SELF STABILIZING AND OVER WEIGHT DETECTOR VEHICLE

Project Stage-I
Report submitted

Partial fulfillment of requirement
for the award of degree of

Bachelor of Engineering in
Electrical Engineering

Submitted by

Pavanekar Pankaj Jagannath	17EE29
Sawant Suyash Sanjay	17EE36
Varma Priyanshu Kamalkishor	17EE50
Ghanekar Shubhankar Girish	17EE47

Under The Guidance Of

PROF. IFTEKAR PATEL

Department of Electrical Engineering

Anjuman-I-Islam's Kalsekar Technical Campus, Panvel
Mumbai University, Mumbai

2020-21

SELF STABILIZING AND OVER WEIGHT DETECTOR VEHICLE

Project Stage-I
Report submitted

Partial fulfillment of requirement
for the award of degree of

Bachelor of Engineering in
Electrical Engineering

Submitted by

Pavanekar Pankaj Jagannath	17EE29
Sawant Suyash Sanjay	17EE36
Varma Priyanshu Kamalkishor	17EE50
Ghanekar Shubhankar Girish	17EE47

Under The Guidance Of

PROF. IFTEKAR PATEL

Department of Electrical Engineering

2020-21

CERTIFICATE

This is to certify that the dissertation titled "**Self Stabilizing And Over Weight Detector Vehicle**", which is beingsubmitted here with for the award of the, **'Bachlor of Engineering'** in **Electrical Engineering** of Anjuman-I-Islam's Kalsekar Technical Campus, New Panvel (M.S., India). This is the result of the original research work and contribution by **'Mr. Pankaj Jagannath Pavanekar, Mr. Suyash Sanjay Sawant, Mr. Priyanshu Kamalkishor Varma, Mr. Shubhankar Girish Ghanekar'** under my supervision and guidance. The work embodied in this dissertation has not formed earlier for the basis of award of any degree or compatible certificate or similar title of this for any other diploma/examining body or university to the best of knowledge and belief.

Place: Panvel
Date:

Prof. Iftekar Patel
Guide

Prof. Rizwan Farade
HOD

Dr. Abdul Razak Honnutagi
Director AIKTC

DECLARATION

I hereby declare that I have formed, completed and written the dissertation entitled "**Self Stabilizing And Over Weight Detector Vehicle**". It has not previously submitted for the basis of the award of any degree or diploma or either similar title of this for any other diploma/examining body/university.

Place: Panvel

Date:

(Signature)

Pankaj Pavanekar

(Signature)

Suyash Sawant

(Signature)

Priyanshu Varma

(Signature)

Shubhankar Ghanekar

ACKNOWLEDGMENT

We would like to express our sincere thanks to Prof. Iftekar Patel for taking time from his busy schedule to provide us with great deal of help, support and encouragement to work diligently at every aspect of our project

We are thankful to our college Director Dr. Abdul Razak Honnutagi, HOD Prof. Rizwan Farade and all staff members of Electrical department for providing us various facilities and guiding us whenever required. Finally, we would like to thank our parents andour friends for constantly supporting and encouraging our efforts.

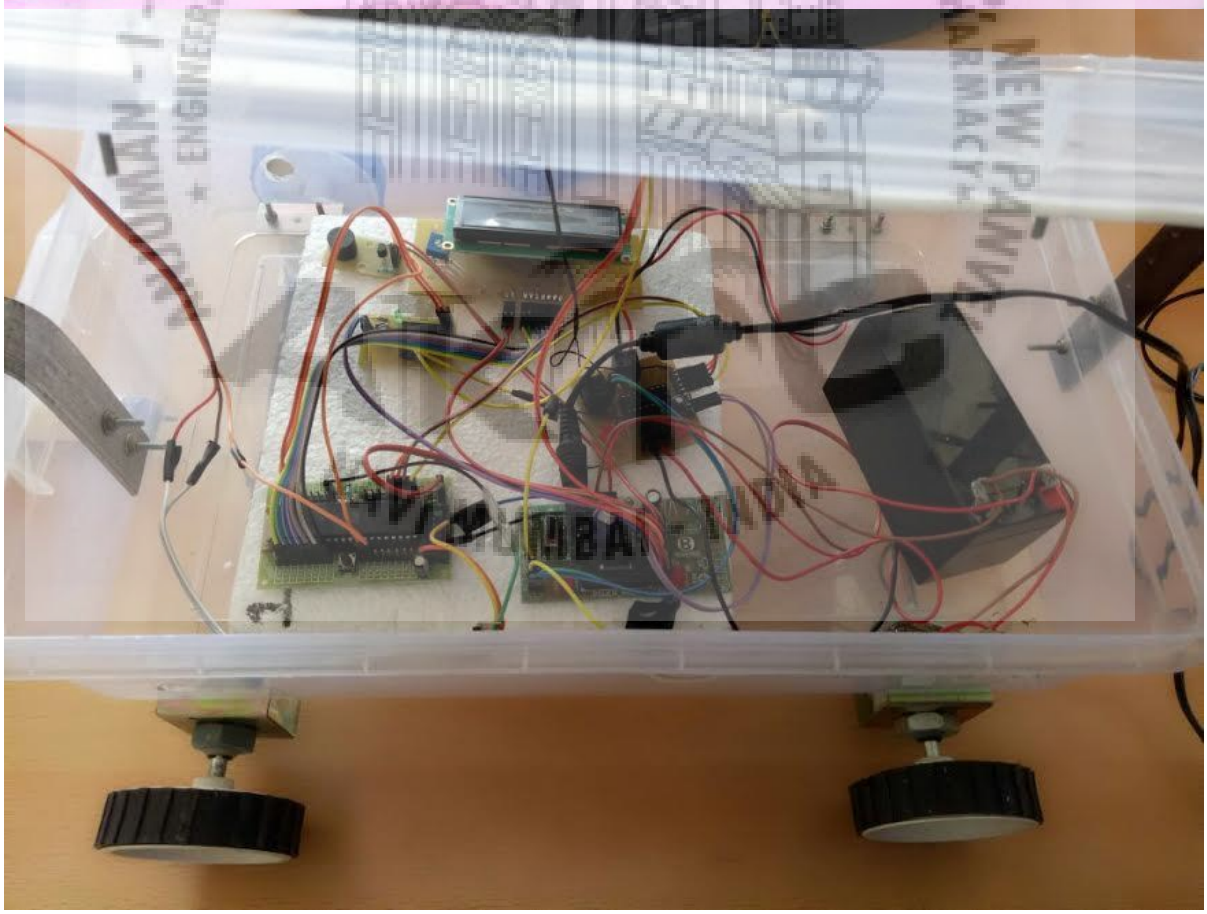
ABSTRACT

Basically self-stabilizing platform consists of platform which is balanced in motion by servomotor In opposite direction to the movement of the platform, motion is sensed by accelerometer. Microcontroller 89S52 process the digital signal obtained by the ADC and give instruction to the servo motor to rotate by certain angle depending on its previous position to balance or control the platform. The platform also consist of force sensor which sense the weight, the input analog signal(weight) is compared with the 10k resistor through voltage divider this signal is converted into digital signal by the ADC and process by the microcontroller and display on the LCD display and the X-axis of accelerometer is also displayed on the LCD display. We have kept the maximum limit 200 grams as the weight exceeds the maximum limit the piezo buzzer get on.The truck uses 2 DC motor which is driven by motor drive through Bluetooth(Bluetooth is connected to the RX pin of microcontroller.

CONTENTS:

SR NO	TOPIC	PAGE NO
1	INTRODUCTION	1
2	SCHEMATIC DIAGRAM	3
3	BLOCK DIAGRAM	6
4	PCB DIAGRAM	24
5	SOFTWARE INTERFACES	27
6	COMPONENT LIST	33
7	HARDWARE DISCRIPTION	35
8	PROGRAM	46
9	CONCLUSION	66
10	REFERENCE	68


I LOVE INDIA
TATA



CHAPTER-1

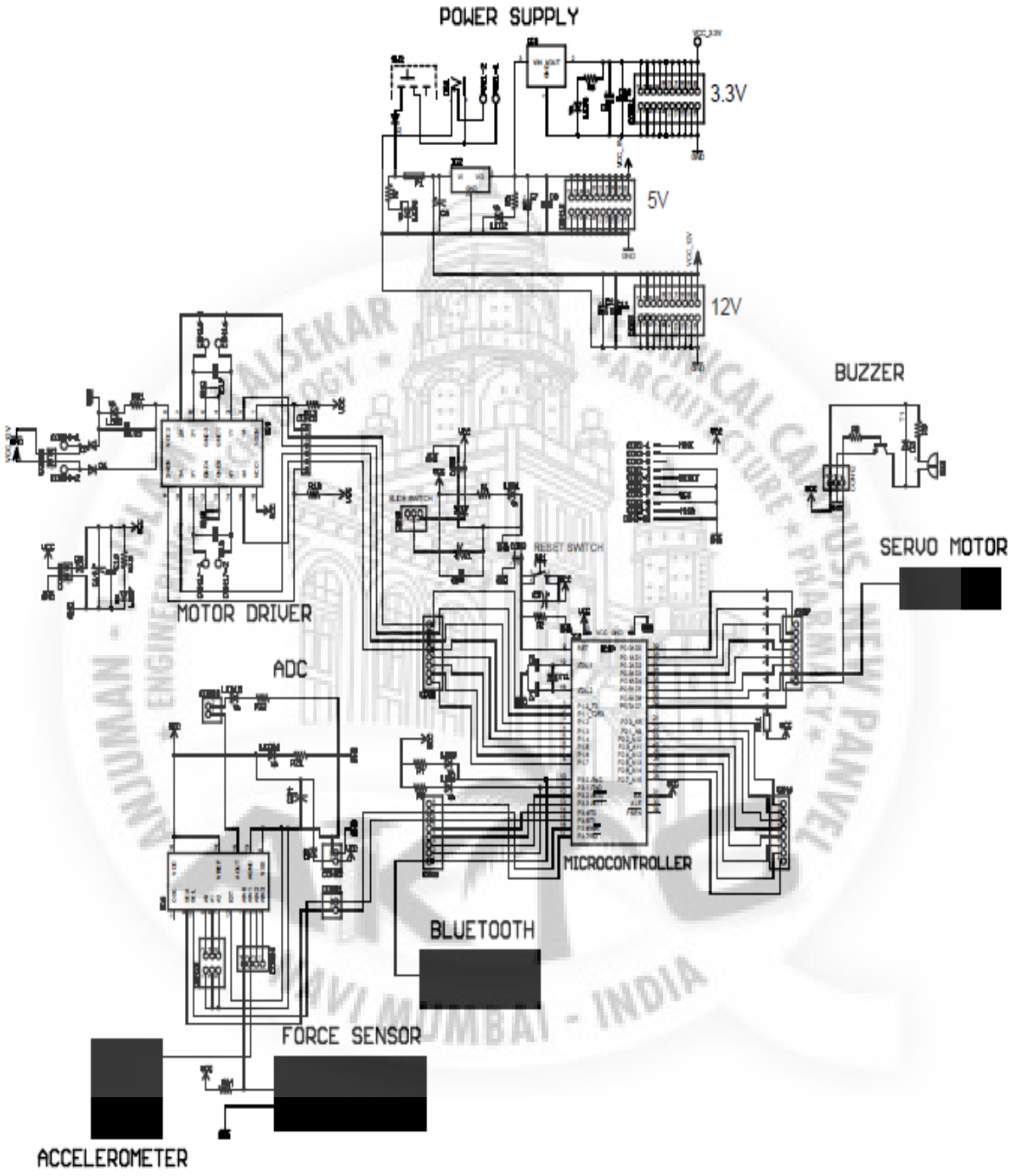
INTRODUCTION

Introduction:-

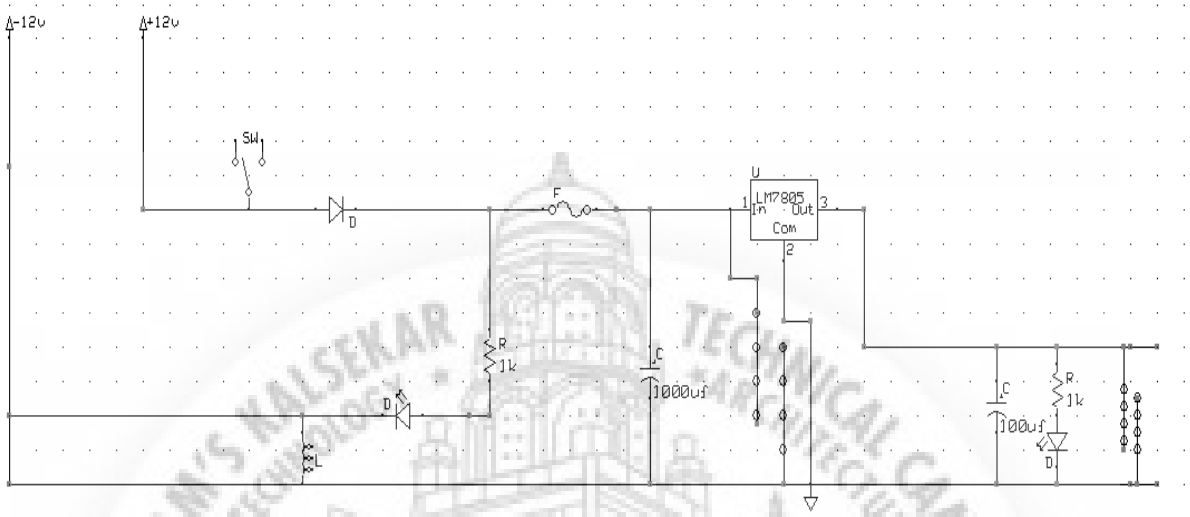
Today in day to day life the world is running with a great speed because of which people drives their car, truck, bikes carelessly due to with many breathtaking incident take place. In this project we had concentrated on the problem of the loading truck it basically focus on the incident take place due to the fall or movement of the heavy loads from the trucks and extra load carrying illegally. Because of this the dangerous accident which occurs due to the loaded truck can be minimized and trucks can easily pass through the potholes without misbalancing. It basically consist of servo motor which stabilizes the platform and microcontroller which give instructions to the servo motor and a weight measuring element which detects the extra load and starts the buzzer. As it will limit the material which has to be loaded in the truck.

CHAPTER-2
SCHEMATIC DIAGRAM

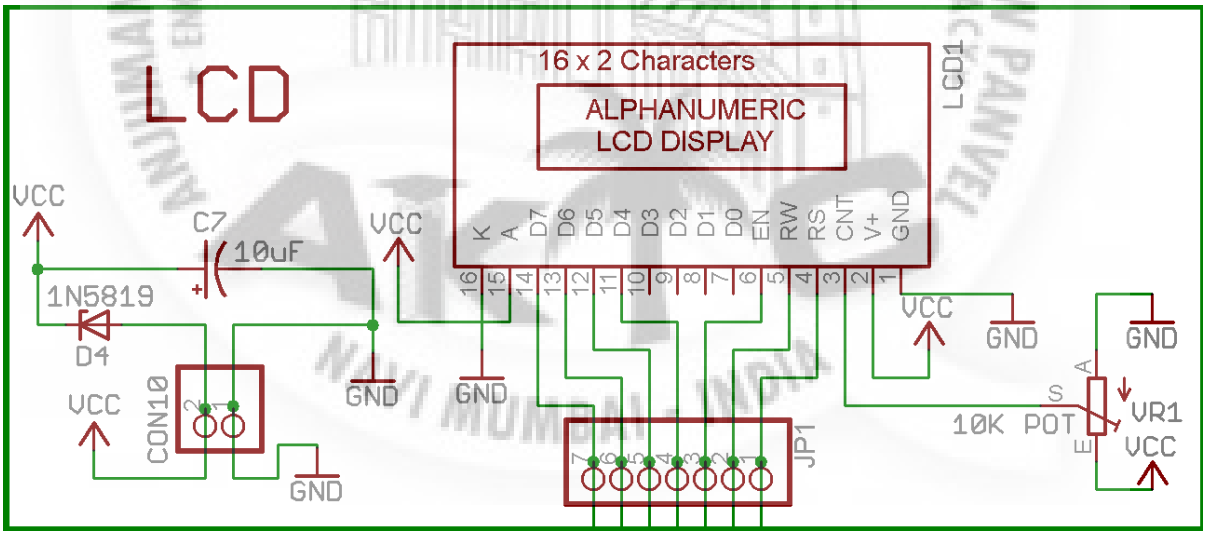
SchematicDiagram:-



POWERSUPPLY:-



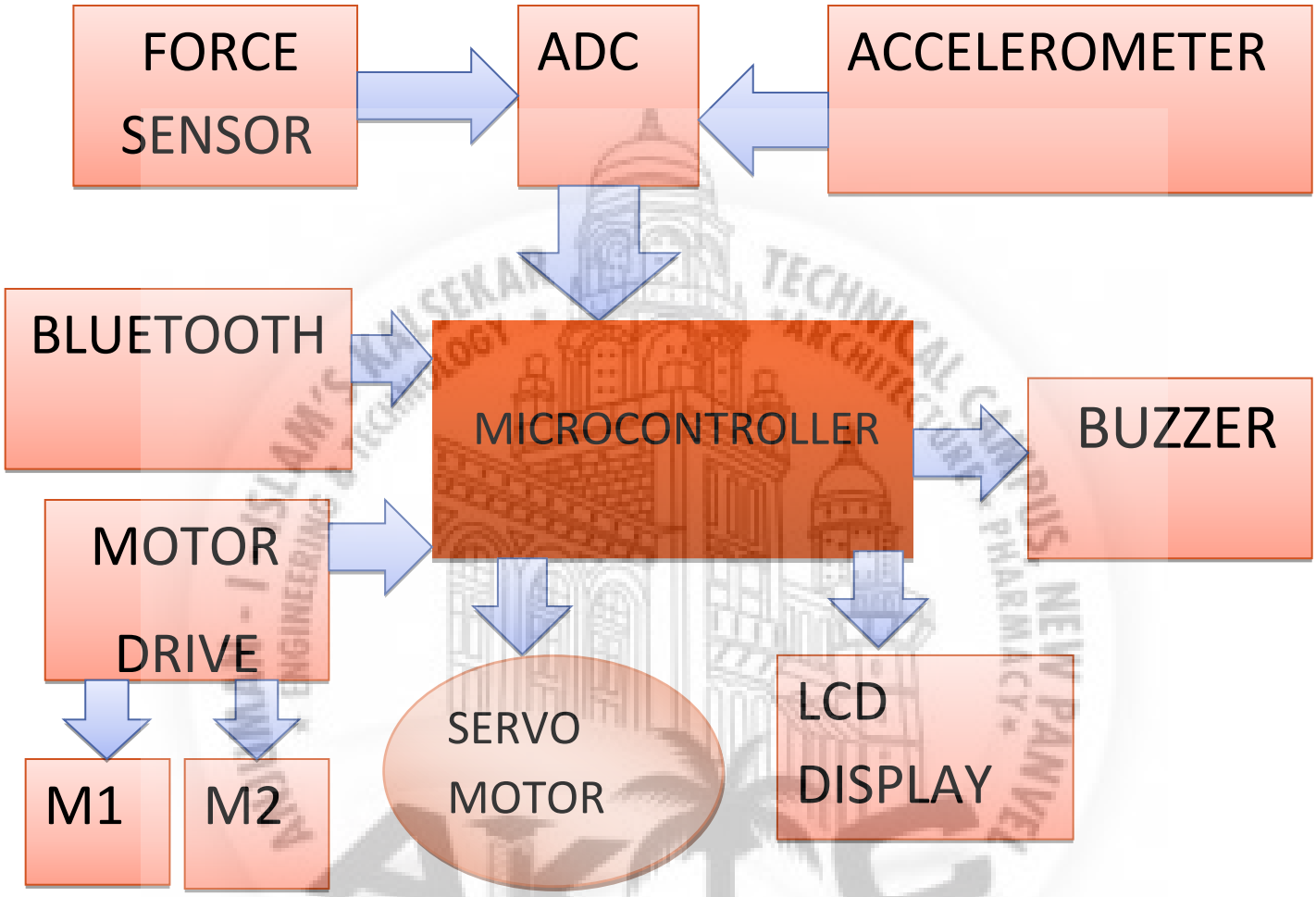
LCD:-



CHAPTER 3

BLOCK DIAGRAM

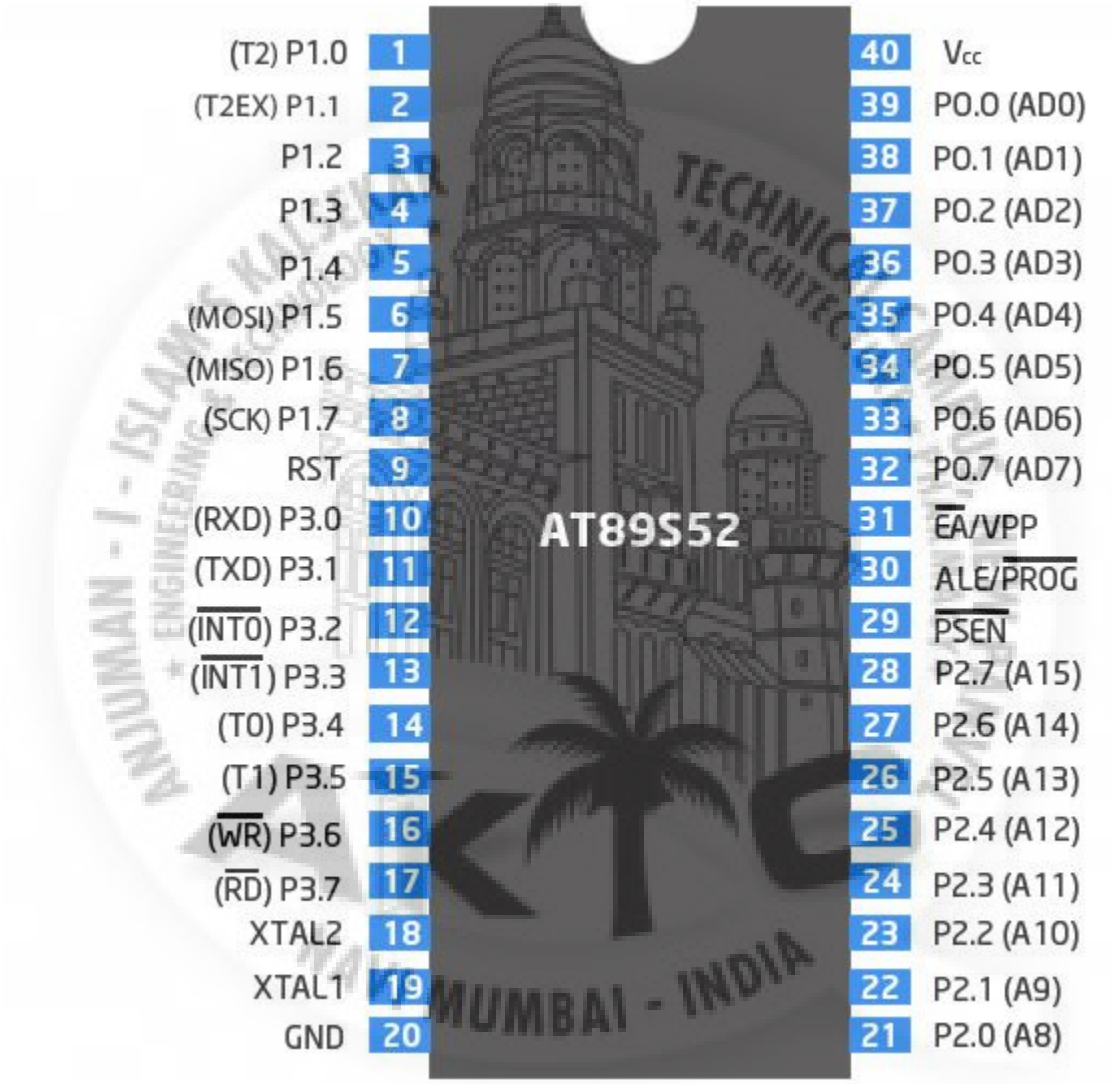
BLOCKDIAGRAM



BLOCK DESCRIPTION:-

MICROCONTROLLER-ATMEL 89S52:-

PIN CONFIGURATION



Specifications:-

- 8 Bit Micro-controller 8 bit ALU
- Operates on +5V DC _+0.5V tolerance
- 128 BYTE static RAMS
- 4 KB Internal RAM
- 32 Programmable I/O Lines
- 2 16-bit Timer/Counters
- 2 External Interrupt
- Full Duplex UART Serial Channel
- 8 General Purpose Registers
- SFR to control Ports/Timers/Interrupt/UART
- Available as 40 Pin DIP

Description:-

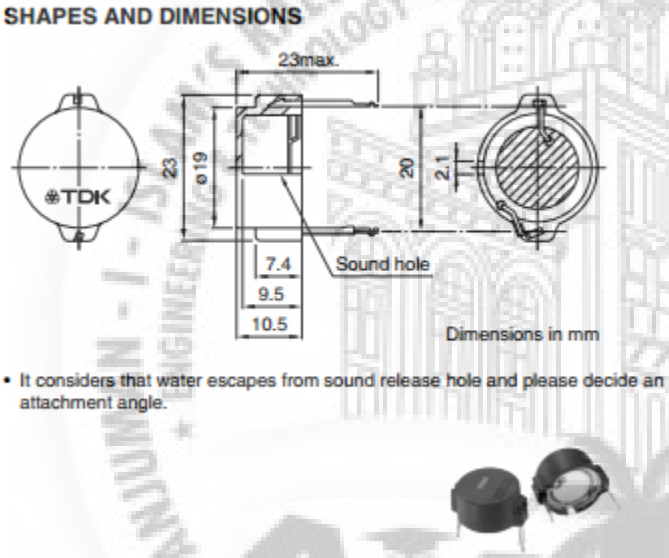
The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the Industry-standard 80C51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications. The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM con-tents but freezes the oscillator, disabling all other chip functions until the next interrupt.

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt)
P3.3	$\overline{\text{INT1}}$ (external interrupt)
P3.4	T0 (Timer/Counter 0 external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Buzzer:-

A buzzer or beeper is an audio signaling device, which may be mechanical, Electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke.

A signaling apparatus similar to an electric bell but without hammer or gong, producing a buzzing sound by the vibration of an armature.



Features:-

The PS series are high-performance buzzers that employ piezoelectric elements and are designed for easy incorporation into various circuits.

• They feature extremely low power consumption in comparison to electromagnetic units.

• Because these buzzers are designed for external excitation, the same part can serve as both a musical tone oscillator and a buzzer

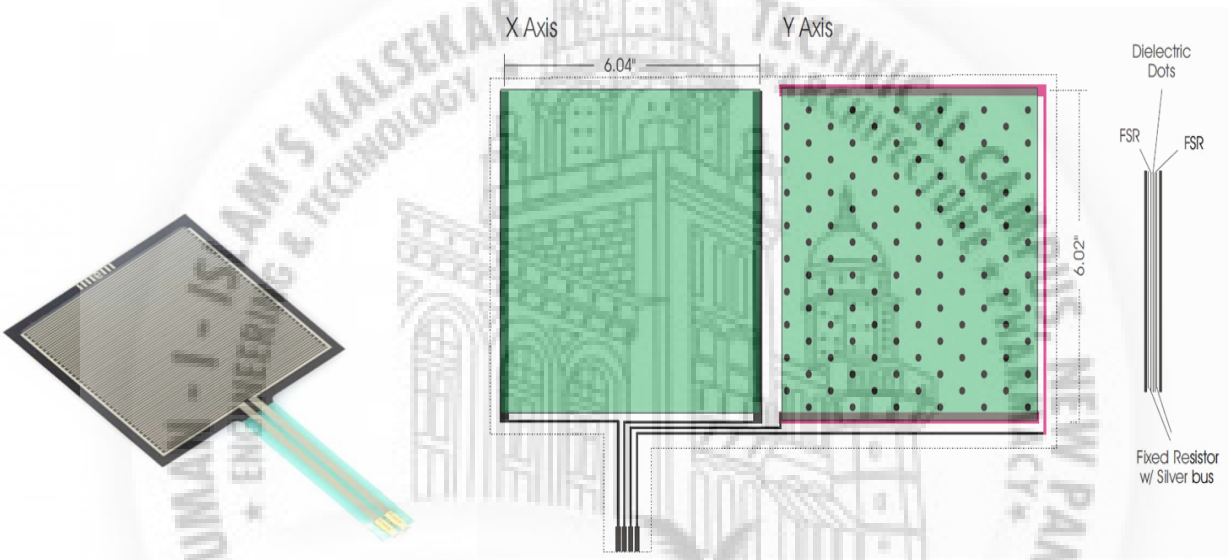
• They can be used with automated inserters.

• The lead wire type (PS1550L40N) with both-sided adhesive tape installed easily is prepared.

APPLICATIONS:-

Electric ranges, washing machines, computer terminals, various devices that require speech synthesis output

Force Sensitive Resistor – Square:



Description:-

This is a force sensitive resistor with a square, 1.75x1.5", sensing area. This FSR will vary its resistance depending on how much pressure is being applied to the sensing area. The harder the force, the lower the resistance. When no pressure is being applied to the FSR its resistance will be larger than 1MΩ. This FSR can sense applied force anywhere in the range of 100g-10kg.

Two pins extend from the bottom of the sensor with 0.1" pitch making it bread board friendly. There is a peel-and-stick rubber backing on the other side of the sensing area to mount the FSR

Application of FSR:

Force-sensing resistors are commonly used to create pressure sensing "buttons" and have applications in many fields, including musical instruments, car occupancy sensors, artificial limbs, Foot pronation systems and portable electronics.

ADC:-

SYMBOL	PIN	DESCRIPTION
AINO	1	analog inputs (A/D converter)
AIN1	2	
AIN2	3	
AIN3	4	
A0	5	hardware address
A1	6	
A2	7	
V_{SS}	8	negative supply voltage
SDA	9	I^2C-bus data input/output
SCL	10	I^2C-bus clock input
OSC	11	oscillator input/output
EXT	12	external/internal switch for oscillator input
AGND	13	analog ground
V_{REF}	14	voltage reference input
AOUT	15	analog output (D/A converter)
V_{DD}	16	positive supply voltage

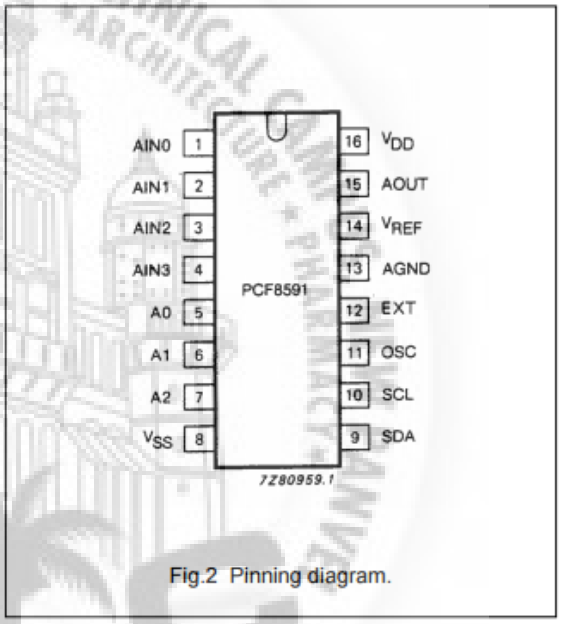


Fig.2 Pinning diagram.

The A/D converter makes use of the successive approximation conversion technique. The on-chip D/A converter and a high-gain comparator are used temporarily during an A/D conversion cycle. An A/D conversion cycle is always started after sending a valid read mode address to a PCF8591 device. The A/D conversion cycle is triggered at the trailing edge of the acknowledge clock pulse and is executed while transmitting the result of the previous conversion Once a conversion cycle is triggered an input voltage sample of the selected channel is stored on the chip and is converted to the corresponding 8-bit binary code. Samples picked up from differential inputs are converted to an 8-bit two's complement code .The conversion result is stored in the ADC data register and awaits transmission. If the auto-increment flag is set the next channel is selected.

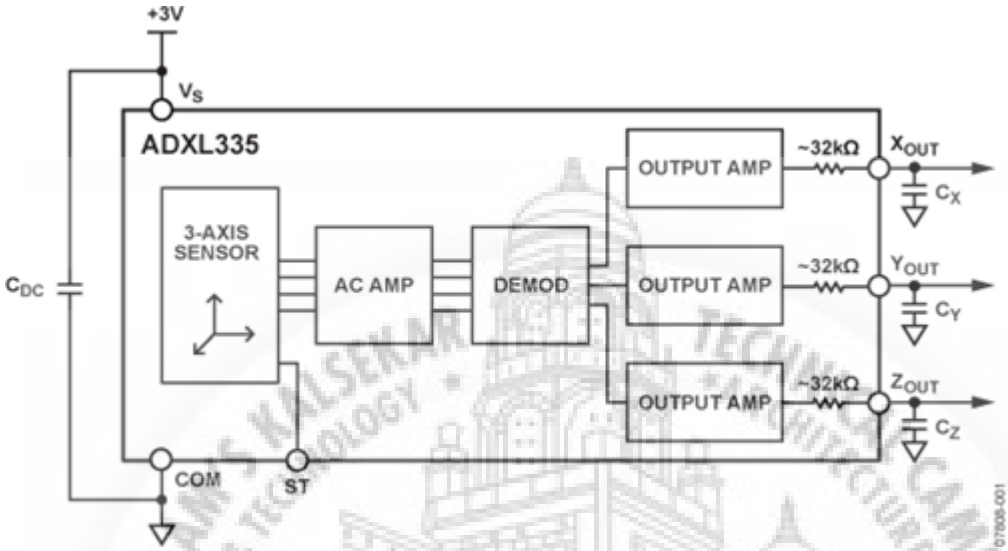
Application:-

- Music recording
- Digital signal processing
- Rotary encoder
- Scientific instrument

ADCFeatures:-

- Converts analog signal from Baseband system to digital signal
- Each analog baseband signal is sampled at the Nyquist rate of 32mhz rate
- Each sampler card digitizes the data coming from two analog channels i.e. Two polarizations signals of 16mhz BW each
- Provides necessary amplification of the signal
- The analog signal bandwidth can be handled by the adcs is -1V to +1V, peak top peak
- No user defined parameters thus no control signal required

Accelerometer :

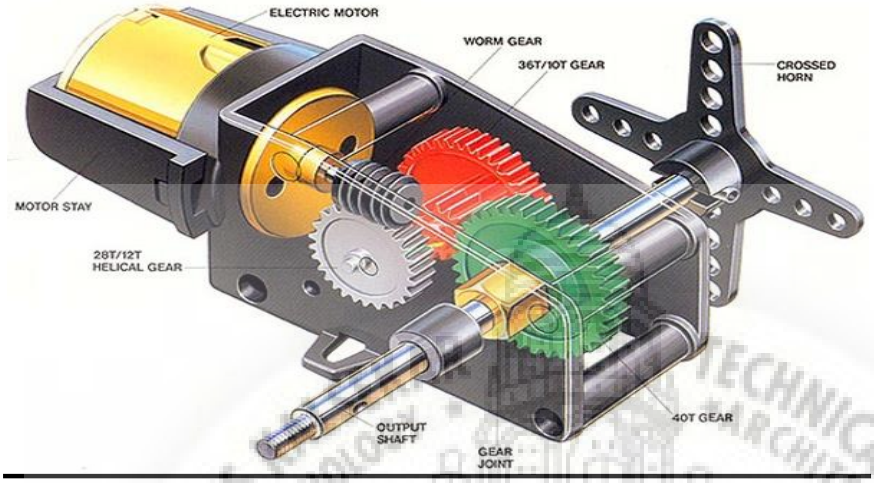


The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 g. It can measure the static acceleration of gravity in tilt sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

FeaturesandBenefits

- 3-axis sensing
- Small
- low-profile package
- 4 mm × 4 mm × 1.45 mm LFCSP
- Low power - 350 μA (typical)
- Single-supply operation
  1.8 V to 3.6 V
- 10,000 g shock survival
- Excellent temperature stability
- BW adjustment with a single capacitor per axis

Servomotor:-



A servo system mainly consists of three basic components - a controlled device, an output sensor, a feedback system. This is an automatic closed loop control system.

Working Principle of Servo Motor

A servo motor is basically a DC motor (in some special cases it is AC motor) along with some other special purpose components that make a DC motor a servo. In a servo unit, you will find a small DC motor, a potentiometer, gear arrangement and an intelligent circuitry. The intelligent circuitry along with the potentiometer makes the servo to rotate according to our wishes. As we know, a small DC motor will rotate with high speed but the torque generated by its rotation will not be enough to move even a light load. This is where the gear system inside a servomechanism comes into the picture. The gear mechanism will take high input speed of the motor (fast) and at the output; we will get an output speed which is slower than original input speed but more practical and widely applicable.

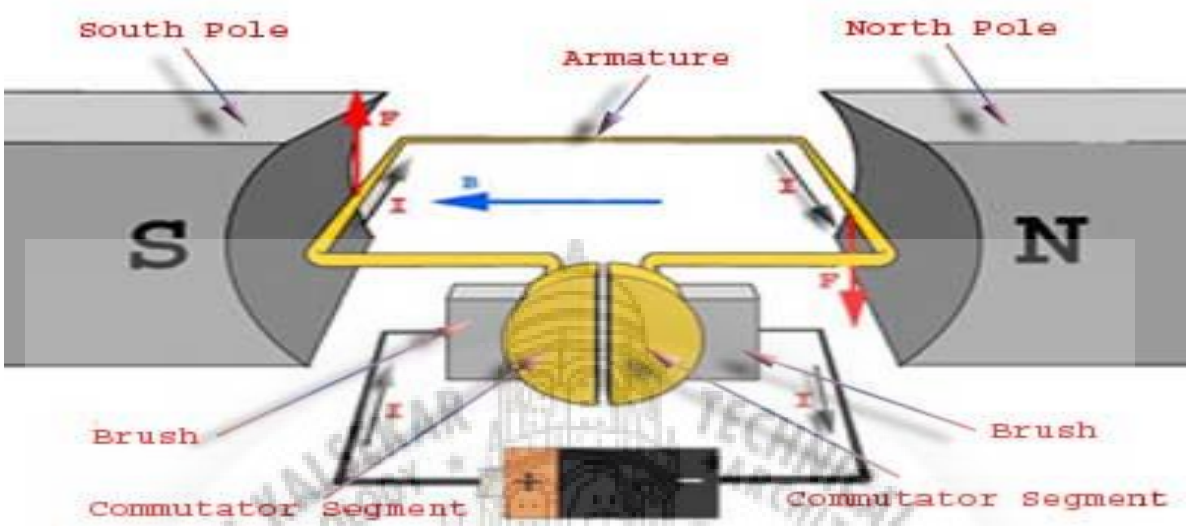
Application

- The servo motor is used in robotics to activate movements, giving the arm to its precise angle.
- The Servo motor is used to start, move and stop conveyor belts carrying the product along with many stages. For instance, product labeling, bottling and packaging
- The servo motor is built into the camera to correct a lens of the camera to improve out of focus images.
- The servo motor is used in robotic vehicle to control the robot wheels, producing plenty torque to move, start and stop the vehicle and control its speed

Features:-

- Minimum loss & high efficiency
- Extremely linear current &vs torque curve
- Big horsepower & compact construction
- High continuous torque output & at low speed range

DC Motor:-



Principle:-A motor is an electrical machine which converts electrical energy into mechanical energy. The principle of working of a DC motor is that "whenever a current carrying conductor is placed in a magnetic field, it experiences a mechanical force"

Applications:

1. Automotive
2. High-voltage power transmission
3. Telecommunication
4. Applications using fuel cells (mixing hydrogen and oxygen together with a catalyst to produce electricity and water as byproducts) also produce only DC.
5. Light aircraft electrical systems are typically 12 V or 20 V DC.

LCD display:-

LCD (liquid crystal display) is the technology used for displays in notebook and other smaller computers. Like light-emitting diode (LED) and gas-plasma technologies, LCDs allow displays to be much thinner than cathode ray tube (CRT) technology.

Working:-

Polarizing filter film with a vertical axis to polarize light as it enters.Glass substrate with ITO electrodes. The shapes of these electrodes will determine the shapes that will appear when the LCD is switched ON. Vertical ridges etched on the surface are smooth.

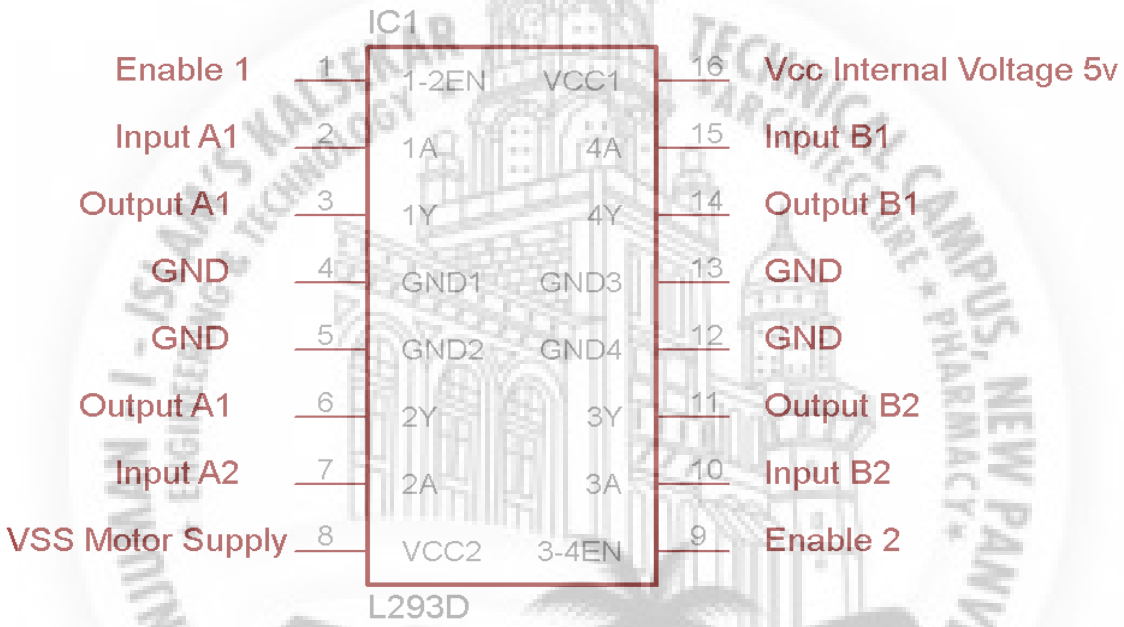
Twisted Pneumatic liquid crystal.Glass substrate with common electrode film (ITO) with horizontal ridges to line up with the horizontal filter.Polarizing filter film with a horizontal axis to block/pass light.Reflective surface to send light back to viewer. (In a backlit LCD, this layer is replaced with a light source.)

Advantages

- Very compact, thin and light, especially in comparison with bulky, heavy CRT displays.
- Little heat emitted during operation, due to low power consumption.
- No geometric distortion.
- The possible ability to have little or no "flicker" depending on backlight technology.
- Usually no refresh-rate flicker, because the LCD pixels hold their state between refreshes (which are usually done at 200 Hz or faster, regardless of the input refresh rate).
- Much thinner than a CRT monitor.
- Sharp image with no bleeding or smearing when operated at native resolution.
- Emits almost no undesirable electromagnetic radiation (in the extremely low frequency range), unlike a CRT monitor.
- Can be made in almost any size or shape.

Motor drive:-

L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control twoDC motor with a single L293D IC. Dual H-bridge Motor Driver integrated circuit (IC).



Working of L293D:-

There are 4 input pins for l293d, pin 2,7 on the left and pin 15 ,10 on the right as shown on the pin diagram. Left input pins will regulate the rotation of motor connected across left side and right input for motor on the right hand side. The motors are rotated on the basis of the inputs provided across the input pins as LOGIC 0 or LOGIC 1.

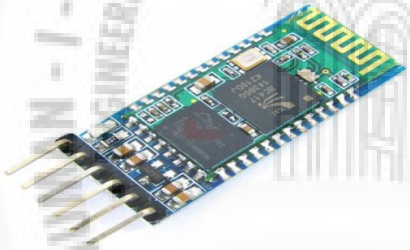
In simple way need to provide Logic 0 or 1 across the input pins for rotating the motor.

L293D Logic Table.

Lets consider a Motor connected on left side output pins (pin 3,6). For rotating the motor in clockwise direction the input pins has to be provided with Logic 1 and Logic 0.

- Pin 2 = Logic 1 and Pin 7 = Logic 0 | Clockwise Direction
- Pin 2 = Logic 0 and Pin 7 = Logic 1 | Anticlockwise Direction
- Pin 2 = Logic 0 and Pin 7 = Logic 0 | Idle [No rotation] [Hi-Impedance state]
- Pin 2 = Logic 1 and Pin 7 = Logic 1 | Idle [No rotation]

BLUETOOTH:-



HC-05 embedded Bluetooth serial communication module (can be short for module) has two work modes: order-response work mode and automatic connection work mode. And there are three work roles (Master, Slave and Loopback) at the automatic connection work mode. When the module is at the automatic connection work mode, it will follow the default way set lastly to transmit the data automatically.

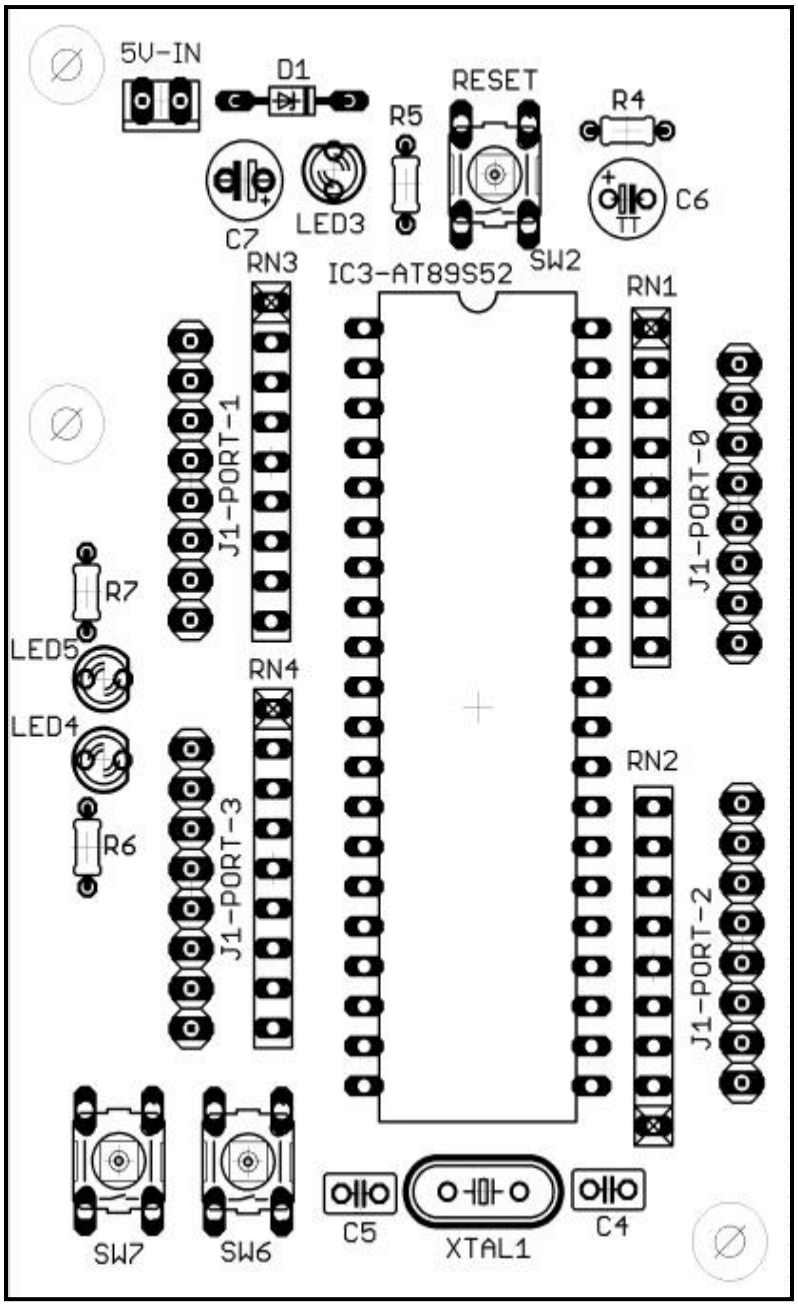
When the module is at the order-response work mode, user can send the AT command to the module to set the control parameters and sent control order. The work mode of module can be switched by controlling the module PIN (PIO11) input level.

Advantages:-

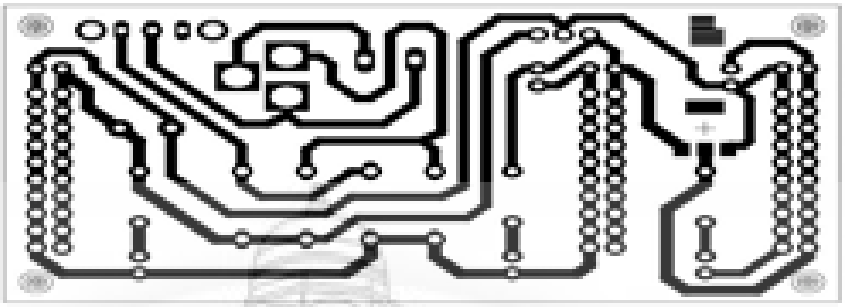
- It is wireless
- It is inexpensive
- It is automatic
- It is standardized
- It has low interference
- It uses low energy

CHAPTER 4
PCB DIAGRAM

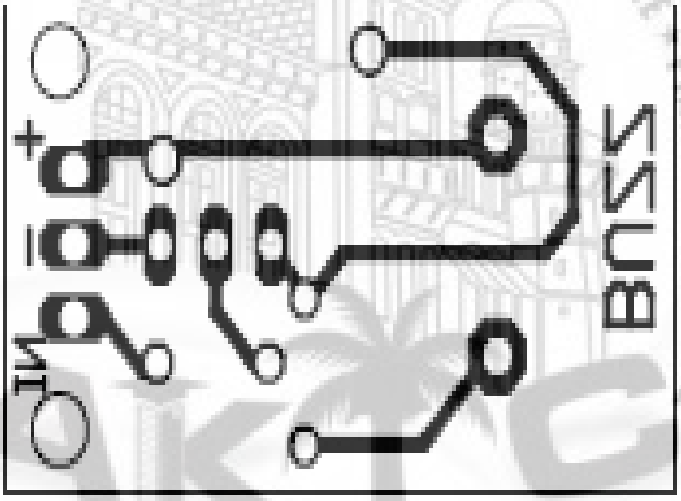
PCB DIAGRAM OF MICROCONTROLLER :



power supply.



buzzer.



CHAPTER 5

SOFTWARE INTERFACES

EMBEDDED C:

- Looking around, we find ourselves to be surrounded by various types of embedded systems. Be it a digital camera or a mobile phone or a washing machine, all of them has some kind of processor functioning inside it.
- Associated with each processor is the embedded software. If hardware forms the body of an embedded system, embedded processor acts as the brain, and embedded software forms its soul.
- It is the embedded software which primarily governs the functioning of embedded systems.
- Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems.
- Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.
- In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to.
- It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing.
- Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, datatype declaration, conditional statements (if, switch, case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.
- A Technical Report was published in 2004[1] and a second revision in 2006.

Advantage:-

- It is small and simpler to learn, understand, program and debug.
- Compared to assembly language, C code written is more reliable and scalable, more portable between different platforms.
- C compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.
- Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems.
- As C combines functionality of assembly language and features of high level languages, C is treated as a 'middle-level computer language' or 'high level assembly language'.
- It is fairly efficient.
- It supports access to I/O and provides ease of management of large embedded projects.
- Java is also used in many embedded systems but Java programs require the Java Virtual Machine (JVM), which consumes a lot of resources. Hence it is not used for smaller embedded devices.
- In Embedded we need read/write data on given address, so in C its Easy to access and modify, because of its pointer concept

Features of Embedded programming:

- code speed:- ode speed is governed by the processing power, timing constraints.
- code size :-code size is governed by available program memory and use of programming language.

KEILCompiler:-

The Keil C51 Cross Compiler is an ANSI C Compiler that is written specifically to generate fast, compact code for the 8051 microcontroller family. The C51 Compiler generates object code that matches the efficiency and speed of assembly programming.

advantages of C over assembly language programming:

1. Knowledge of the processor instruction set is not required. Rudimentary knowledge of the memory structure of the 8051 CPU is desirable.

2. Details like register allocation and addressing of the various memory types and data types is managed by the compiler.

3. Programs get a format structure (which is imposed by the C programming language) and can be divided into separate functions. This contributes to source code reusability as well as better overall application structure.

4. The ability to combine variable selection with specific operations improves program readability.

5. Keywords and operational functions that more nearly resemble the humans thought process may be used.

6. Programming and program test time is drastically reduced.

7. The C run-time library contains many standard routines such as: formatted output, numeric conversions, and floating –point arithmetic.

8. Existing program parts can be more easily included into new programs because of modular program construction techniques.

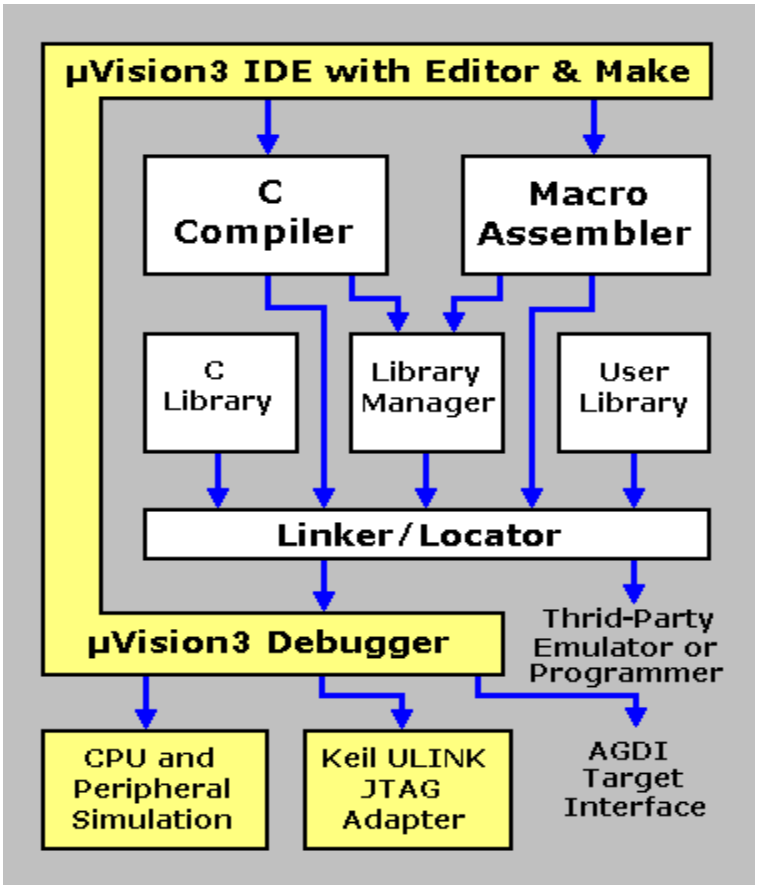
9. The language C is a very portable language (based on the ANSI standard) that enjoys wide popular support and is easily obtained for most systems. Existing program investment s can be quickly adapted to other processors as needed.

Software Development Cycle in KEIL :-

When we use Keil software tools, the project development cycle is roughly the same as it is for any other software development project.

1. Create a project, select the target chip from the device database, and configure the toll settings.
2. Create source file in C or assembly.
3. Build your application with the project manager.
4. Correct errors in the source files.
5.Test the linked application. A block diagram of the complete 8051 tool set may best illustrate the development cycle.

Block diagram:



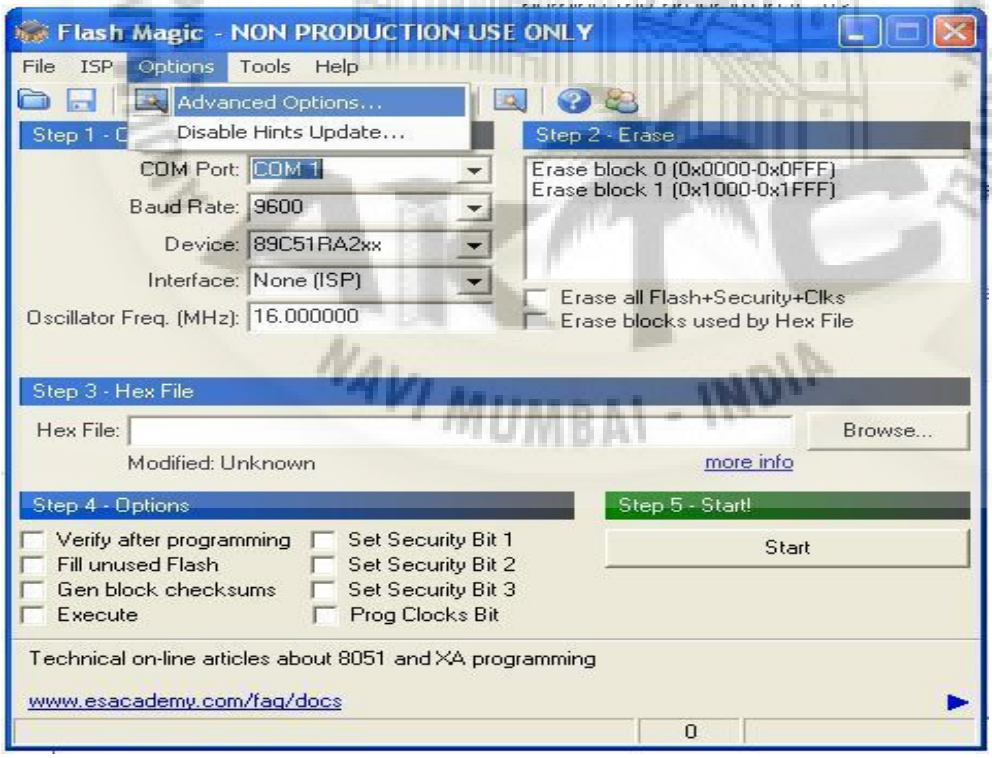
FLASH MAGIC SOFTWARE :-

The following is a screenshot of the main Flash Magic window. The apperance may differ slightly depending on the device selected.

The window is divided up into five sections. Work your way from section 1 to section 5 to program a device using the most common functions. Each section is described in detail in the following sections At the very bottom left of the window is an area where progress messages will be displayed and at the very bottom right is where the progress bar is displayed. In between the messages and the progress bar is a count of the number of times the currently selected hex file has been programmed since it was last modified or selected.

Just above the progress information EmbeddedHints are displayed. These are rotating Internet links that you can click on to go to a web page using your default browser.

.



CHAPTER 6

COMPONENT LIST

COMPONENT	QUANTITY	COST
POWER SUPPLY		
Resistor: 1) 10k 2) 1k	2	4
Capacitor:-100uf	2	3
Diode	1	2
LED	1	2
Fuse	1	15
Switch	1	8
IC 7805	1	15
Berg strip 10 pin 20 pin	2	15
A/D CONVERTER (PCF 8591)		**150**
Resistor-270Ω	1	2
LED	1	2
Capacitor: 10 uf	1	3
Diode	1	2
PIEZZO BUZZER		**15**
Resistor: 1)50Ω 2)1kΩ	2	4
IC 747B	1	15
LCD DISPLAY		**100**
MOTOR DRIVE		75
Capacitor:-100nF	4	3
Diode	3	2
LED	2	2
Resistor: 10k	2	2
MICROCONTROLLER(89S52)		**50**
Resistor:-1kΩ	2	5
LED	3	2
Diode	1	2
Switch	1	8
Capacitor: 1uf	1	3
BLUETOOTH(HC-05)	1	**400**
FORCE SENSOR	1**750**	
TOTAL COST		**RS 1659**

CHAPTER 7

HARDWARE DESCRIPTION

HARDWARE DESCRIPTION:

Switch :

Fig.push button switch

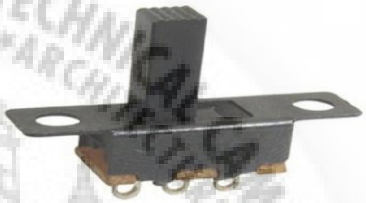


Fig. slide button switch

- a **switch** is an electrical component that can break an electrical circuit, interrupting the current or diverting it from one conductor to another
- A **push-button** (also spelled **pushbutton**) or simply **button** is a mechanism for simple switch controlling some aspect of a machine or a process. Buttons are typically made out of hard material, usually plastic or metal.

Capacitor:-

- **Type of capacitor**
- **Electrolytic capacitor**
- **Ceramic capacitor**

Electrolytic capacitor:-

A capacitor consists of two conductors separated by a non-conductive region. The non-conductive region is called the dielectric. In simpler terms, the dielectric is just an electricalinsulator.

Fig. electrolytic capacitor

Examples of dielectric media are glass, air, paper, vacuum, and even a semiconductor depletion chemically identical to the conductors. A capacitor is assumed to be self-contained and isolated, with no electric charge and no influence from any external electric field. The conductors thus hold equal and opposite charges on their facing surfaces,[11] and the dielectric develops an electric field. In SI units, a capacitance of one farad means that one coulomb of charge on each conductor causes a voltage of one voltage cross the device.

Ceramic capacitor :

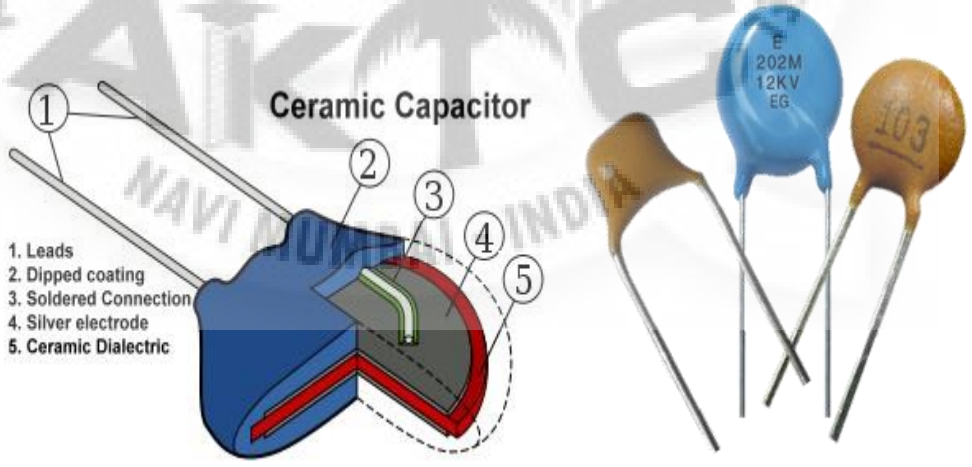


Fig.ceramic capacitor

A ceramic capacitor uses a ceramic material as the dielectric. Ceramics were one of the first materials to be used in the production of capacitors, as it was a known insulator. Many geometries were used in ceramic capacitors, of which some, like ceramic tubular capacitors and barrier layer capacitors are obsolete today due to their size, parasitic effects or electrical characteristics. The types of ceramic capacitors most often used in modern electronics are the multi-layer ceramic capacitor, otherwise named ceramic multi-layer chip capacitor (MLCC) and the ceramic disc capacitor. MLCCs are the most produced capacitors with a quantity of approximately 1000 billion devices per year. They are made in SMD (surface-mounted) technology and are widely used due to their small size. Ceramic capacitors are usually made with very small capacitance values, typically between 1nF and 1µF, although values up to 100µF are possible. Ceramic

capacitors are also very small in size and have a low maximum rated voltage. They are not polarized, which means that they may be safely connected to an AC source. Ceramic capacitors have a great frequency response due to low parasitic effects such as resistance or inductance.

➢ CERAMIC CAPACITOR CODE TABLE

Picofarad (pF)	Nanofarad (nF)	Microfarad (uF)	Code	Picofarad (pF)	Nanofarad (nF)
10	0.01	0.00001	100	4700	4.7
15	0.015	0.000015	150	5000	5
22	0.022	0.000022	220	5600	5.6
33	0.033	0.000033	330	6800	6.8
47	0.047	0.000047	470	10000	10
100	0.1	0.0001	101	15000	15
120	0.12	0.00012	121	22000	22
130	0.13	0.00013	131	33000	33
150	0.15	0.00015	151	47000	47
180	0.18	0.00018	181	68000	68
220	0.22	0.00022	221	100000	100
330	0.33	0.00033	331	150000	150
470	0.47	0.00047	471	200000	200
560	0.56	0.00056	561	220000	220
680	0.68	0.00068	681	330000	330
750	0.75	0.00075	751	470000	470
820	0.82	0.00082	821	680000	680
1000	1	0.001	102	1000000	1000
1500	1.5	0.0015	152	1500000	1500
2000	2	0.002	202	2000000	2000
2200	2.2	0.0022	222	2200000	2200
3300	3.3	0.0033	332	3300000	3300

IC AT89S52 :-

Fig. IC AT89S52

The **Atmel AT89 series** is an Intel 8051-compatible family of 8 bit microcontrollers(µCs) manufactured by the Atmel Corporation.

based on the Intel 8051 core, the AT89 series remains very popular as general purpose microcontrollers, due to their industry standard instruction set, and low unit cost. This allows a great amount of legacy code to be reused without modification in new applications. While considerably less powerful than the newer AT90 series of AVR RISC microcontrollers, new product development has continued with the AT89 series for the aforementioned advantages.

More recently, the AT89 series has been augmented with 8051-cored special function microcontrollers, specifically in the areas of USB, I²C (two wire interface), SPI and CAN bus controllers, MP3decoders and hardware PWM.

All four ports in the AT89C51 and AT89C52 are bidirectional.Each consists of a latch (Special Function Registers P0 through P3), an output driver, and an input buffer.

Theoutput drivers of Ports 0 and 2, and the input buffers of Port 0, are used in accesses to external memory.

In this application, Port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise the Port 2 pins continue to emit the P2 SFR content. All the Port 3 pins, and two Port 1 pins (in the AT89C52)aremultifunctional. The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise the port pin is stuck at 0. It has less complex feature than other microprocessor

Connector (dc-socket):-

fig. connector (dc socket)

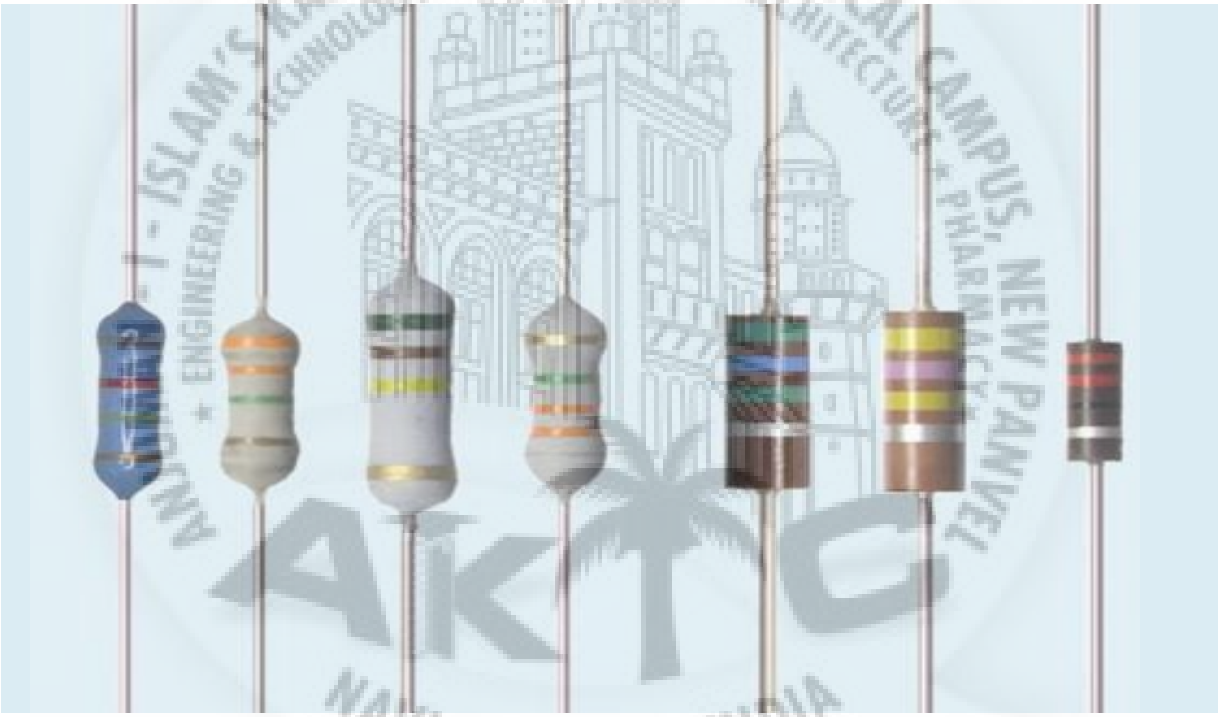
Small cylindrical connectors come in a variety of sizes. They may be known as "coaxial power connectors", "barrel connectors", "concentric barrel connectors" or "tip connectors". The intended use of these plugs is on the cable connected to a power supply.

Resistor :

A **resistor** is a passive two electrical that implements electrical resistance as a circuit element. The current through a resistor is in direct proportion to the voltage across the resistor's terminals. This relationship is represented by Ohm's law:

$$I = \frac{V}{R}$$

R is the resistance of the conductor in units of ohms.



Crystal :

Crystal is a solid in which the constituent atoms, molecules, or ions are packed in a regularly ordered, repeating pattern extending in all three spatial dimensionsAlmost any object made of an elastic material could be used like a crystal, with appropriatetransducers, since all objects have natural resonant frequencies of vibration. The resonant frequency dependson size, shape, elasticity, and the speed of sound in the material. We have used an 11.0592MHz crystal forboth the P89V51RD2FN and AT89S52

Features & application

- 3.3V Operation HCMOS Output Standby Function Tape and Reel (2,000 pcs. STD)
- Frequency Range (Fo)
- Storage Temperature Range (Tstg)
- Supply Voltage (Vdd)
  - Input Current (Idd) ~ 20.000 MHz ~ 40.000 MHz ~ 60.000 MHz ~ 75.000 MHz
  - Output Symmetry (50% Vdd) Rise Time ~ 90% Vdd)(Tr) Fall Time ~ 10% Vdd)(Tf)
  - Output Voltage (Vole) (Voh)
  - Output Current (Iol) (Ioh)

LED :

When a **light emitting diode** is forward-biased (switched on), electrons are able to recombine with electron holes within the device, releasing energy in the form of photons.

This effect is called **electroluminescence** and the color of the light (corresponding to the energy of the photon) is determined by the energy gap of the semiconductor. An **LED** is often small in area (less than 1 mm^2), and integrated optical components may be used to shape its radiation pattern.

The LED consists of a chip of semiconducting material doped with impurities to create a p-n junction. As in other diodes, current flows easily from the p-side, or anode, to the n-side, or cathode, but not in the reverse direction. Charge-carrierselectrons and holesflowinto the junction from electrodes with different voltages. When an electron meets a hole, it falls into a lower energy level, and releases energy in the form of a photon.

CHAPTER 8

PROGRAM

```
#include <REG52.H>
#define ALCD_PORT P2
sbit EN=P2^3;
unsigned char STRING_out[5];
voiddelay_ms(unsigned intdelayms);
voidALCD_delay_ms(unsigned intdelayms);
void ALCD_DATA(unsigned char data_to_lcd);
void ALCD_CMD(unsigned char data_to_lcd);
void ALCD_INIT();
void ALCD_CLEAR();
void ALCD_GOTO(unsigned char display_no,unsigned char line_no);
void ALCD_STRING(unsigned char *local_string);
unsigned char ADC_CALL(unsigned char);
sbit SDA=P3^6;
sbit SCL=P3^5;
///////////////I2C DELAY//////////////////////
void I2C_DELAY(unsigned int I2C_delayms);
void I2C_CLOCK();
void I2C_START();
void I2C_STOP();
void I2C_WRITE(unsigned char store_write);
unsigned char I2C_READ(void);
void I2C_ACK();
void I2C_NOACK();
```

```
void I2S(unsigned intinteger,unsigned char digit_no);
//////////////////////////////////////////////////////
void UART_INIT();
void UART_TX(unsigned char char_tx);
void UART_STRINGTX(unsigned char *string_tx);
unsigned char UART_RX();
//////////////////////////////////////////////////////
void SERVOCALL(unsigned int GG);
sbit IN1=P1^0;
sbit IN2=P1^1;
sbit IN3=P1^2;
sbit IN4=P1^3;
sbit BUZZ=P0^7;
sbit SERVO=P0^6;
voidserialinturrpt();
unsigned char TEMP,TEMP2,i,SBUF1,m;
unsignedint KK;
void main(void)
{
ALCD_INIT();
UART_INIT();
BUZZ=0;
//////////servo///////////////
/*while(1)
{
for(m=70;m<=120;m++)
```

```
{
SERVOCALL(m);
}
while(1);
}*/
for(m=0;m<=100;m++)
{
SERVOCALL(94);
}
RI=0;
IE=0X90;
while(1)
  {
////////FORCE////////////////
TEMP=ADC_CALL(1);
I2S(TEMP,3);
ALCD_GOTO(1,1);
ALCD_STRING(STRING_out);
//         delay_ms(10);

if(TEMP<=60)
{
BUZZ=1;
}
else
{
BUZZ=0;
```

```
}
///////////////ACC////////////
TEMP2=ADC_CALL(2);
I2S(TEMP2,3);
ALCD_GOTO(5,1);
ALCD_STRING(STRING_out);
//delay_ms(10);

ADC_CALL(4);

/*      if(TEMP2==95)
{
SERVOCALL(105);
}
if(TEMP2==96)
{
SERVOCALL(105);
}
if(TEMP2==97)
{
SERVOCALL(84);
}
if(TEMP2==98)
{
SERVOCALL(110);
}
if(TEMP2==99)
```

```
{
SERVOCALL(111);
}
if(TEMP2==100)
{
SERVOCALL(112);
}
if(TEMP2==101)
{
SERVOCALL(113);
}*/
if(TEMP2==102)
{
SERVOCALL(114);
}
if(TEMP2==103)
{
SERVOCALL(115);
}
if(TEMP2==104)
{
SERVOCALL(116);
}
if(TEMP2==105)
{
SERVOCALL(117);
}
```

```
if(TEMP2==106)
{
SERVOCALL(118);
}
if(TEMP2==107)
{
SERVOCALL(118);
}
if(TEMP2>=108)
{
SERVOCALL(120);
}
////////////////////////////////
if((TEMP2==97))
{
SERVOCALL(90);
}
if((TEMP2==96))
{
SERVOCALL(85);
}
if(TEMP2==95)
{
SERVOCALL(80);
}
if(TEMP2==94)
{
```

```
SERVOCALL(75);
}
if(TEMP2==93)
{
SERVOCALL(70);
}
if(TEMP2==92)
{
SERVOCALL(65);
}
if(TEMP2==91)
{
SERVOCALL(60);
}
if(TEMP2==90)
{
SERVOCALL(55);
}
if(TEMP2==89)
{
SERVOCALL(50);
}
if(TEMP2==88)
{
SERVOCALL(50);
}
if(TEMP2==87)
```

```
{
SERVOCALL(50);
}
if(TEMP2==86)
{
SERVOCALL(50);
}
if(TEMP2==85)
{
SERVOCALL(50);
}
if(TEMP2<=84)
{
SERVOCALL(50);
}
//////////////MOTOR CONROL/////////
if(SBUF1=='F' || SBUF1=='f')
{
ALCD_GOTO(1,2);
ALCD_STRING("FORWARD   ");
IN1=1;
IN2=0;
IN3=1;
IN4=0;
UART_TX(0X0D);
UART_TX(0X0A);
UART_STRINGTX("FORWARD      ");
```

```
SBUF1=0;
}
if(SBUF1=='B' || SBUF1=='b')
{
ALCD_GOTO(1,2);
ALCD_STRING("BACKWARD  ");
IN1=0;
IN2=1;
IN3=0;
IN4=1;
UART_TX(0X0D);
UART_TX(0X0A);
UART_STRINGTX("BACKWARD     ");
SBUF1=0;
}
if(SBUF1=='R' || SBUF1=='r')
{
ALCD_GOTO(1,2);
ALCD_STRING("RIGHTWARD ");
IN1=1;
IN2=0;
IN3=0;
IN4=1;
UART_TX(0X0D);
UART_TX(0X0A);
UART_STRINGTX("RIGHTWARD     ");
```

```
SBUF1=0;
}
if(SBUF1=='L' || SBUF1=='l')
{
ALCD_GOTO(1,2);
ALCD_STRING("LEFTWARD  ");
IN1=0;
IN2=1;
IN3=1;
IN4=0;
UART_TX(0X0D);
UART_TX(0X0A);
UART_STRINGTX("LEFTWARD     ");
SBUF1=0;
}
if(SBUF1=='S' || SBUF1=='s')
{
ALCD_GOTO(1,2);
ALCD_STRING("STOP          ");
IN1=0;
IN2=0;
IN3=0;
IN4=0;
UART_TX(0X0D);
UART_TX(0X0A);
UART_STRINGTX("STOP         ");
```

```
SBUF1=0;
}
//////////////////////////////////////
}///while end///
}///main end////
////////////////////////////////////////////////////////
////////////////////////////////////////////////////////
voidserialinturrpt() interrupt 4
{
SBUF1=SBUF;
RI=0;
}
void SERVOCALL(unsigned int GG)
{
//for(i=0;i<=1;i++)
//
{
SERVO=1;
for(KK=0;KK<=GG;KK++);//ccw=365//cw=80//c=175
SERVO=0;
delay_ms(20); //
//
}
//
}
```

```
//////////////////////////////////////////////////////
//////////////////////////////////////////////////////
unsigned char ADC_CALL(unsigned char adc_channel)
{
unsigned char adc_value;
I2C_START();                    // Start Operation
I2C_WRITE(0x90);                // Send identifier I2C address 1001XXX0-Write
I2C_WRITE(0x40 | adc_channel);    // Send control byte to device (last 2 bits is the channel)
////////////////////
I2C_START();                    // Send I2C Start Transfer
I2C_WRITE(0x91);                // Send identifier I2C address - Read
adc_value=I2C_READ();                // Receiving DATA
I2C_NOACK();
I2C_STOP();
returnadc_value;
}
///////////////I2C DELAY//////////////////////////
void I2C_DELAY(unsigned int I2C_delayms)        //max. delay 65535
{
unsignedint loop_delay1, loop_delay2;
for(loop_delay1=0;loop_delay1<I2C_delayms;loop_delay1++)
{
for(loop_delay2=0;loop_delay2<115;loop_delay2++);
}
}
/////////////I2C CLOCK//////////////////////////
void I2C_CLOCK()
```

```
{
SCL = 1;                //Start CLOCK
I2C_DELAY(1);           //min 4us
SCL = 0;                //Clear SCL
I2C_DELAY(1);           //min 4us
}
//////////////I2C START//////////////////////////
void I2C_START()
{
SCL = 1;                //Set SCL
SDA = 1;        //Set SDA
I2C_DELAY(1);           //START condition set-up time 4.7us
SDA = 0;        //Clear SDA
I2C_DELAY(1);                   //START condition hold time min 4us
SCL = 0;        //Clear SCL
}
//////////////I2C STOP//////////////////////////
void I2C_STOP()
{
SCL=1;                          //Set SCL
SDA=0;                          //Clear SDA
I2C_DELAY(1);                   //STOP condition set-up time min 4us
SDA=1;
}
//////////////I2C WRITE TO SLAVE//////////////
void I2C_WRITE(unsigned char store_write)
{
```

```
unsigned char loop_write;
for(loop_write=0;loop_write<8;loop_write++)        //Bitwise Operation to WRITE
{
if(store_write& 0x80)                    //Filter MSB bit and send through SDA
SDA=1;
else
SDA=0;
store_write=store_write<< 1; //left shift by 1
I2C_CLOCK();                                  //1---0 clock to Latch Data
}
I2C_CLOCK();                                                  //9th pulse for Acknowledgement from SLAVE
}
//////////////I2C READ FROM SLAVE//////////////
unsigned char I2C_READ(void)
{
unsigned char loop_read, store_read;
//        SDA=1;
for(loop_read=0;loop_read<8;loop_read++) //Bitwise Operation to READ
{
store_read=store_read<<1;              //Left shift by 1
store_read=store_read | SDA; //Storing bits into a Byte DR
I2C_CLOCK();
}
returnstore_read;
}
//Acknowledgement is sent to SLAVE during READ operation after every 8bit Data received
```

```
void I2C_ACK()
{
SDA=0;                          //Clear SDA
I2C_CLOCK();
}
//NOAcknowledgement is sent when READ operation is end
void I2C_NOACK()
{
SDA=1;                          //Set SDA
I2C_CLOCK();
}
////////////////////////////////////////////////////////
////////////////INTEGER TO STRING CONVERSION//////////////////////
void I2S(unsigned intinteger,unsigned char digit_no)
{
unsignedintremainder,divisor;
unsigned char loop_i2s;
if(digit_no==1)
divisor=1;
if(digit_no==2)
divisor=10;
if(digit_no==3)
divisor=100;
if(digit_no==4)
divisor=1000;
if(digit_no==5)
divisor=10000;
```

```
remainder=integer;
for(loop_i2s=0;loop_i2s<digit_no;loop_i2s++)
 {
STRING_out[loop_i2s]=(remainder/divisor)+0x30;
remainder=integer%divisor;
divisor=divisor/10;
  }
STRING_out[loop_i2s]=0;
}
/////LCD//////////////////////
voidALCD_delay_ms(unsigned intdelayms)
{
unsignedint delay_count,delay_count1;
for(delay_count=0;delay_count<delayms;delay_count++)
{
for(delay_count1=0;delay_count1<115;delay_count1++)    ;
}
}
///////////////LCD ASCII DATA//////////////////
void ALCD_DATA(unsigned char data_to_lcd)
{
ALCD_delay_ms(2);
ALCD_PORT=((data_to_lcd& 0XF0)|0x0A);            //Higher 4 bit
EN=0;
ALCD_PORT=((data_to_lcd<<4)|0x0A);              //Lower 4 bit
EN=0;
}
```

```
//////////////////LCD COMMAND////////////////////
void ALCD_CMD(unsigned char data_to_lcd)
{
ALCD_delay_ms(2);
ALCD_PORT=((data_to_lcd& 0XF0)|0x08);                    //Higher 4 bit
EN=0;
ALCD_PORT=((data_to_lcd<<4)|0x08);                //Lower 4 bit
EN=0;
}////////////////// LCD INITIALIZATION//////////////
void ALCD_INIT()
{
ALCD_delay_ms(100);                                //settling time of LCD
ALCD_CMD(0X28);
ALCD_CMD(0X28);                                    //function set 4bit
ALCD_CMD(0X01);                                    //clear display
ALCD_CMD(0X06);                                    //character entry mode
ALCD_CMD(0X0F);                                    //display on and cursor blink
}
//////////////////CLEAR DDRAM OF LCD//////////////
void ALCD_CLEAR()
{
ALCD_CMD(0X01);
}
////////////////X-Y POSITION ACROSS LCD////////////
void ALCD_GOTO(unsigned char display_no,unsigned char line_no)        //POSITION
START FROM 1TO...
{
```

```
if(line_no==1)
ALCD_CMD(0X7F + display_no);
if(line_no==2)
ALCD_CMD(0XBF + display_no);
if(line_no==3)
ALCD_CMD(0X93 + display_no);
if(line_no==4)
ALCD_CMD(0XD3 + display_no);
}
////////////LOCAL STRING TO DISPLAY ON LCD//////////
void ALCD_STRING(unsigned char *local_string)
{
while(*local_string!=0)
{
ALCD_DATA(*local_string++);
}
}
////////////////////////////////////////////////////////////////////////////UART////////////
void UART_INIT()
{
SCON=0X50;
TMOD=0X20;
TH1=0XFD;
TR1=1;
}
unsigned char UART_RX()
{
```

```
while(RI==0);
RI=0;
return SBUF;
}
void UART_TX(unsigned char char_tx)
{
SBUF=char_tx;
while(TI==0);
TI=0;
}
void UART_STRINGTX(unsigned char *string_tx)
{
while(*string_tx!=0)
{
UART_TX(*string_tx);
string_tx++;
}
}
//////////////////delay//////////////////////////////////
voiddelay_ms(unsigned intdelayms)
{
unsignedint delay_count,delay_count1;
for(delay_count=0;delay_count<delayms;delay_count++)
{
for(delay_count1=0;delay_count1<115;delay_count1++)    ;
}
} //////////
```

CHAPTER 9

CONCLUSION

Advantages:-

1) Reduction Illegal trafficking of extra weight
2) Reduction in accident due to stabilized load

Application:

1) For transportation safety
2) I n load measurement

Future scope:

1) Magnetometer can used to make platform more stable

CHAPTER 10

REFERENCE

REFERENCE:-

Books:

- Microcontroller and embedded system by MohammadaliMazidi and Gillespie Mazidi.
- Embedded system by Raj kamal.

Websites:

- ✓ "www.google.com
- ✓ "8051-Basic microcontrollers "by Jan Axel son 1994
- ✓ Ultrasonic Basics (Banner Engineering)
- ✓ http://www.therailengineer.com

Journal Papers:

1. Moazzam, K. Kamal, S. Mathavan, S. Usman, and M. Rahman, "Metrology and visualization of potholes using the Microsoft Kinect sensor," in Proc. 16th Int. IEEE Conf. Intell. Transp. Syst., Oct. 2013, pp. 1284–1291

2. R. Sundar, S. Hebbar, and V. Golla, "Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection," IEEE Sensors J., vol. 15, no. 2, pp. 1109–1113, Feb. 2015