughout our professional life.

Working under his experience has not only helped us developing as a programmer or designer but as an individual by letting us realize our potential and believing in our capabilities and hence boosting our confidence.

This project will always remain an integral part of our resume and we hope our wonderful experience here would provide a boost in our career as well.