# SELF GUIDED ROBOT

*PROJECT*

*Submitted in partial fulfillment*

*of the requirements for the degree of*

BACHELOR OF TECHNOLOGY
(ECE)

*by*

*ABHISHEK JEE – 071012*

*AKHILESH SHARMA -071096*

*Under the Supervision of*
*Mr.MUNISH SOOD*



***JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY***

***WAKNAGHAT***

***SOLAN, HIMACHAL PRADESH***

***INDIA***

*2011*

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY**

**WAKNAGHAT**

**SOLAN , HIMACHAL PRADESH**

**Date:**

## CERTIFICATE

This is to certify that the thesis entitled **SELF GUIDED ROBOT** is submitted in the partial fulfillment of the award of degree of Bachelor of Technology (E.C.E.) by **JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT.**

Signature in full of Supervisor: ____________________

Name in Capital block letters: **Mr. MUNISH SOOD**

Designation: **Lecturer**

# ACKNOWLEDGEMENT

No venture can be completed without the blessing of Almighty.We consider it our bounded duty to bow to Almighty whose kind blessings always inspire us on the right path of life This project is our combined effort and realizes the potential of team and gives us a chance to work in co-ordination.

Science has caused many frontiers so has human efforts towards human research. Our revered guide **Mr. MUNISH SOOD,** Lecturer, Department of ELECTRONICS AND COMMUNICATION, JUIT, has indeed acted as a light house showing us the need of sustained effort in the field of robotics to learn more and more. So we take this opportunity to thank him, for lending us stimulating suggestions, innovative quality guidance and creative thinking. He provides us the kind of strategies required for the completion of a task. We are grateful to him for the support, he provided us in doing thinks at our pace and for being patient for our mistakes.

# ABSTRACT


The project entitled as **SELF GUIDED ROBOT** is a system developed for detecting and avoding obstacles in its path based on the programming.

The basic thinking of our project is inspired by the thought of creating a robot which can be capable of recognizing and maybe eliminate or avoid the obstacle in front and guide itself in hostile environments whose applications in real world maybe of very much need of the hour.

- Support for a large number of works
- Fast, easy deployment
- Security
- Low cost
- Ability to be used globally.

*In short :Robots* will enable *us* to work in hostile environments where we could never would have done before.



The basic thinking of our project is inspired by the thought of creating a robot which can be capable of recognizing and maybe eliminate or avoid the obstacle in front and guide itself in hostile environments whose applications in real world maybe of very much need of the hour.

## TABLE OF CONTENTS

# LIST OF FIGURES

# 1. INTRODUCTION

Robotics, computer-controlled machine that is programmed to move, manipulate objects, and accomplish work while interacting with its environment. Robots are able to perform repetitive tasks more quickly, cheaply, and accurately than humans. The term robot originates from the Czech word robota, meaning compulsory labor.

## What is robotics?

Robot is a system with a mechanical body using computer as its brain. Integrating the sensors and actuators built into the mechanical body, the motions are realized with the computer software to execute the desired tasks

To begin with the electronic components such as microcomputers, and motors and other mechanical components etc., recently there are much remarkable and staring advancement in the fundamental technologies of robotics.

## Why robotics??

- Robots today can be used for purposes which may be beyond human capabilities. We can use robots in such fields in case of areas where conditions which support human life are not there
- In space
- There are a lot of desired environment that the human body and sensory organs are Incapacitated to reach.
- In areas with nuclear contamination

## 1.1 SCOPE

Robots today can be used for purposes which may be beyond human capabilities. We can use robots in such fields in case of areas where conditions which support human life are not there. Robots will enable *us* to work in hostile environments where we could never would have done before.

## 1.2 PROBLEM STATEMENT

To develop a system that is able to be controlled through a Bluetooth device using dtmf device and also in case a communication loss is dere it itself is able to move or control its motion in hostile environment with the help of a reflector circuit and successively performs any function.It can be used to eradicate the problems in any hostile environment which may be lethal for human's. It may also be able to suppres any kind of faults in these places so as to avoid any future damages.

## 1.3 METHODOLOGY

We have created a robot whose step wise description Is mentioned below:

**Step 1:**

A mechanical robot with installed motors for any desired motion.

**Step 2:**

A DTMF device through which it can be controlled wirelessly.

**Step 3:**

An reflector circuit is installed in the device so that in case of communication breakdown the robot guides itself through the hostile environment.

**Step 4:**

For step 1-3 the 8052 device has to be burned and programmed accordingly.

**Step 5:**

In other cases like recovery in case of turn over of the robot has be also be kept in mind.

# 2. HISTORY

In the early 1800's mechanical puppets were first built in Europe, just for entertainment value. And these were called robots since there parts were driven by linkage and cams and controlled by rotating drum selectors. In 1801 Joseph Maria Jacquard made the next great change and invented the automatic draw loom. The draw loom would punch cards and was used to control the lifting of thread in fabric factories. This was the first to be able to store a program and control a machine. After that there were many small changes in robotics but we were slowly moving forward.
(UK Robot. 2001)

The first industrial robots were Unimates developed by George Devol and Joe Engelberger in the late 50's and early 60's. The first patents we by Devol but Engelberger formed Unimation which was the first market robots. So Engelberger has been called the "father of robotics". For a while the economic viability of these robots proved disastrous and thing slowed down for robotics. But the industry recovered and by the mid-80's robotics was back on track.
(Dowling, Kevin. 1996 )

George Devol Jr, in 1954 developed the multijointed artificial arm which leads to the modern robots. But mechanical engineer Victor Scheinman, developed the truly flexible arm know as the Programmable Universal Manipulation Arm (PUMA).
(MSN Leaning & Research. 2002 )

In 1950 Isaac Asimov came up with laws for robots and these were:

1. A robot may not injure a human being, or through inaction allow a human being to come to harm.
2. A robot must obey the orders given it by human beings, except where such orders would conflict with the first law.

3. A robot must protect its own existence as long as such protection does not conflict with the first or second law.
(Robotics Introduction. 2001.)

Mobile Robotics moved into its own in 1983 when Odetics introduced this six-legged vehicle which was capable of climbing over objects. This robot could lift over 5.6 times its own weight parked and 2.3 times it weight moving.
(UK Robotics 2001)

# 3. BLOCK DIAGRAM AND DIGITAL CIRCUIT BOARD DESIGN

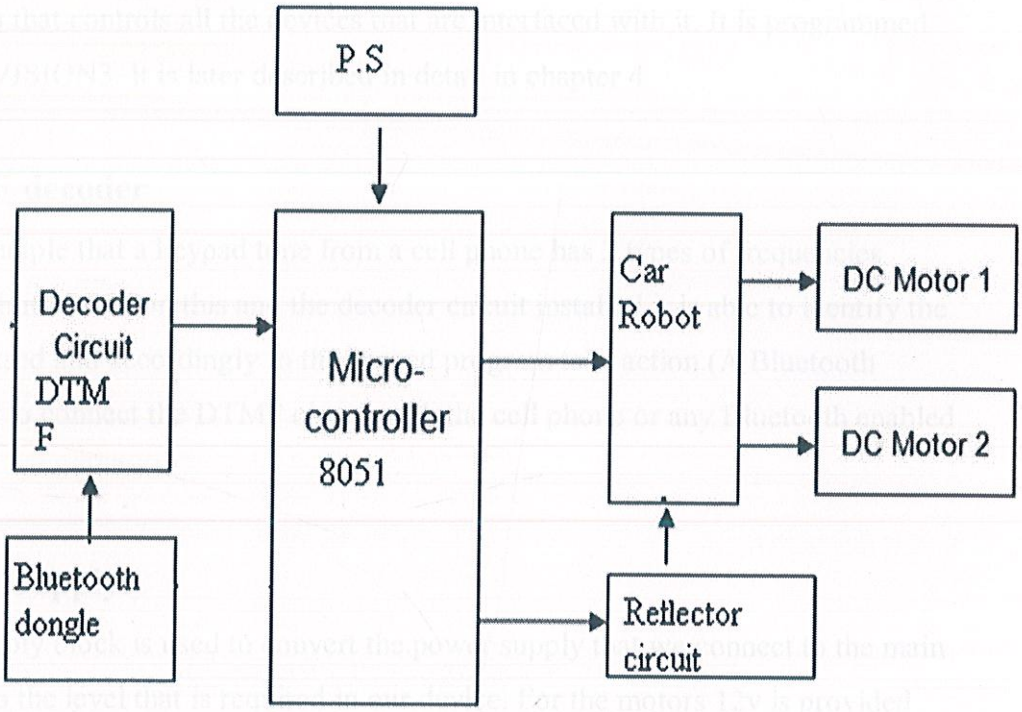



Fig 3.1 Block Diagram of the circuit.

Our project can basically be divided in these systems interfaced with each other so as to be able to perform the desired operations.

Some of the components mentioned above have following functions :

### 3.1.1 Microcontroller 8051

The main chip that controls all the devices that are interfaced with it. It is programmed with KEIL-μVISION3. It is later described in detail in chapter 4.

### 3.1.2 DTMF decoder

It uses the principle that a keypad tone from a cell phone has 2 types of frequencies associated with it .Based on this and the decoder circuit installed it is able to identify the key being pressed and accordingly to the burned program take action.(A Bluetooth dongle is used to connect the DTMF circuit with the cell phone or any Bluetooth enabled device)

### 3.1.3 Power Supply:

The power supply block is used to convert the power supply that we connect to the main circuit board to the level that is required in our device. For the motors 12v is provided and for the main chip 8051 a regulated supply of 5v is given.

### 3.1.4 Car Robot.

We have created a car robot using a circularly cut cardboard which has 2 main wheels connected to the motors and a control wheel to give direction to its motion.

### 3.1.5 Reflector Circuit.

It consists of a photodiode and a LED. The latter one is used to emit the infrared rays so as to detect the obstacles and the previous one absorbs the reflected IR ray and informs the MCU about the obstacle ahead and the MCU takes action accordingly to the program.

All of the above mentioned components are discussed in detail in later chapters ahead.

## 3.2 CIRCUIT DIAGRAM

The circuit diagram made in fig was used to install the different components of the device accordingly to the pins allotted for different components. Different ports were allotted different individual circuits we have used to make our robot.
In this Fig we can easily fabricate the various different components like the power supply diagram ,the DTMF decoding circuit ,the motors ,LCD interfacing etc.

Interfacing the devices with the ports of MCU:
Port 0 is used to connect the LCD to the device.
Port 1 is used to connect the DTMF device to the MCU
Port 2 is used to connect the motors to the MCU
Port 3 is used to connect to the MCU, the power supply of the system.

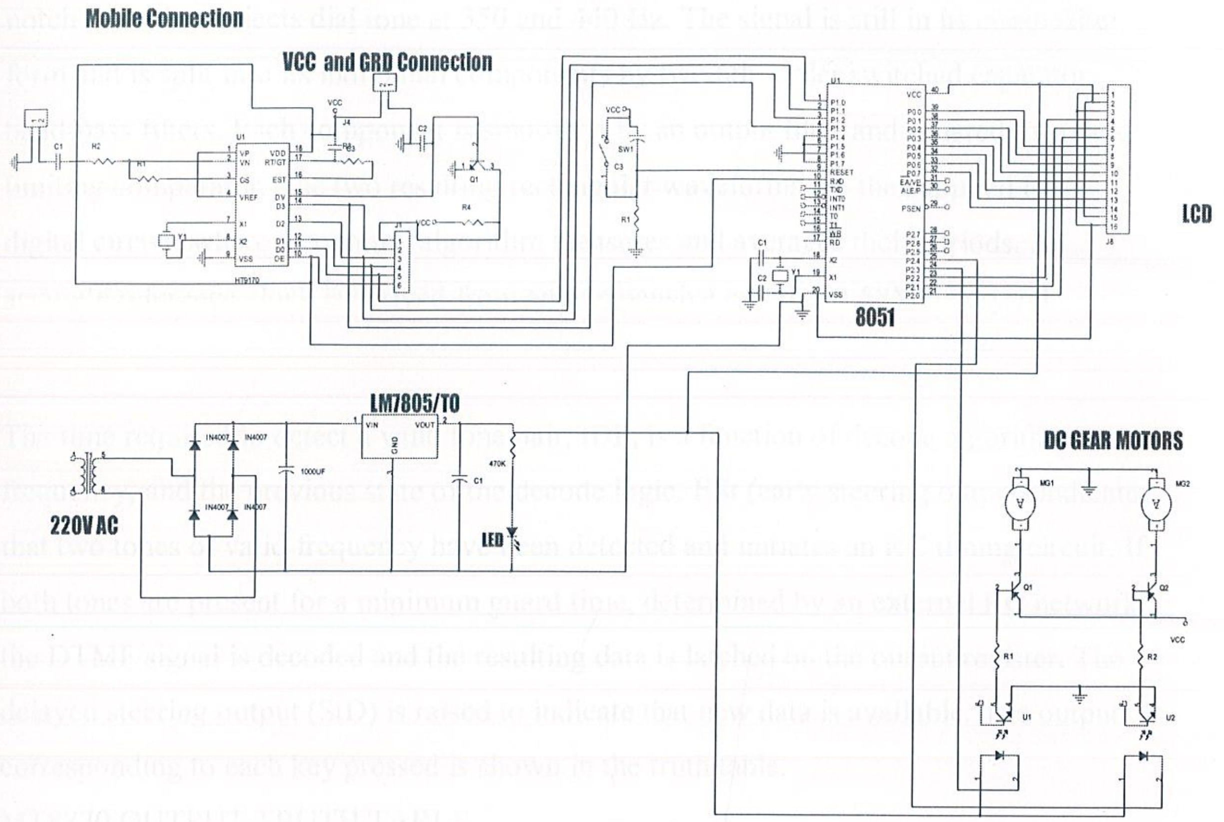



Fig 3.2-cicuit board design for the device

## 3.3 DTMF Decoder:

The MT8870 is a single-chip DTMF receiver incorporating switched capacitor filter technology and an advanced digital counting/averaging algorithm for period measurement. The functional block diagram of Fig. 4 depicts the internal working of this device.

The DTMF signal is first buffered by an input op-amp that allows adjustment of gain and choice of input configuration. The input stage is followed by a low-pass RC active filter, which performs anti-aliasing function. A third-order switched capacitor notch filter then rejects dial tone at 350 and 440 Hz. The signal is still in its composite form and is split into its individual components by two 6th -order switched capacitor band-pass filters. Each component is smoothed by an output filter and squared by a hard limiting comparator. The two resulting rectangular waveforms are then applied to a digital circuit, where a counting algorithm measures and averages their periods. An accurate reference clock is derived from an inexpensive external 3.58MHz crystal.

The time required to detect a valid tone pair, tDP, is a function of decode algorithm, tone frequency, and the previous state of the decode logic. ESt (early steering output) indicates that two tones of valid frequency have been detected and initiates an RC timing circuit. If both tones are present for a minimum guard time, determined by an external RC network, the DTMF signal is decoded and the resulting data is latched on the output register. The delayed steering output (StD) is raised to indicate that new data is available. The output corresponding to each key pressed is shown in the truth table.

MT8870 OUTPUT TRUTH TABLE

| FLOW | FHIGH | KEY | TOE | Q4 | Q3 | Q2 | Q1 |
|---|---|---|---|---|---|---|---|
| 697 | 1209 | 1 | 1 | 0 | 0 | 0 | 1 |
| 697 | 1336 | 2 | 1 | 0 | 0 | 1 | 0 |
| 697 | 1477 | 3 | 1 | 0 | 0 | 1 | 1 |
| 770 | 1209 | 4 | 1 | 0 | 1 | 0 | 0 |
| 770 | 1336 | 5 | 1 | 0 | 1 | 0 | 1 |
| 770 | 1477 | 6 | 1 | 0 | 1 | 1 | 0 |
| 852 | 1209 | 7 | 1 | 0 | 1 | 1 | 1 |
| 852 | 1336 | 8 | 1 | 1 | 0 | 0 | 0 |
| 852 | 1477 | 9 | 1 | 1 | 0 | 0 | 1 |
| 941 | 1209 | 0 | 1 | 1 | 0 | 1 | 0 |
| 941 | 1336 | * | 1 | 1 | 0 | 1 | 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 941 | 1477 | # | 1 | 1 | 1 | 0 | 0 |
| 697 | 1633 | A | 1 | 1 | 1 | 0 | 1 |
| 770 | 1633 | B | 1 | 1 | 1 | 1 | 0 |
| 852 | 1633 | C | 1 | 1 | 1 | 1 | 1 |
| 941 | 1633 | D | 1 | 0 | 0 | 0 | 0 |
| ---- | ANY | | 0 | Z | Z | Z | Z |

|---------------------------------------------------------|

|TOE is three-state output-enable input at pin 10 of the IC|

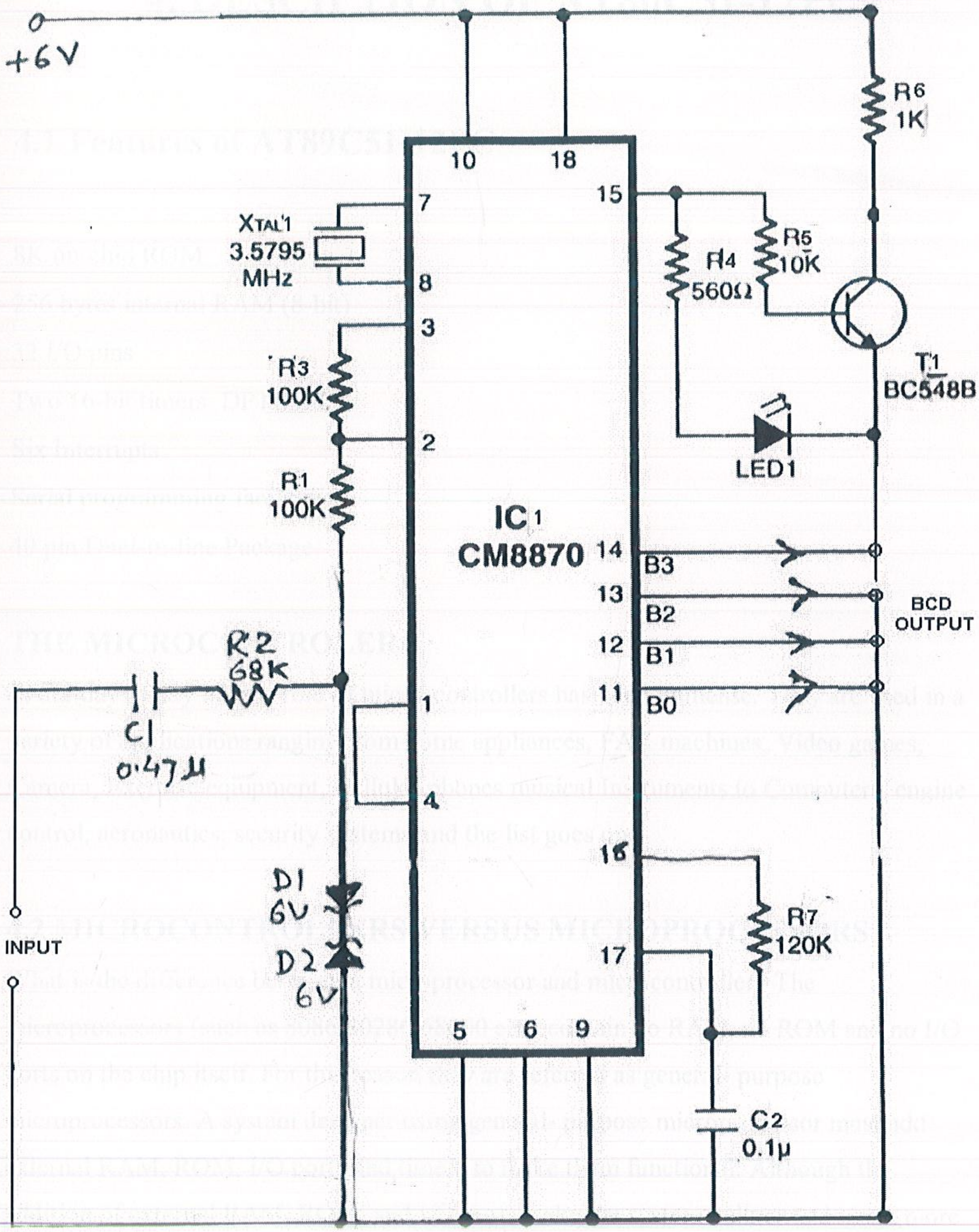



Fig 3.3 Circuit Diagram of DTMF Decoder

# 4. DESCIPTION OF AT80C51-12PC

## 4.1 Features of AT89C51-12PC

- 8K on-chip ROM
- 256 bytes internal RAM (8-bit)
- 32 I/O pins
- Two 16-bit timers: DPTR ,PC
- Six Interrupts
- Serial programming facility
- 40 pin Dual-in-line Package

### THE MICROCONTROLER:

In our day to day life the role of micro-controllers has been immense. They are used in a variety of applications ranging from home appliances, FAX machines, Video games, Camera, Exercise equipment, Cellular phones musical Instruments to Computers, engine control, aeronautics, security systems and the list goes on.

## 4.2 MICROCONTROLLERS VERSUS MICROPROCESSORS

What is the difference between a microprocessor and microcontroller? The microprocessors (such as 8086,80286,68000 etc.) contain no RAM, no ROM and no I/O ports on the chip itself. For this reason they are referred as general- purpose microprocessors. A system designer using general- purpose microprocessor must add external RAM, ROM, I/O ports and timers to make them functional. Although the addition of external RAM, ROM, and I/O ports make the system bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/o ports needed to fit the task at hand. This is the not the case with microcontrollers. A microcontroller has a CPU (a microprocessor) in addition to the fixed amount of RAM, ROM, I/O ports, and timer are all embedded together on the chip: therefore, the designer cannot add any external memory, I/O, or timer to it. The fixed amount of on chip RAM, ROM, and number of I/O ports in microcontrollers make them ideal for many applications in which cost and space are critical. In many applications, for example a TV remote control, there is no need for the computing power of a 486 or even a 8086 microprocessor. In many applications, the space it takes, the power it consumes, and the price per unit are much more critical considerations than the computing power. These applications most often require some I/O operations to read signals and turn on and off certain bits. It is interesting to know that some microcontrollers manufactures have gone as far as integrating an ADC and other peripherals into the microcontrollers.

## 4.3 INTRODUCTION TO 8051

In 1981, Intel Corporation introduced an 8-bit microcontroller called the 8051. This microcontroller had 128 bytes of RAM, 4K bytes of on-chip ROM, two timers, one serial port, and four ports (8-bit) all on a single chip. The 8051 is an 8-bit processor, meaning the CPU can work on only 8- bit pieces to be processed by the CPU. The 8051 has a total of four I/O ports, each 8- bit wide. Although 8051 can have a maximum of 64K bytes of on-chip ROM, many manufacturers put only 4K bytes on the chip.sucessor of 8051 was 8052 with better features like 256 bytes of RAM,8k on chip ROM, 4 timers,2 serial ports The 8051 became widely popular after Intel allowed other manufacturers to make any flavor of the 8051 they please with the condition that they remain code compatible with the 8051. This has led to many versions of the 8051 with different speeds and amount of on-chip ROM marketed by more than half a dozen manufacturers. It is important to know that although there are different flavors of the 8051, they are all compatible with the original 8051 as far as the instructions are concerned. This means that if you write your program for one, it will run on any one of them regardless of the manufacturer. The major 8051 manufacturers are Intel, Atmel, Dallas Semiconductors, Philips Corporation, Infineon.

## 4.4 PIN DESCRIPTION

The 89C51 have a total of 40 pins that are dedicated for various functions such as I/O, RD, WR, address and interrupts. Out of 40 pins, a total of 32 pins are set aside for the four ports P0, P1, P2, and P3, where each port takes 8 pins. The rest of the pins are designated as Vcc, GND, XTAL1, XTAL, RST, EA, and PSEN. All these pins except PSEN and ALE are used by all members of the 8051 and 8031 families. In other words, they must be connected in order for the system to work, regardless of whether the microcontroller is of the 8051 or the 8031 family. The other two pins, PSEN and ALE are used mainly in 8031 based systems.

**Vcc**

Pin 40 provides supply voltage to the chip. The voltage source is +5 V.

**GND**

Pin 20 is the ground.

**XTAL1 and XTAL2**

The 8051 have an on-chip oscillator but requires external clock to run it. Most often a quartz crystal oscillator is connected to input XTAL1 (pin 19) and XTAL2 (pin 18). The quartz crystal oscillator connected to XTAL1 and XTAL2 also needs two capacitors of 30 pF value. One side of each capacitor is connected to the ground.

It must be noted that there are various speeds of the 8051 family. Speed refers to the maximum oscillator frequency connected to the XTAL. For example, a 12 MHz chip must be connected to a crystal with 12 MHz frequency or less. Likewise, a 20 MHz microcontroller requires a crystal frequency of no more than 20 MHz. When the 8051 is connected to a crystal oscillator and is powered up, we can observe the frequency on the XTAL2 pin using oscilloscope.

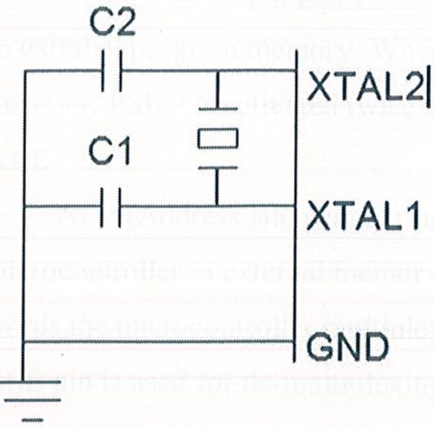



Fig 4.4. XTAL1 and XTAL2

**RST**

Pin 9 is the reset pin. It is an input and is active high (normally low). Upon applying a high pulse to this pin, the microcontroller will reset and terminate all activities. This is often referred to as a power –on reset. Activating a power-on reset will cause all values in the registers to be lost. Notice that the value of Program Counter is 0000 upon reset, forcing the CPU to fetch the first code from ROM memory location 0000. This means that we must place the first line of source code in ROM location 0000 that is where the CPU wakes up and expects to find the first instruction. In order to RESET input to be effective, it must have a minimum duration of 2 machine cycles. In other words, the high pulse must be high for a minimum of 2 machine cycles before it is allowed to go low.

**EA**

All the 8051 family members come with on-chip ROM to store programs. In such cases, the EA pin is connected to the Vcc. For family members such as 8031 and 8032 in which there is no on-chip ROM, code is stored on an external ROM and is fetched by the 8031/32. Therefore for the 8031 the EA pin must be connected to ground to indicate that the code is stored externally. EA, which stands for "external access," is pin number 31 in the DIP packages. It is input pin and must be connected to either Vcc or GND. In other words, it cannot be left unconnected.

**PSEN**

This is an output pin. PSEN stands for "program store enable." It is the read strobe to external program memory. When the microcontroller is executing from external memory, PSEN is activated twice each machine cycle.

**ALE**

ALE (Address latch enable) is an output pin and is active high. When connecting a microcontroller to external memory, potr 0 provides both address and data. In other words the microcontroller multiplexes address and data through port 0 to save pins. The ALE pin is used for de-multiplexing the address and data by connecting to the G pin of the 74LS373 chip.

## 4.5 I/O port pins and their functions

The four ports P0, P1, P2, and P3 each use 8 pins, making them 8-bit ports. All the ports upon RESET are configured as output, ready to be used as output ports. To use any of these as input port, it must be programmed.

**Port 0**

Port 0 occupies a total of 8 pins (pins 32 to 39). It can be used for input or output. To use the pins of port 0 as both input and output ports, each pin must be connected externally to a 10K-ohm pull-up resistor. This is due to fact that port 0 is an open drain, unlike P1, P2 and P3. With external pull-up resistors connected upon reset, port 0 is configured as output port. In order to make port 0 an input, the port must be programmed by writing 1 to all the bits of it. Port 0 is also designated as AD0-AD7, allowing it to be used for both data and address. When connecting a microcontroller to an external memory, port 0 provides both address and data. The microcontroller multiplexes address and data through port 0 to save pins. ALE indicates if P0 has address or data. When ALE=0, it provides data D0-D7, but when ALE=1 it has address A0-A7. Therefore, ALE is used for de-multiplexing address and data with the help of latch 74LS373.

**Port 1**

Port 1 occupies a total of 8 pins (pins 1 to 8). It can be used as input or output. In contrast to port 0, this port does not require pull-up resistors since it has already pull-up resistors internally. Upon reset, port 1 is configures as an output port. Similar to port 0, port 1 can be used as an input port by writing 1 to all its bits.

**Port 2**

Port 2 occupies a total of 8 pins (pins 21 to 28). It can be used as input or output. Just like P1, port 2 does not need any pull-up resistors since it has pull-up resistors internally. Upon reset port 2 is configured as output port. To make port 2 input, it must be programmed as such by writing 1s to it.

**Port 3**

Port 3 occupies a total of 8 pins (pins 10 to 17). It can be used as input or output. P3 does not need any pull-up resistors, the same as P1 and P2 did not. Although port 3 is configured as output port upon reset, this is not the way it is most commonly used. Port 3 has an additional function of providing some extremely important signals such as interrupts. Some of the alternate functions of P3 are listed below:

P3.0 RXD (Serial input)

P3.1 TXD (Serial output)

P3.2 INT0 (External interrupt 0)

P3.3 INT1 (External interrupt 1)

P3.4 T0 (Timer 0 external input)

P3.5 T1 (Timer 1 external input)

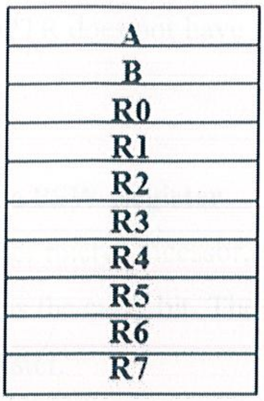
P3.6 WR (External memory write strobe)

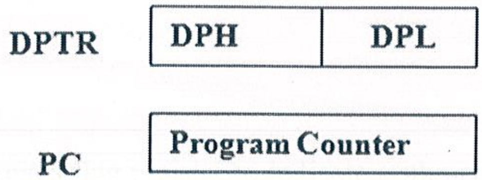
P3.7 RD (External memory read strobe)

## 4.6 INSIDE THE 89C51

**Registers**

In the CPU, registers are used to store information temporarily. That information could be a byte of data to be processed, or an address pointing to the data to be fetched. In the 8051 there us only one data type: 8 bits. With an 8- bit data type, any data larger than 8 bits has to be broken into 8-bit chunks before it is processed.




(b) Some 8051 16-bit registers

(a) Some 8051 8-bit registers

Fig 4.6.1-Register Banks

The most commonly used registers of the 8051 are A(accumulator), B, R0, R1, R2, R3, R4, R5, R6, R7, DPTR (data pointer) and PC (program counter). All the above registers are 8-bit registers except DPTR and the program counter. The accumulator A is used for all arithmetic and logic instructions.

Memory Allocation

RAM(00 to 7FH)

00 to 1FH-register banks

20-2F- Bit addresable read\write memory

30-7F- Scratch Pad

**Program Counter and Data Pointer**

The program counter is a 16- bit register and it points to the address of the next instruction to be executed. As the CPU fetches op-code from the program ROM, the program counter is incremented to point to the next instruction. Since the PC is 16 bit wide, it can access program addresses 0000 to FFFFH, a total of 64K bytes of code. However, not all the members of the 8051 have the entire 64K bytes of on-chip ROM installed.

The DPTR register is made up of two 8-bit registers, DPH and DPL, which are used to furnish memory addresses for internal and external data access. The DPTR is under the control of program instructions and can be specified by its name, DPTR. DPTR does not have a single internal address, DPH and DPL are assigned an address each.

**Flag bits and the PSW Register**

Like any other microprocessor, the 8051 have a flag register to indicate arithmetic conditions such as the carry bit. The flag register in the 8051 is called the program status word (PSW) register.

The program status word (PSW) register is an 8-bit register. It is also referred as the flag register. Although the PSW register is 8-bit wide, only 6 bits of it are used by the microcontroller. The two unused bits are user definable flags. Four of the flags are conditional flags, meaning they indicate some conditions that resulted after an instruction was executed. These four are CY (carry), AC (auxiliary carry), P (parity), and OV (overflow). The bits of the PSW register are shown below:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **CY** | **AC** | **F0** | **RS1** | **RS0** | **OV** | -- | **P** | |

| | | |
|---|---|---|
| CY | PSW.7 | Carry flag |
| AC | PSW.6 | Auxiliary carry flag |
| -- | PSW.5 | Available to the user for general purpose |
| RS1 | PSW.6 | Register bank selector bit 1 |
| RS0 | PSW.3 | Register bank selector bit 0 |
| OV | PSW.2 | Overflow flag |
| F0 | PSW.1 | User definable bit |
| P | PSW.0 | Parity flag |

Fig 4.6.2-CY, the carry flag

This flag is set whenever there is a carry out from the d7 bit. This flag bit is affected after an 8-bit addition or subtraction. It can also be set to 1 or 0 directly by an instruction such as "SETB C" and "CLR C" where "SETB C" stands for set bit carry and "CLR C" for clear carry.

**AC, the auxiliary carry flag**

If there is carry from D3 to D4 during an ADD or SUB operation, this bit is set: otherwise cleared. This flag is used by instructions that perform BCD arithmetic.

**P, the parity flag**

The parity flag reflects the number of 1s in the accumulator register only. If the register A contains an odd number of 1s, then P=1. Therefore, P=0 if Ahas an even number of 1s.

**OV, the overflow flag**

This flag is set whenever the result of a signed number operation is too large, causing the high order bit to overflow into the sign bit. In general the carry flags is used to detect errors in unsigned arithmetic operations.

## 4.7 MEMORY SPACE ALLOCATION

1. **Internal ROM**

The 89C51 has a 4K bytes of on-chip ROM. This 4K bytes ROM memory has memory addresses of 0000 to 0FFFh. Program addresses higher than 0FFFh, which exceed the internal ROM capacity will cause the microcontroller to automatically fetch code bytes from external memory. Code bytes can also be fetched exclusively from an external memory, addresses 0000h to FFFFh, by connecting the external access pin to ground. The program counter doesn't care where the code is: the circuit designer decides whether the code is found totally in internal ROM, totally in external ROM or in a combination of internal and external ROM.

2. **Internal RAM**

The 1289 bytes of RAM inside the 8051 are assigned addresses 00 to 7Fh. These 128 bytes can be divided into three different groups as follows:

1. A total of 32 bytes from locations 00 to 1Fh are set aside for register banks and the stack.
2. A total of 16 bytes from locations 20h to 2Fh are set aside for bit addressable read/write memory and instructions.
3. A total of 80 bytes from locations 30h to 7Fh are used for read and write storage, or what is normally called a scratch pad. These 80 locations of RAM are widely used for the purpose of storing data and parameters by 8051 programmers.

## 4.8 Assembling And Running in 8051

Editor Program

Produces a file with extension ".src" or "asm".

Assembler Program

Produces a file with extension ".lst" and the other file with extension ".obj" is fed up to the next stage

Linker Program

Produces an object file with extension ".abs"

OH Program

Next this ".abs" file is fed into a programer called "OH" which creates a file with extension hex which is machine language to be burned into the ROM.

# 5. Motion Control And Power Supply

## 5.1 Block diagram of power supply

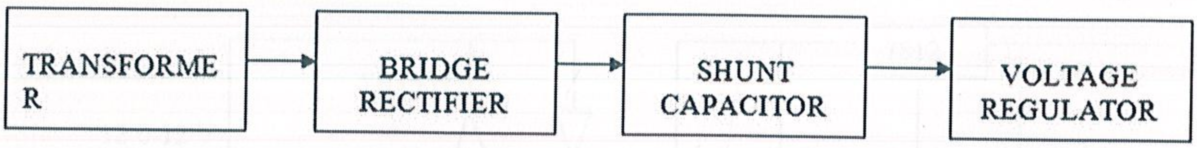



Fig 5.1.1 –Block Diagram of Power supply

These 4 main components of the power supply regulate the amount of voltage in the circuit.

Transformer is used to step down the voltage supply. Rectifier converts the ac to dc. Capacitors are used to nullify the ripples from the pulsating dc voltagesand regulator provides regulated power supply to the components. Their detailed working is explained in the next section.

## 5.2 Working of power supply

### The power supply circuit comprises of four basic parts:

The transformer steps down the 220 V a/c. into 12 V a/c. The transformer work on the principle of magnetic induction, where two coils: primary and secondary are wound around an iron core. The two coils are physically insulated from each other in such a way that passing an a/c. current through the primary coil creates a changing voltage in the primary coil and a changing magnetic field in the core. This in turn induces a varying a/c. voltage in the secondary coil.

The a/c. voltage is then fed to the bridge rectifier. The rectifier circuit is used in most electronic power supplies is the single-phase bridge rectifier with capacitor filtering, usually followed by a linear voltage regulator. A rectifier circuit is necessary to convert a signal having zero average value into a non-zero average value. A rectifier transforms alternating current into direct current by limiting or regulating the direction of flow of current. The output resulting from a rectifier is a pulsating D.C. voltage. This voltage is not appropriate for the components that are going to work through it.

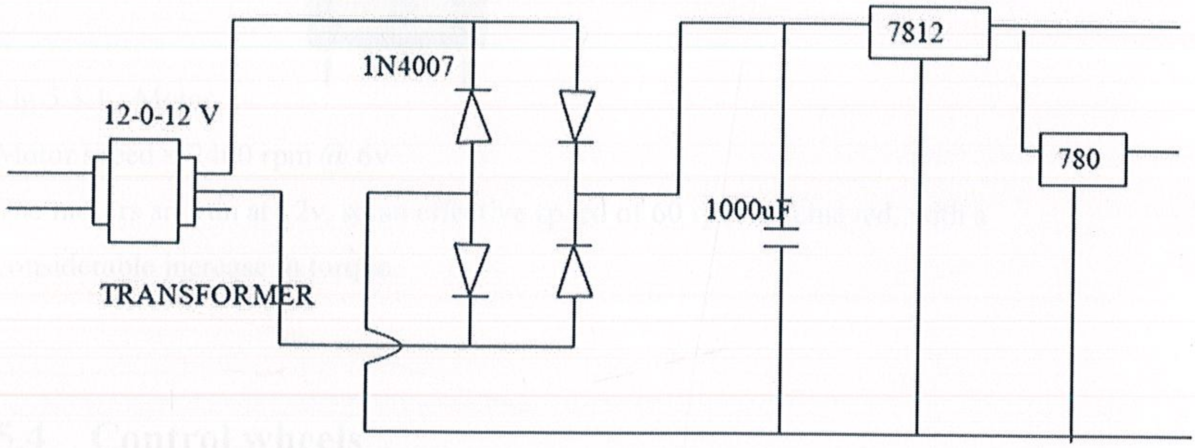



Fig 5.2.1-Power supply circuit

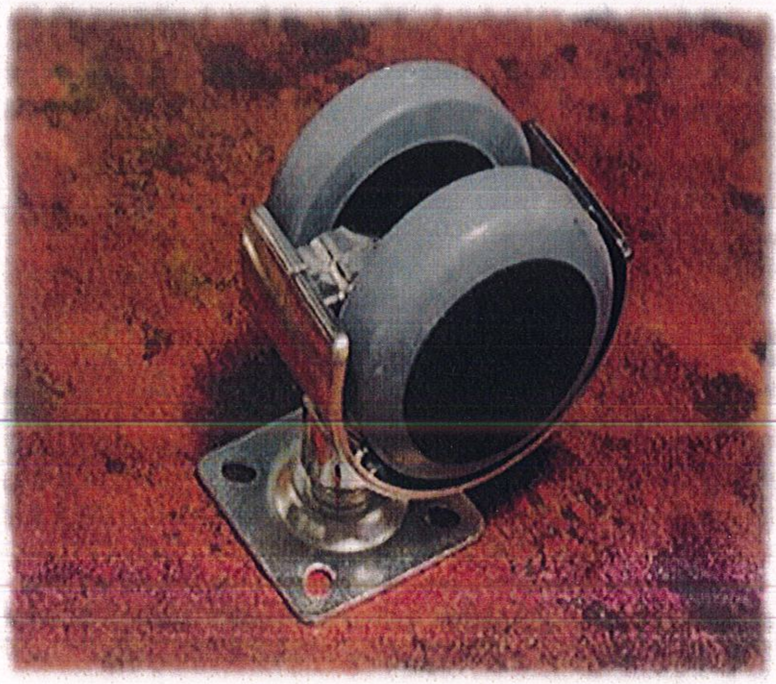
The ripple of the D.C. voltage is smoothened using a filter capacitor of 1000 microF 25V. The filter capacitor stores electrical charge. If it is large enough the capacitor will store charge as the voltage rises and give up the charge as the voltage falls. This has the effect of smoothing out the waveform and provides steadier voltage output. A filter capacitor is connected at the rectifier output and the d.c voltage is obtained across the capacitor. When this capacitor is used in this project, it should be twice the supply voltage. When the filter is used, the RC charge time of the filter capacitor must be short and the RC discharge time must be long to eliminate ripple action. In other words the capacitor must charge up fast, preferably with no discharge.

## 5.3 Motors


Fig 5.3.1 : Motor

- Motor speed = 2400 rpm @ 6v
- The motors are run at 12v, so an effective speed of 60 rpm is achieved, with a considerable increase in torque.

## 5.4 Control wheels



5.3.2 -type of control wheel

# 6. REFLECTOR CIRCUIT

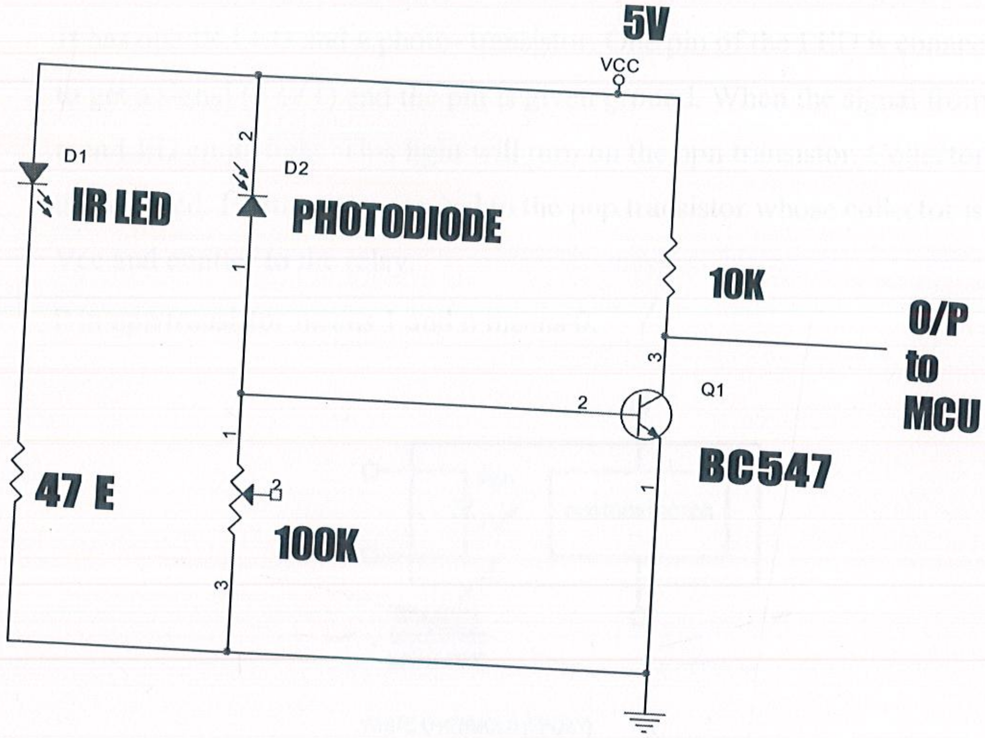



Fig 6.1 Circuit of Reflector

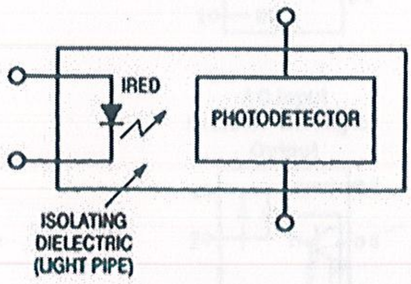
## 6.2 Description of components in reflector circuit

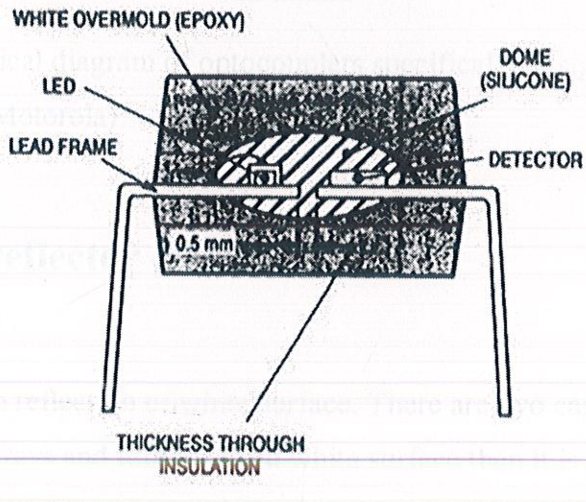
### 6.2.1 Optocoupler

The simple solution to **THE REFLECTOR CIRCUIT** is to combine an LED with a phototransistor. The new device is totally encapsulated so that the light from the LED is focused directly on the opening in the phototransistor, and no other forms of light could be detected. The input signal is connected to the LED and the output signal is connected to the transistor. The device is called an optocoupler or optoisolator. Fig. shows a block diagram of an optocoupler that shows an LED shining light directly on a photodector, which is usually a phototransistor. The second diagram in the figure shows how the LED is located so its light is focused directly on the phototransistor.

**It** has one IR LED and a photo- transistor. One pin of the LED is connected to the MCU to get a signal (0 or 1) and the pin is given ground. When the signal from the MCU is 0, then LED emits light. This light will turn on the npn transistor. Collector of the transistor is grounded. Emitter is connected to the pnp transistor whose collector is connected to Vcc and emitter to the relay.

P in npn transistor means 1 and n means 0.




### 6.2.2 Adding Bias to the Phototransistor of the Optocoupler

The optocoupler is also useful in circuits where the bias of the phototransislor is changed. When bias voltage is added to the base of the phototransistor in an optocoupler, it can make the optocoupler more or less sensitive. If the bias voltage to the base of an NPN transistor is slightly positive, the optocoupler will become more sensitive because the LED does not need to produce as much light to make the phototransistor begin to conduct. If the bias voltage is slightly negative, the optocoupler will become less sensitive and the LED must have more current applied before it can produce enough light to overcome the bias on the phototransistor. The optocoupler must have a base terminal for the transistor brought out to a pin so that it is usable to add bias to it. The optocouplers in Fig. 3-36 show transistors with their base brought out to pin 6.

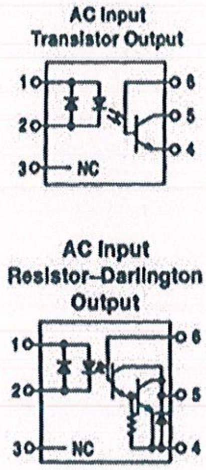



FIGURE 6.2.3 _ Electrical diagram of optocouplers specifically designed for AC input signals. (Copyright of Motorola)

## 6.3 Working of reflector circuit

This Circuit is works on reflection of white surface. There are two cases, In First case when IR LED emits IR rays and reflects from white surface then it is received by photodiode. When IR rays fall on photodiode then it passes 5V to base of transistor (BC 547).Transistor gets turn on and passes 5V from collector to emitter. The output of reflector circuit is 0.

This zero send to MCU then MCU take action according to condition which is written in program.

In second case when IR LED emits IR rays and absorb by black surface then it is not received by photodiode. Photodiode remains OFF and it is not pass 5V to base of transistor (BC 547).Transistor remains turn OFF . The output of reflector circuit is 1.This one send to MCU then MCU take action according to condition which is written in program

# 7. SOFTWARE REQUIREMENT SPECIFICATION

**7.1 FOR SIMULATION OF THIS PROJECT WE HAVE USED TWO SOFTWARES**

**KEIL-µVISION3: **

The µVision3 IDE is a Windows-based software development platform that combines a robust editor, project manager, and makes facility. µVision3 integrates all tools including the C compiler, macro assembler, linker/locator, and HEX file generator.

**Proteus 7 Professional:**

All uses can run a pre-configured routing schedule automatically. Per net 'strategies' are replaced by 'net 'classes' and the management of these is incorporated into the Design Rule Manager.

## 7.2 Code which supports our device

```
#include<lcdrout.h>
#define lb P20
#define lf P21
#define rb P22
```

```
#define rf P23
#define ref P24          //reflector
bit kk=1;
#define std P27
#define dport P0

void left(void)
{
lb=lf=rf=rb=1;
lf=0;
ms_delay(200);
ms_delay(200);
lb=lf=rf=rb=1;
}

void right(void)
{
lb=lf=rf=rb=1;
rf=0;
ms_delay(200);
ms_delay(200);
lb=lf=rf=rb=1;

}

void fwd(void)
{
lb=lf=rf=rb=1;
rf=0;lf=0;

}
```

```
void stop(void)
{
lb=lf=rf=rb=1;

}
void back(void)
{
lb=lf=rf=rb=1;
lb=rb=0;
secdelay(1);
lb=lf=rf=rb=1;
}
 void external (void)interrupt 0
{
                stop();
                lcd_cmd1(0xc0);
                lcd_puts("obstacle ") ;
                back();
                secdelay(3);
                right();
                fwd();

}
void main()
{ unsigned char k=0;
        lcd_init();
        lcd_puts("Robot Control");
        lcd_cmd1(0xc0);
        lcd_puts("DTMF");
        secdelay(2);
```

```
clrscr();
std=1;              IE=0x81;
lcd_cmd1(0xc0);
lcd_puts("Waiting...      ");
while(std==1);
std=1;
secdelay(2);
while(1)
{       lcd_cmd1(0x80);
        lcd_puts("Connected          ");
  while(std==1);
        ms_delay(15);
        k=dport & 0x0f;
        lcd_cmd1(0x80);
        lcd_puts("Recieved: ");
        displaypval(k);
        if(k==2)
        {
        lcd_cmd1(0xc0);
        lcd_puts("Moving FWD.           ");
        fwd();

        }
        if(k==1)
        {
        lcd_cmd1(0xc0);
        lcd_puts("Moving Left.           ");
        left();
        }
        if(k==3)
        {
```

```
lcd_cmd1(0xc0);
lcd_puts("Moving Right          ");
right();
}
if(k==8)
{
lcd_cmd1(0xc0);
lcd_puts("Moving Back.");
back();
}
if(k==5)
{
lcd_cmd1(0xc0);
lcd_puts("     stop        ");
stop();
}
P0=255;
std=1;


}

}
```

- ## INTEGRATION AND SYSTEM TESTING PHASE

In the integration and system testing phase all the program units are integrated and tested to ensure that the complete system meets the device requirements. After this stage the device is assembled and ready to use

- ## MAINTENANCE PHASE

The maintenance phase the usually the longest stage of the device to :

- Meet the changing needs of the enviroment
- Adapted to accommodate changes in the external environment
- Enhancing the efficiency of the device

# CONCLUSION

In this project ,we have used 8051 microcontroller and interfaced it with DTMF decoder which in turn I connected to a Bluetooth dongle. This has been done in order to establish connection between the robot and a Bluetooth enabled device.

In order to provide our robot with a regulated power supply ,we have used a power supply block consisting of a step down transformer which steps down 220v ac into 12 v ac. Then we have a bridge rectifier which converts 12v ac into a pulsating 12v dc.This is followed by a 100pf capacitor which removes the ripples from the pulsating 12v dc.This in turn is followed by a regulator ,which provides a regulated power supply to the robot and the microcontroller.

For the robot to detect and avoid collision ,we have used a reflector circuit.A reflector circuit consist of a photo diode and a LED.The LED emits infra red rays which gets reflected from the obstacle and is received by the photodiode.On receiving the IR ray is received by the diode it sends an interrupt to the microcontroller warning it of the obstacle in the path of the robot. Then , the microcontroller takes decision based on the coding that we have done.

Thus, we have made our project successfully.To operate it ,we first connect the Bluetooth dinggle to a Bluetooth enabled device.Then we press the keys (1,2,3,5,8) which is decoded by the DTMF decoder and thus microcontroller takes appropriate action based on our coding i.e. if we press 1 it turns left, on pressing 2 it moves forward,on pressing 3 it moves towards right and so on.

Hence, we have successfully completed our project.

# FUTURE SCOPE OF WORK

Future enhancements recommended are :

1. This project can be enhanced using a mechanical arm for bringing back samples .
2. It can be equipped with a camera for surveillance purposes.
3. A battery can be fixed on the device to make it totally mobile.
4. User Interface can still be made better.
5. A bigger version can be made for real situations.

# 7. REFERENCES


1. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. Mc Kinlay , *The 8051*
2. *Microcontroller & Embedded Systems*, Pearson Education Inc. 2nd Edition, 2008.
3. Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming &*
4. *Applications,*Penram International, 2nd Edition, 1996.
5. www.8051projects.net/microcontroller_tutorials/
6. www.8051projects.net/lcd-interfacing