MOBILE CONTROLLED ROBOT

Submitted in partial fulfillment of the Degree
of
Bachelor of Technology

May – 2014

Enrollment. No. - 101042 , 101061
Name of Students - Parth Khurana,
Abhinav Sharma
Name of supervisor - Dr. Rajiv Kumar

DEPARTMENT OF ELECTRONICS AND
COMMUNICATION ENGINEERING
JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
,WAKNAGHAT

CERTIFICATE

This is to certify that project report entitled Mobile Controlled Robot submitted by Parth Khurana (101042) & Abhinav Sharma (101061) in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to JayPee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: **Supervisor's Name**

Designation

ACKNOWLEDGEMENT

I would like to thank many people who have helped us directly or indirectly in the successful completion of the project. I would like to thank one and all from the core of my heart.

First of all I would like to express my deep sense of gratitude towards my project Guide Dr.Rajiv Kumar, (ECE Deptt.) for always being available whenever I require his guidance as well as for motivating me through out the project work.

I would like to express my deep gratitude towards our other teachers and other staff for giving their valuable suggestions and co operation for doing my project.

I am deeply thankful to Brig. Balbir singh , Director of Jaypee university of Information Technology Waknaghat for providing necessary facilities during the execution of this project.

I would like to thank all my friends for their help and constructive criticism during this project period. Finally, I am very much indebted to my parents for their moral support and encouragement to achieve higher goals. I have no words to express my gratitude and still I am very thankful to my parents who have given me their every support.

Signature

Parth Khurana {101042}

Abhinav Sharma {101061}

TABLE OF CONTENT

ABSTRACT

In this project we show that how we control the movement of the robot with the help of any mobile phone using DTMF tones.

Conventionally, wireless-controlled robots use RF circuits, which have the drawbacks of limited working range, limited frequency range and limited control. Use of a mobile phone for robotic control can overcome these limitations. It provides the advantages of robust control, working range as large as the coverage area of the service provider, no interference with other controllers and up to twelve controls.

In this project, the robot is so controlled by a mobile phone that makes a call to the mobile phone attached to the robot. In the course of a call, if any button is pressed, a tone corresponding to the button pressed is heard at the other end of the call. This tone is called 'dual tone multiple frequency' (DTMF) tone. The robot perceives this DTMF tone with the help of the phone stacked in the robot.

The receiver tone is processed by ATMEL 89S52 microcontroller with the help of a DTMF decoder MT8870. The decoder decodes the DTMF tone into its equivalent binary digit and this binary number is sent to the microcontroller. The microcontroller is preprogrammed to take a decision for any given input and outputs its decision to motor drivers in order to drive the motors for forward or backward motion or a turn.

The mobile that makes a call to the mobile phone stacked in the robot acts as a remote. So this simple robotic project does not require the construction of receiver and transmitter units.

CHAPTER-1

1.1 BASIC OPERATION

In order to control the robot, you need to make a call to the cell phone attached to the robot (through headphone) from any phone, which sends DTMF tunes on pressing the numeric buttons. The cell phone in the robot is kept in 'auto answer' mode.(if the mobile does not have the auto answering facility ,receive the call by 'OK' key on the rover connected mobile and then made it in hands-free mode.) so after a ring, the cell phone accepts the call. Now you may press any button on your mobile to perform actions as listed in the table. The DTMF tones thus produced are received by the cell phone in the robot. These tones are fed to the circuit by headset of the cell phone. The MT8870 decodes the received tone and sends the equivalent binary number to the microcontroller. According to the program in the microcontroller, the robot starts moving. When you press key '2' (binary equivalent 00000010) on your mobile phone, Then robo move to forward and move forward until you press the 5 button which is assigned for stop code. As we press 5 code robot is be stop automatically. When we press 8 code from mobile then Robo moves to backward position. As we press 6 for right and press 4 for left direction.

Button 1 and 3 for the camera and search light option. When we press 1 then camera or light is on/off and is same for the 3. Basically 1 and 3 is for on/off action. We use it for any type of on/off operation

Table 1

Actions performed corresponding to the keys pressed

Number pressed by user	output of DTMF decoder	input to microcontroler	output from microcontroler	action performed
2	0x02 00000010	0xFD 11111101	0x89 10001001	Forward motion
4	0x04 00000100	0xFB 11111011	0x85 10000101	Left turn
6	0x06 00000110	0xF9 11111001	0x8A 10001010	Right turn
8	0x08 00001000	0xF7 11110111	0x86 10000110	Backward motion
5	0x05 00000101	0xFA 11111010	0x00 00000000	Stop

Figure 1 - ACTIONS PERFORMED

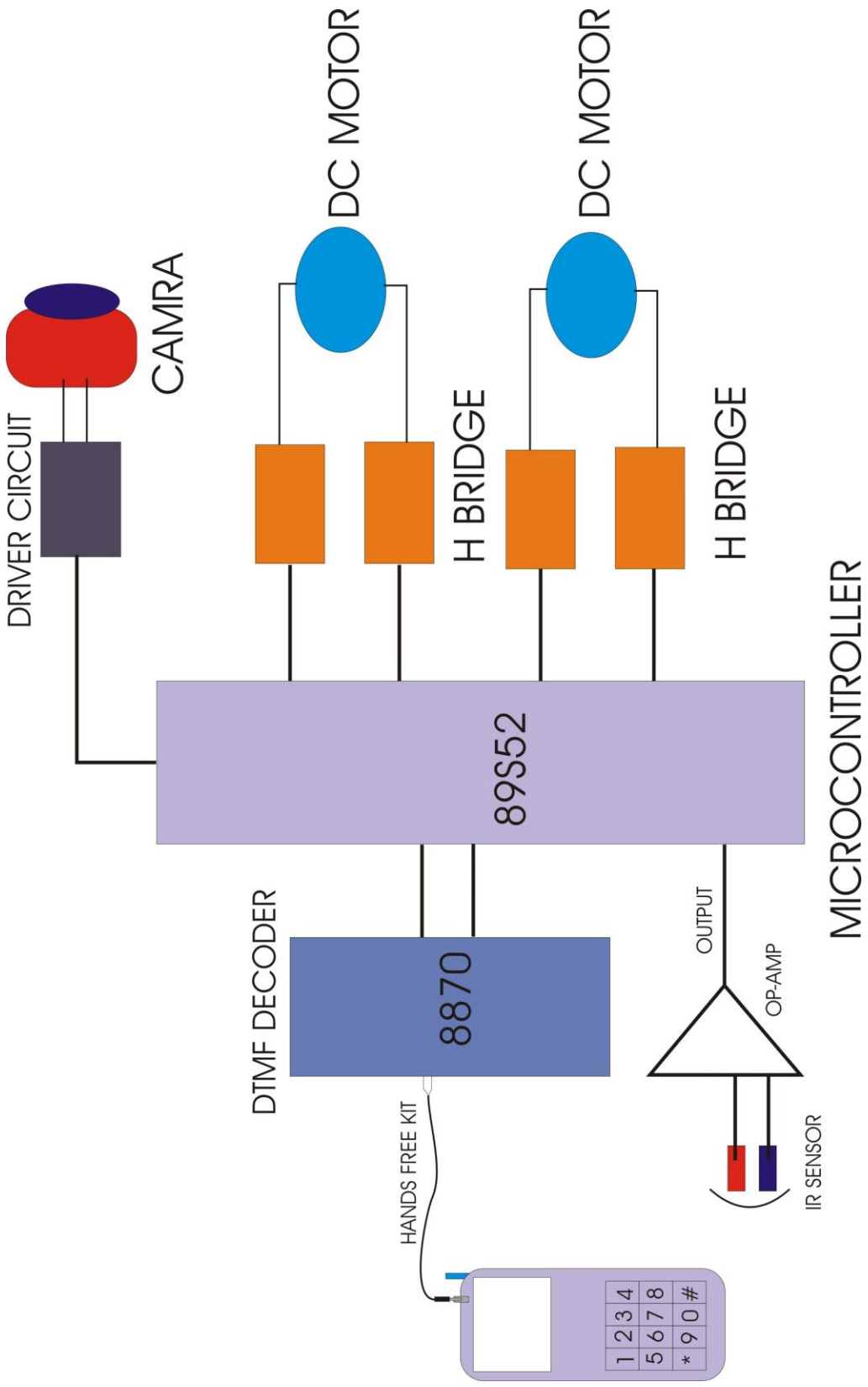


Figure 2 - BASIC BLOCK DIAGRAM

1.3 Circuit Diagram

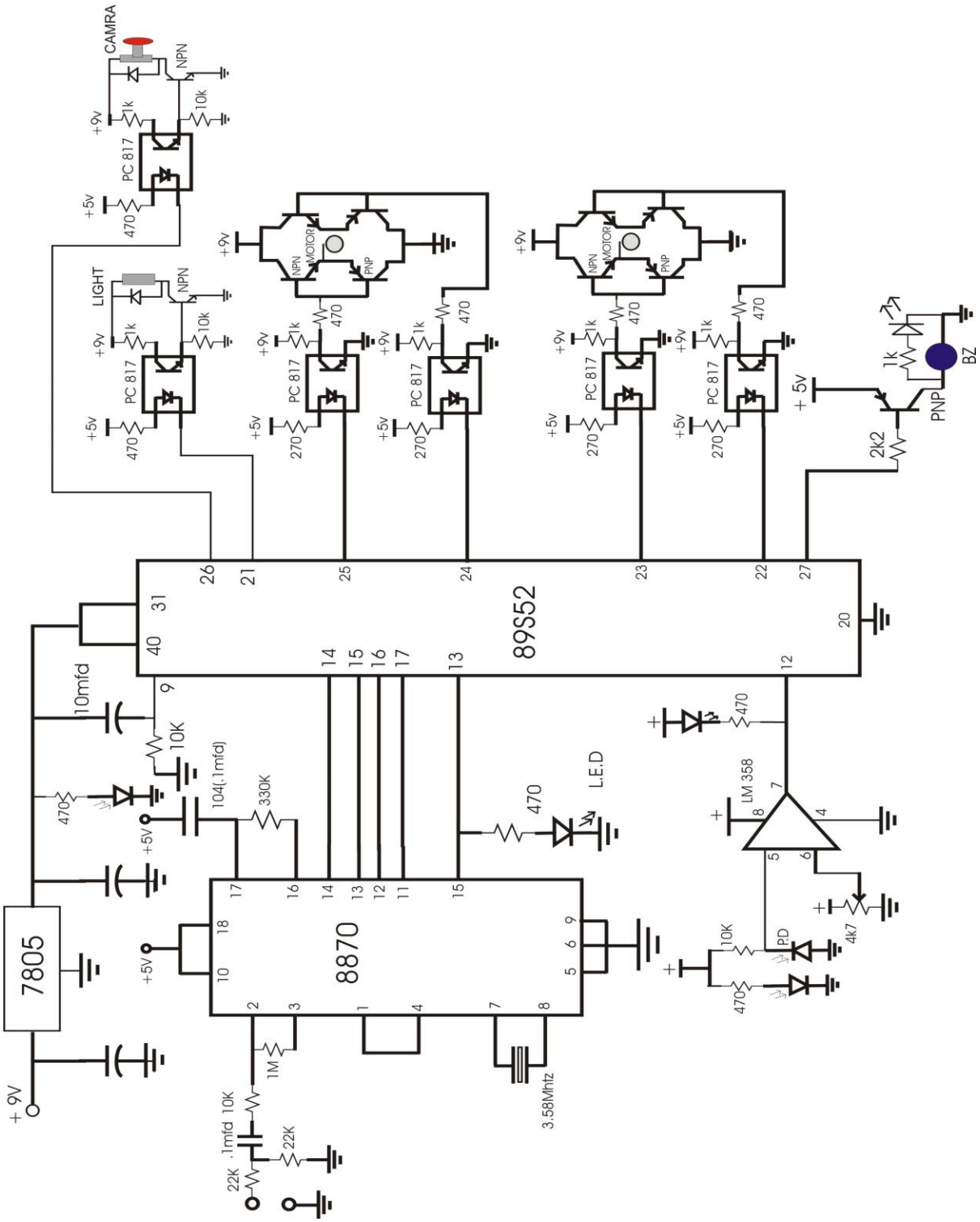


Figure 3 - CIRCUIT DIAGRAM

1.4 COMPONENTS USED

A. **Semiconductors**

MT8870 DTMF decoder
ATMEL89S52
LM358 OPAMP AS A COMPARATOR FOR INFRA RED INTERRUPTION
7805 REGULATOR IC
PC 817 OPTOCOUPLER (6) OPTO ISOLATOR TO PROVIDE ISOLATION
NPN BEL 100N(6)
PNP BEL 100P (5)

B. Resistors

270 OHM (10)
1K(4)
2K2(5)
1M OHM
330K
10K
22K(2)

C. Capacitors:

1000MFD (3)
27PF (2)
.01 MFD (1)
10 MFD (1)

4. Miscellaneous:

M1, M2-6V, 50-rpm geared DC motor

Batt-6V, 4.5Ah battery

2 pin male and female connector

L,E,D (6)

BI-COLOUR L.E.D (2)

HANDS FREE KIT (3.5 MM)

TYRES (2)

CHAPTER - 2

2.1 H BRIDGE

In this project we use ic 89s52 as a main processor to control all the sensor's and output motor's. In this project we use two slow speed dc gear motor for forward, reverse, left and right movement. When both the motor's on clock wise then robot move forward. When both the motor's on anticlock wise then robot moves backward. If one motor is forward and second motor is reverse then robot from left to right position. If we change the sequence of motor then robot turns right to left

We use slow speed gear motor in this project. Working voltage of these motor's are 9 volt to 12 volt dc. We use two power source in this project. One for the motor's and second for the controller circuit.

For controlling a dc motor we use H bridge circuit. IN this project we use four transistor circuit to control the movement of dc motor for forward and reverse movement.

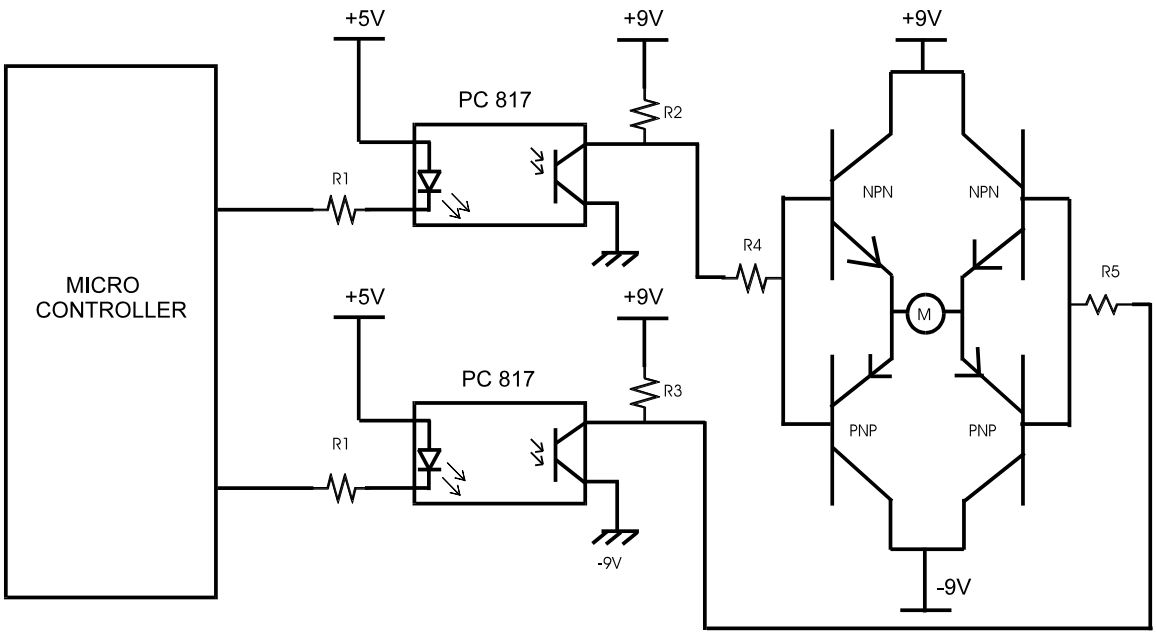


Figure 4- H BRIDGE

Collector of both the transistor is connected to the positive supply 9 volt. This 9 volt supply is for the DC motor. If we use 12 volt motor then we use 12 volt dc supply here. Emitter of both the transistor is connected to the DC motor. Emitter of the PNP transistor is connected to the emitter of NPN transistor. Collector of both the PNP transistor is connected to the ground potential. Base point of both transistor is join together. On this point we give a voltage.

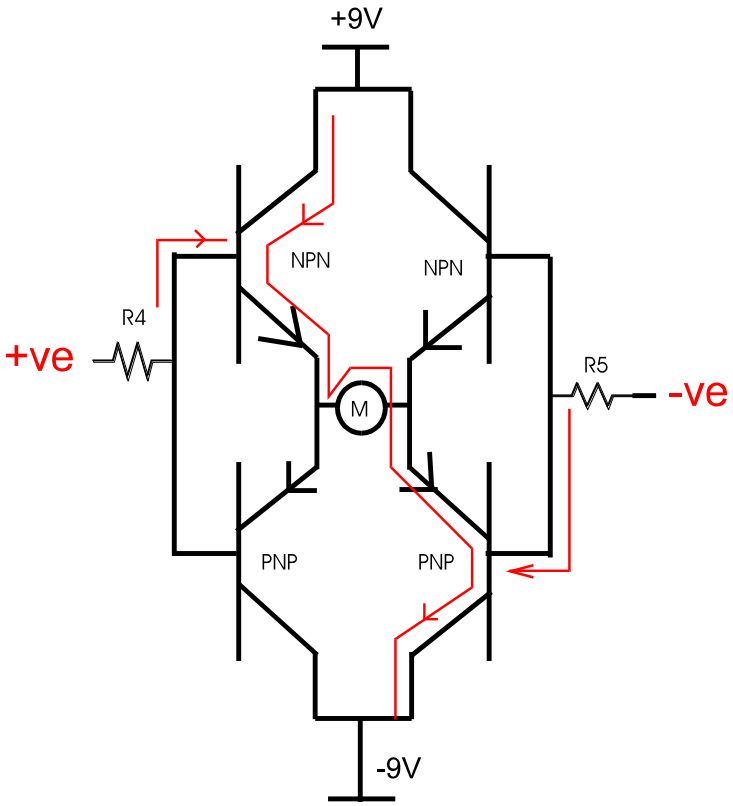


Figure 5- WORKING OF H BRIDGE

if we give a positive voltage to the base of left junction and negative voltage to the right junction then motor moves to one direction. Because due to positive on base NPN is on and due to negative on base PNP is on. If left side NPN is on and right side PNP is on then motor moves to the one direction. If the voltage is reverse on the base point then motor's moves to the reverse direction.

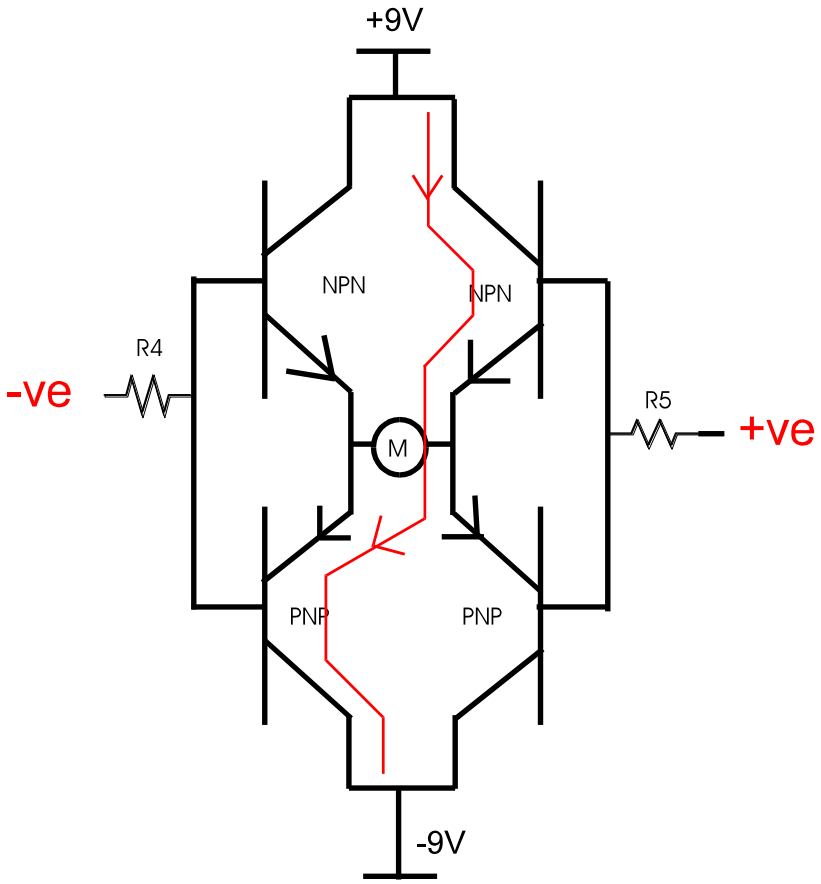


Figure 6- WORKING OF H BRIDGE

Motor move to the reverse direction because base voltage is change . Now left NPN and right PNP is on and motor moves to the reverse direction. Now when we attach the H bridge to the logical output of the micro-controller. So to interface the micro-controller with this H bridge we must connect a OPTO-COUPLER with the controller.

2.2 OPTO-COUPLER

Opto-Coupler is a special optically isolated device to interface the input with output using light. Opto-Coupler provide a electrical isolation between the input and output circuit .Opto-coupler provide a isolation between the two power supply. Microcontroller power supply is 5 volt dc and motor supply is vary from 9volt to 12 volt dc. With the help of the microcontroller we provide a optical isolation between two power supply

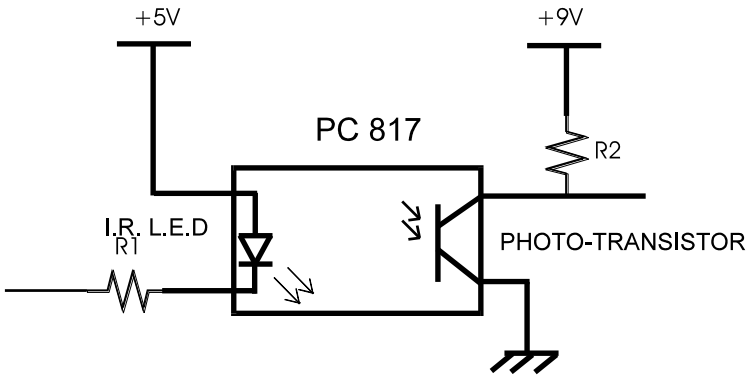


Figure 7-OPTOCOUPLER CIRCUIT

In opto-coupler there is one input and one output and there is no connection between input and output. On input point there is one infra red l.e.d. cathode point of the l.e.d is connected to the resistor R1 and further connected to the microcontroller ports. In this project we use two dc motor, so we use two H bridge circuit with the four opto-coupler.

Pin no 40 of the controller is connected to the positive supply and pin no 20 is connected to the ground pin. Pin no 9 is for the reset pin, on this pin we connect a one resistor and capacitor to provide a auto reset circuit. With the help of the auto reset circuit micro-controller reset automatically and start from the zero location every time when power is on.

When robo moves to forward then infra red sensor is activate. For infra red sensors we use one LM 358 op-amp as a comparator to sense the interruption from any object. When robo moves towards forward then infra red beam is reflected from the surface and reverse back to the photodiode . Photodiode is connected through the resistor in series. When photodiode sense a reflected light from the infra red then photodiode resistance become change and current flowing through the photodiode is also change. This change of voltage is compare in the comparator and connected to the

microcontroller. We use LM 358 as a comparator. Lm 358 is a dual op-amp inside but we use one side of op-amp

CHAPTER 3

3.1 HOW TO PROGRAM A BLANK CHIP.

The 8051

The 8051 developed and launched in the early 80`s, is one of the most popular micro controller in use today. It has a reasonably large amount of built in ROM and RAM. In addition it has the ability to access external memory.

The generic term `8x51` is used to define the device. The value of x defining the kind of ROM, i.e. x=0, indicates none, x=3, indicates mask ROM, x=7, indicates EPROM and x=9 indicates EEPROM or Flash.

Different micro controllers in market.

- PIC One of the famous microcontrollers used in the industries. It is based on RISC Architecture which makes the microcontroller process faster than other microcontroller.

- INTEL These are the first to manufacture microcontrollers. These are not as sophisticated other microcontrollers but still the easiest one to learn.

- ATMEL Atmel's AVR microcontrollers are one of the most powerful in the embedded industry. This is the only microcontroller having 1kb of ram even the entry stage. But it is unfortunate that in India we are unable to find this kind of microcontroller.

3.2 MICROCONTROLLERS

Look around. Notice the smart "intelligent" systems? Be it the T.V, washing machines, video games, telephones, automobiles, aero planes, power systems, or any application having a LED or a LCD as a user interface, the control is likely to be in the hands of a micro controller!

Measure and control, that's where the micro controller is at its best.

Micro controllers are here to stay. Going by the current trend, it is obvious that micro controllers will be playing bigger and bigger roles in the different activities of our lives. These embedded chips are very small, but are designed to replace components much bigger and bulky In size. They process information very intelligently and efficiently. They sense the environment around them. The signals they gather are tuned into digital data that streams through tributaries of circuit lines at the speed of light. Inside the microprocessor collates and calculators. The software has middling intelligence. Then in a split second, the processed streams are shoved out.

A.)What is the primary difference between a microprocessor and a micro controller?

Unlike the microprocessor, the micro controller can be considered to be a true "Computer on a chip".In addition to the various features like the ALU, PC, SP and registers found on a microprocessor, the micro controller also incorporates features like the ROM, RAM, Ports, timers, clock circuits, counters, reset functions etc. While the microprocessor is more a general-purpose device, used for read, write and calculations on data, the micro controller, in addition to the above functions also controls the environment.

B.)The 8051

The 8051 developed and launched in the early 80`s, is one of the most popular micro controller in use today. It has a reasonably large amount of built in ROM and RAM. In addition it has the ability to access external memory.

The generic term `8x51` is used to define the device. The value of x defining the kind of ROM, i.e. x=0, indicates none, x=3, indicates mask ROM, x=7, indicates EPROM and x=9 indicates EEPROM or Flash.

3.3 ROM

The early 8051, namely the 8031 was designed without any ROM. This device could run only with external memory connected to it. Subsequent developments lead to the development of the PROM or the programmable ROM. This type had the disadvantage of being highly unreliable.

The next in line, was the EPROM or Erasable Programmable ROM. These devices used ultraviolet light erasable memory cells. Thus a program could be loaded, tested and erased using ultra violet rays. A new program could then be loaded again.

An improved EPROM was the EEPROM or the electrically erasable PROM. This does not require ultra violet rays, and memory can be cleared using circuits within the chip itself.

Finally there is the FLASH, which is an improvement over the EEPROM. While the terms EEPROM and flash are sometimes used interchangeably, the difference lies in the fact that flash erases the complete memory at one stroke, and not act on the individual cells. This results in reducing the time for erasure.

A.)Different microcontrollers in market.

- PIC One of the famous microcontrollers used in the industries. It is based on RISC Architecture which makes the microcontroller process faster than other microcontroller.
- INTEL These are the first to manufacture microcontrollers. These are not as sophisticated other microcontrollers but still the easiest one to learn.
- ATMEL Atmel's AVR microcontrollers are one of the most powerful in the embedded industry. This is the only microcontroller having 1kb of ram even the entry stage. But it is unfortunate that in India we are unable to find this kind of microcontroller.

B.)Intel 8051

Intel 8051 is CISC architecture which is easy to program in assembly language and also has a good support for High level languages.The memory of the microcontroller can be extended up to 64k.This microcontroller is one of the easiest microcontrollers to learn.The 8051 microcontroller is in the field for more than 20 years. There are lots of books and study materials are readily available for 8051.

C.)Derivatives

The best thing done by Intel is to give the designs of the 8051 microcontroller to everyone. So it is not the fact that Intel is the only manufacture for the 8051 there more than 20 manufactures, with each of minimum 20 models. Literally there are hundreds of models of 8051 microcontroller available in market to choose. Some of the major manufactures of 8051 are

- Atmel
- Philips

I) Philips

The Philips's 8051 derivatives has more number of features than in any microcontroller. The costs of the Philips microcontrollers are higher than the Atmel's which makes us to choose Atmel more often than Philips

II) Dallas

Dallas has made many revolutions in the semiconductor market. Dallas's 8051 derivative is the fastest one in the market. It works 3 times as fast as a 8051 can process. But we are unable to get more in India.

III) Atmel

These people were the one to master the flash devices. They are the cheapest microcontroller available in the market. Atmel's even introduced a 20pin variant of 8051 named 2051. The Atmel's 8051 derivatives can be got in India less than 70 rupees. There are lots of cheap programmers available in India for Atmel. So it is always good for students to stick with 8051 when you learn a new microcontroller.

3.4 Architecture

Architecture is must to learn because before learning new machine it is necessary to learn the capabilities of the machine. This is some thing like before learning about the car you cannot become a good driver. The architecture of the 8051 is given below.

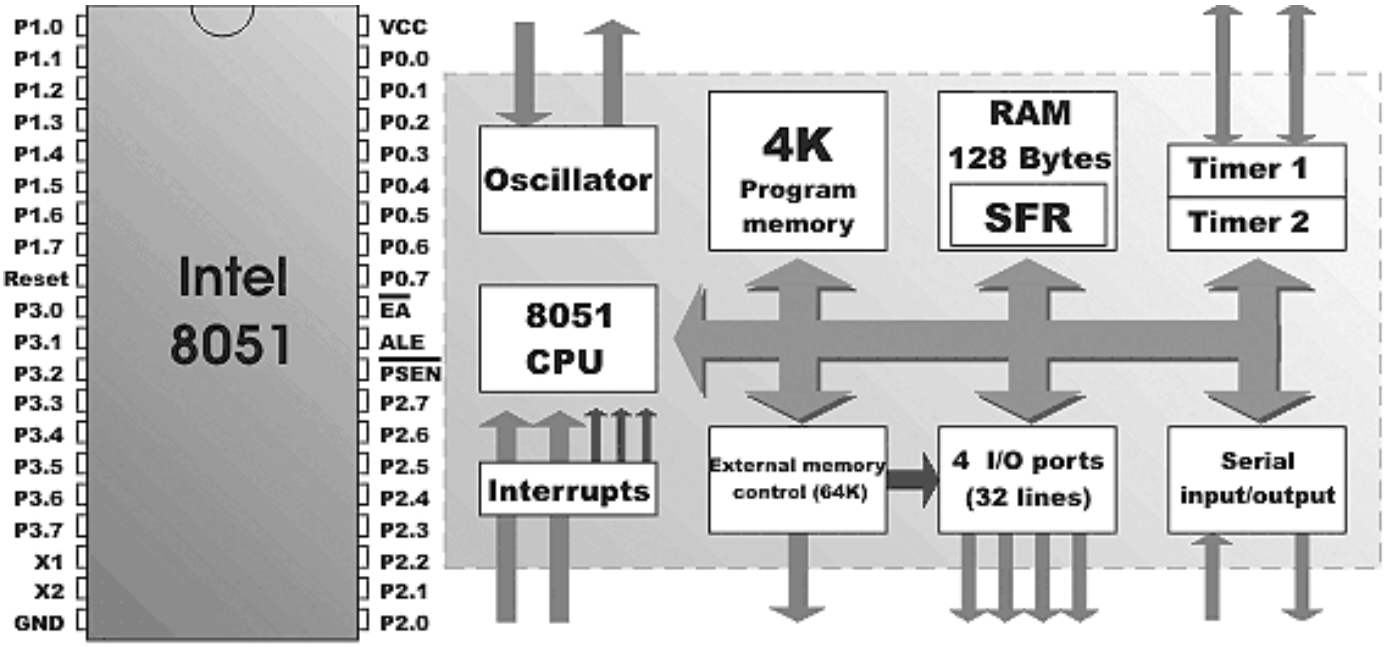


Figure 8-ARCHITECTURE

The 8051 doesn't have any special feature than other microcontroller. The only feature is that it is easy to learn. Architecture makes us to know about the hardware features of the microcontroller. The features of the 8051 are

- 4K Bytes of Flash Memory
- 128 x 8-Bit Internal RAM
- Fully Static Operation: 1 MHz to 24 MHz
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources (5 Vectored)
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

The 8051 has a 8-Bit CPU that means it is able to process 8 bit of data at a time. 8051 has 235 instructions. Some of the important registers and their functions are

Execution time

A.)clr p1.0

This is an assembly language instruction. It means we are instructing the microcontroller to put a value of 'zero' in bit zero of port one. This instruction is equivalent to telling the microcontroller to switch on the bulb. The instruction then to instruct the microcontroller to switch off the bulb is,

B.)Set p1.0

This instructs the microcontroller to put a value of 'one' in bit zero of port one.
There are a set of well defined instructions, which are used while communicating with the microcontroller. Each of these instructions requires a standard number of cycles to execute. The cycle could be one or more in number.

C.)How is this time then calculated?

The speed with which a microcontroller executes instructions is determined by what is known as the crystal speed. A crystal is a component connected externally to the microcontroller. The crystal has different values, and some of the used values are 6MHZ, 10MHZ, and 11.059 MHz etc.
Thus a 10MHZ crystal would pulse at the rate of 10,000,000 times per second.

The time is calculated using the formula

No of cycles per second = Crystal frequency in HZ / 12.

For a 10MHZ crystal the number of cycles would be,

10,000,000/12=833333.33333 cycles.

This means that in one second, the microcontroller would execute 833333.33333 cycles.

Therefore for one cycle, what would be the time? Try it out.

The instruction clr p1.0 would use one cycle to execute. Similarly, the instruction setb p1.0 also uses one cycle.

So go ahead and calculate what would be the number of cycles required to be executed to get a time of 30 seconds!

Getting back to our bulb example, all we would need to do is to instruct the microcontroller to carry out some instructions equivalent to a period of 30 seconds, like counting from zero upwards, then switch on the bulb, carry out instructions equivalent to 30 seconds and switch off the bulb.

Just put the whole thing in a loop, and you have a never ending on-off sequence.

Features of the 8051

1. 8-bit CPU.(Consisting of the 'A' and 'B' registers)

Most of the transactions within the microcontroller are carried out through the 'A' register, also known as the Accumulator. In addition all arithmetic functions are carried out generally in the 'A' register. There is another register known as the 'B' register, which is used exclusively for multiplication and division.

Thus an 8-bit notation would indicate that the maximum value that can be input into these registers is '11111111'. Puzzled?

The value is not decimal 111, 11,111! It represents a binary number, having an equivalent value of 'FF' in Hexadecimal and a value of 255 in decimal.

We shall read in more detail on the different numbering systems namely the Binary and Hexadecimal system in our next module.

2. 4K on-chip ROM

Once you have written out the instructions for the microcontroller, where do you put these instructions?

Obviously you would like these instructions to be safe, and not get deleted or changed during execution. Hence you would load it into the 'ROM'

The size of the program you write is bound to vary depending on the application, and the number of lines. The 8051 microcontroller gives you space to load up to 4K of program size into the internal ROM.

4K, that's all? Well just wait. You would be surprised at the amount of stuff you can load in this 4K of space.

Of course you could always extend the space by connecting to 64K of external ROM .

3. 128 bytes on-chip RAM

This is the space provided for executing the program in terms of moving data, storing data etc.

4. 32 I/O lines. (Four- 8 bit ports, labeled P0, P1, P2, P3)

In our bulb example, we used the notation p1.0. This means bit zero of port one. One bit controls one bulb.

Thus **port one** would have 8 bits. There are a total of four ports named p0, p1, p2, p3, giving a total of 32 lines. These lines can be used both as input or output.

5. Two 16 bit timers / counters.

A microcontroller normally executes one instruction at a time. However certain applications would require that some event has to be tracked independent of the main program.

The manufacturers have provided a solution, by providing two timers. These timers execute in the background independent of the main program. Once the required time has been reached, (remember the time calculations described above?), they can trigger a branch in the main program.

These timers can also be used as counters, so that they can count the number of events, and on reaching the required count, can cause a branch in the main program.

6. Full Duplex serial data receiver / transmitter.

The 8051 microcontroller is capable of communicating with external devices like the PC etc. Here data is sent in the form of bytes, at predefined speeds, also known as baud rates.The transmission is serial, in the sense, **one bit at a time**

7. 5- interrupt sources with two priority levels (Two external and three internal)

During the discussion on the timers, we had indicated that the timers can trigger a branch in the main program. However, what would we do in case we would like the microcontroller to take the branch, and then return back to the main program, without having to constantly check whether the required time / count has been reached?

This is where the interrupts come into play. These can be set to either the timers, or to some external events. Whenever the background program has reached the required criteria in terms of time or count or an external event, the branch is taken, and on completion of the branch, the control returns to the main program.

Priority levels indicate which interrupt is more important, and needs to be executed first in case two interrupts occur at the same time.

8. On-chip clock oscillator.

This represents the oscillator circuits within the microcontroller. Thus the hardware is reduced to just simply connecting an external crystal, to achieve the required pulsing rate.

PIN FUNCTION OF IC 89C51.

1 ***Supply*** *pin of this ic is pin no 40. Normally we apply a 5 volt regulated dc power supply to this pin. For this purpose either we use step down transformer power supply or we use 9 volt battery with 7805 regulator*

.

2 ***Ground*** *pin of this ic is pin no 20. Pin no 20 is normally connected to the ground pin (normally negative point of the power supply.*

3 ***XTAL*** *is connected to the pin no 18 and pin no 19 of this ic. The quartz crystal oscillator connected to XTAL1 and XTAL2 PIN. These pins also needs two capacitors of 30 pf value. One side of each capacitor is connected to crystal and other pis is connected to the ground point. Normally we connect a 12 MHz or 11.0592 MHz crystal with this ic.. But we use crystal upto 20 MHz to this pins*

4 ***RESET*** *PIN.. Pin no 9 is the reset pin of this ic.. It is an active high pin. On applying a high pulse to this pin, the micro controller will reset and terminate all activities. This is often referred to as a power on reset. The high pulse must*
be high for a minimum of 2 machine cycles before it is allowed to go low.

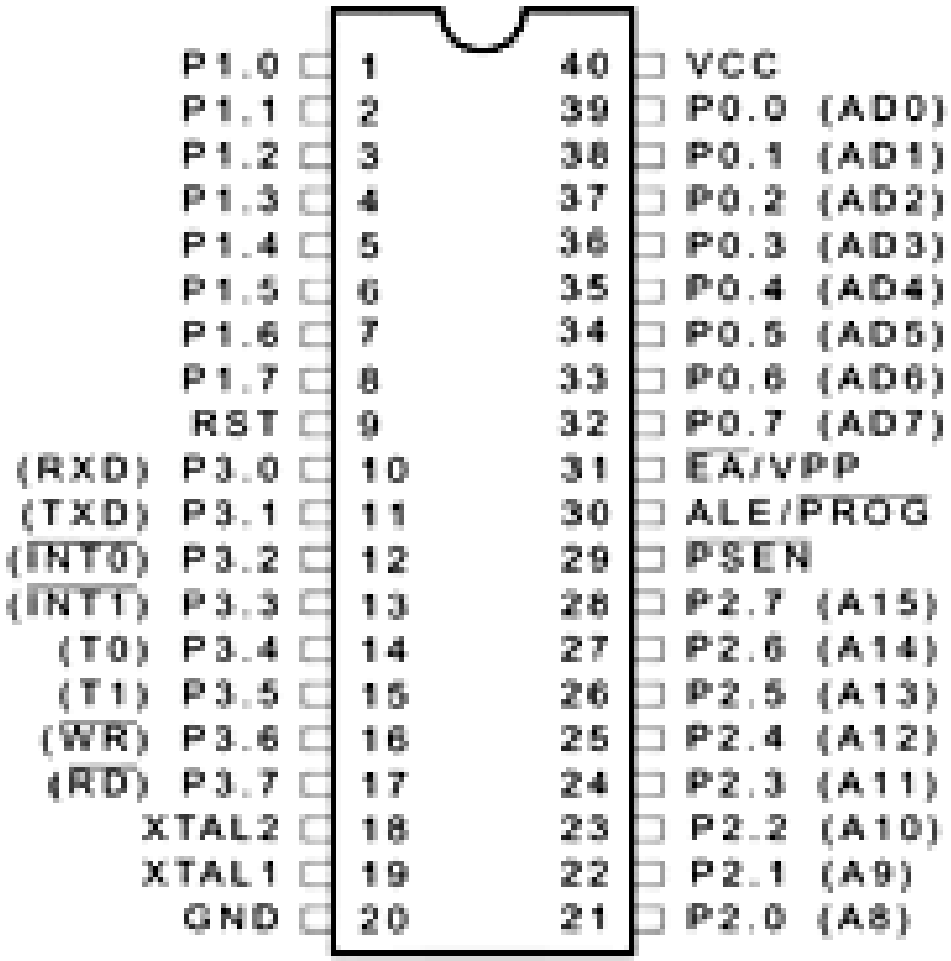
5. ***PORT0*** *Port 0 occupies a total of 8 pins.* ***Pin no 32 to pin no 39****. It can be used for input or output. We connect all the pins of the port 0 with the pullup resistor (10 k ohm) externally. This is due to fact that port 0 is an open drain mode. It is just like a open collector transistor.*

6. ***PORT1****. ALL the ports in micrcontroller is 8 bit wide* ***pin no 1 to pin no 8*** *because it is a 8 bit controller. All the main register and sfr all is mainly 8 bit wide. Port 1 is also occupies a 8 pins. But there is no need of pull up resistor in this port. Upon reset port 1 act as a input port. Upon reset all the ports act as a input port*

7. ***PORT2.*** *port 2 also have a 8 pins. It can be used as a input or output. There is no need of any pull up resistor to this pin.*

PORT 3. Port3 occupies a totoal 8 pins from pin no 10 to pin no 17. It can be used as input or output. Port 3 does not require any pull up resistor. The same as port 1 and port2. Port 3 is configured as an output port on reset. Port 3 has the additional function of providing some important signals such as interrupts. Port 3 also use for serial communication.

PIN DIAGRAM OF 89C51



AT89C51

ALE *ALE is an output pin and is active high. When connecting an 8031 to external memory, port 0 provides both address and data. In other words, the 8031 multiplexes address and data through port 0 to save pins. The ALE pin is used for demultiplexing the address and data by connecting to the ic 74ls373 chip.*

PSEN. *PSEN stands for program store eneable. In an 8031 based system in which an external rom holds the program code, this pin is connected to the OE pin of the rom.*

EA. *EA. In 89c51 8751 or any other family member of the ateml 89c51 series all come with on-chip rom to store programs, in such cases the EA pin is connected to the Vcc. For family member 8031 and 8032 is which there is no on chip rom, code is stored in external memory and this is fetched by 8031. In that case EA pin must be connected to GND pin to indicate that the code is stored externally.*

There are two ways to increase the baud rate of data transfer in the 8051

1. *To use a higher frequency crystal*
2. *To change a bit in the PCON register*

PCON register is an 8 bit register . Of the 8 bits, some are unused, and some are used for the power control capability of the 8051. the bit which is used for the serial communication is D7, the SMOD bit. When the 8051 is powered up, D7 (SMOD BIT) OF PCON register is zero. We can set it to high by software and thereby double the baud rate

IE (INTERRUPT ENABLE REGISTOR)

EA	---	ET2	ES	ET1	EX1	ETO	EXO

EA IE.7 Disable all interrupts if EA = 0, no interrupts is acknowledged
If EA is 1, each interrupt source is individually enabled or disbaled
By sending or clearing its enable bit.

IE.6 NOT implemented

ET2 IE.5 enables or disables timer 2 overflag in 89c52 only

ES IE.4 Enables or disables all serial interrupt

ET1 IE.3 Enables or Disables timer 1 overflow interrupt

EX1 IE.2 Enables or disables external interrupt

ET0 IE.1 Enables or Disbales timer 0 interrupt.

EX0 IE.0 Enables or Disables external interrupt 0

INTERRUPT PRIORITY REGISTER

---	---	PT2	PS	PT1	PX1	PTO	PXO

If the bit is 0, the corresponding interrupt has a lower priority and if the bit is 1 the corresponding interrupt has a higher priority

IP.7 NOT IMPLEMENTED, RESERVED FOR FUTURE USE.

IP.6 NOT IMPLEMENTED, RESERVED FOR FUTURE USE

PT2 IP.5 DEFINE THE TIMER 2 INTERRUPT PRIORITY LELVEL

PS IP.4 DEFINES THE SERIAL PORT INTERRUPT PRIORITY LEVEL

PT1 IP.3 DEFINES THE TIMER 1 INTERRUPT PRIORITY LEVEL

PX1 IP.2 DEFINES EXTERNAL INTERRUPT 1 PRIORITY LEVEL

PT0 IP.1 DEFINES THE TIMER 0 INTERRUPT PRIORITY LEVEL

PX0 IP.0 DEFINES THE EXTERNAL INTERRUPT 0 PRIORITY LEVEL

SCON: SERIAL PORT CONTROL REGISTER , BIT ADDRESSABLE

SCON

SM0	SM1	SM2	REN	TB8	RB8	T1	R1

SM0 : SCON.7 Serial Port mode specifier

SM1 : SCON.6 Serial Port mode specifier

SM2 : SCON.5

REN : SCON.4 Set/cleared by the software to Enable/disable reception

TB8 : SCON.3 The 9th bit that will be transmitted in modes 2 and 3, Set/cleared

By software

RB8 : SCON.2 In modes 2 &3, is the 9th data bit that was received. In mode 1,

If SM2 = 0, RB8 is the stop bit that was received. In mode 0 RB8 is not used

T1 : SCON.1 Transmit interrupt flag. Set by hardware at the end of the 8th bit

Time in mode 0, or at the beginning of the stop bit in the other

Modes. Must be cleared by software

R1 SCON.0 Receive interrupt flag. Set by hardware at the end of the 8th bit

Time in mode 0, or halfway through the stop bit time in the other

Modes. Must be cleared by the software.

exists which allows us to communicate with the vast 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used, the LCD The 44780 standard requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used, the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used, the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

The three control lines are referred to as **EN**, **RS**, and **RW**.

will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used, the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

The three control lines are referred to as **EN**, **RS**, and **RW**.

The **EN** line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should first set this line high (1) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring **EN** low (0) again. The 1-0 transition tells the 44780 to take the data currently found on the other control lines and on the data bus and to treat it as a command.

The **RS** line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

The **RW** line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands--so RW will almost always be low.

Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

Fortunately, a very popular standard exists which allows us to communicate with the vast majority of data from an external source (in this case, the 8051) and communicates directly with the LCD.

CHAPTER - 4

4.1 Program code for auto robot

```
Org 0000h
        Sjmp main

Org 0030h

Main:

        Clr p1.0
        Setb p1.1

        Clr p1.2
        Setb p1.3

        Jnb p3.0,left
        Jnb p3.1,right

        Sjmp main

Left:

        Setb p1.0
        Setb p1.0

        Clr p1.2
        Setb p1.3

Acall delay
```

```
Sjmp main


Right:

        Clr p1.0
        Setb p1.0

        Setb p1.2
        Setb p1.3

Acall delay
Sjmp main


Delay:

        Mov r0,#255
        Mov r1,#255
Sim:    djnz r0,sim
        djnz r1,sim
        ret

end
```

CHAPTER 5

5.1 CONCLUSION

In this project we showed how we control the movement of the robot with the help of any mobile phone using DTMF tones.

Use of a mobile phone for robotic control can overcome these limitations. It provides the advantages of robust control, working range as large as the coverage area of the service provider, no interference with other controllers and up to twelve controls.

In this project, the robot is so controlled by a mobile phone that makes a call to the mobile phone attached to the robot. In the course of a call, if any button is pressed, a tone corresponding to the button pressed is heard at the other end of the call. This tone is called 'dual tone multiple frequency' (DTMF) tone. The robot perceives this DTMF tone with the help of the phone stacked in the robot.

5.2 PROBLEMS ENCOUNTERED

- During soldering, many of the connection become short cktd. So we desolder the connection and did soldering again.
- LED`s get damaged when we switched ON the supply so we replace it by the new one.
- TROUBLESHOOT
- Care should be taken while soldering. There should be no shorting of joints.
- Proper power supply should maintain.

References:

1. 8051 and embedded system‖ by Mazidi and Mazidi

2. All datasheets from www.datasheetcatalog.com

3. About AT89s8252 from www.atmel.com

4. And www.triindia.co.in

5. About DS1820 from www.dallas.com