CHAPTER 1: INTRODUCTION

Introduction:

Electronic code door locking system or Password based door locking system. To the human race in this situation, security is the prime concern in which a man wants to secure all his things. Now-a-days, everyone is opting for the more secured home, secured accounts, secured society which is impossible these days. The Password protected system is the most éclatic in these days. But the electronic developments are making it possible to find a way to make everyone feel free.

In the previous days, a home is secured when it had mechanical locks which be can opened only with a single key. Later it had become the least used.Its because of the disadvantage it is having. The key can be duplicated easily whoever wishes to do so. This problem has been overcome with the electronic devices. In this, a password can be set, with which a home is said to be secured. The password cannot be duplicated or hacked or can be known by any other means. This is the major advantage of this system. A stranger can easily breakdown the lock and enter into our premises. But with the password protected system, even his greatest tool cannot break the password. This system is of porcelain which cannot be easily got down.

In this age of digital technology, every device and operation has become digital based. Now with digital based door lock systems, it is easier to control the door movement of the car or house. The new automated door lock system does not need a key to lock or unlock the door of the car or house. This digital door entry system is in fact controlled by a keyless.

The keyless door remote is an electronic circuit based device. This remote transmits radio Signals via air to another device of the system installed in the car door or house door. The remote device of most digital appliances can be operated with either single or double batteries.

The keyless car remote can be operated from a distance of up to 75 to 500 meter The working of a keyless remote is similar to the remote of a digital music player system like LCD television. However, some advanced digital remotes of a car lock system work on the entry number. A car owner has to press that entry number on the remote for controlling the movement of his car doors. The entry number is unique in nature and so is the remote device of an automatic keyless system. The remote of one keyless system cannot work on another.

The special feature of a keyless automatic door system is that it cannot be opened or closed manually. This digital system also carries an alarm signal which starts beeping when the digital door of a vehicle or home is touched by anyone. This system is thus essential for preventing

You can install a keyless entry system in your house as well as in your vehicle. Now a day, theft cases are increasing everywhere and the key based locking system is no longer a safe option. It is quite easy for thieves and robbers to open the traditional key based doors either with duplicate keys or some other way. Therefore for complete safety of your house and vehicle, you should install digital security systems at both places. With this system, you will never be in tension of.

In case your keyless remote stops functioning or you forget to remember the entry code of the door system, you can get repair services of the keyless digital remote from a car dealer or licensed locksmith. If you are on a journey and have lost your car remote, then too you can get a replacement of your lost remote from a local car dealer. The keyless entry remote systems are compact in size and there are numerous companies that manufacture these security gadgets. You can buy integrated electronic door lock systems either from a shop.

Modern keyless car lock appliances are equipped with added functions such as fuel indicator, temperature indicator of the car, etc. Therefore for security and added functionalities, you should install the latest version of a digital remote system in your car. However, this security system is also not a safe option anymore as thieves are now using high tech encrypted software's for unlocking the car doors. Cyberlock is an access control system that increases key control and accountability throughout your organization. Based on a unique design of electronic lock cylinders and intelligent keys, Cyberlock solves security problems that no other system can. Cyberlock cylinders replace standard mechanical lock cylinders**.** Cylinders install without wiring, easily converting your existing lock hardware into an access control system.

This concept of electronic cylinders means there are Cyberlock solutions for much more than just doorways**.** For customers large and small, Cyberlocks secure cabinets, cash boxes, trucks, gates, narcotic boxes, safes, vending machines, cell tower sites, traffic control boxes, server cabinets—and, yes, doors—around the world.

Cyberkeys are programmed by the software with the access permissions for each user. Keys work only during specific hours, and only for the locks each person is allowed access to. Because these intelligent keys restrict access and cannot be duplicated, the need to re-key a facility is eliminated.

Each time a key is used, a record of that event is stored in the key and the cylinder, for later transfer to the management software. For convenience, this information transfer can be done securely in a variety of ways, either at the computer where the software is being accessed or from remote locations. The data gathered is a valuable tool to better understand workplace efficiencies and areas of possible risk. And, there’s management software to meet the needs of every user**:** small-to medium-sized businesses to large campus settings and international corporations. In this digital world, everything is made to our convenience to restrict the restricted persons to access. Gradually the security systems are getting more and more digitized, so that everyone can have their things in person.

CHAPTER 2 : SCHEMATIC DESCRIPTION

2.1 BLOCK DIAGRAM:

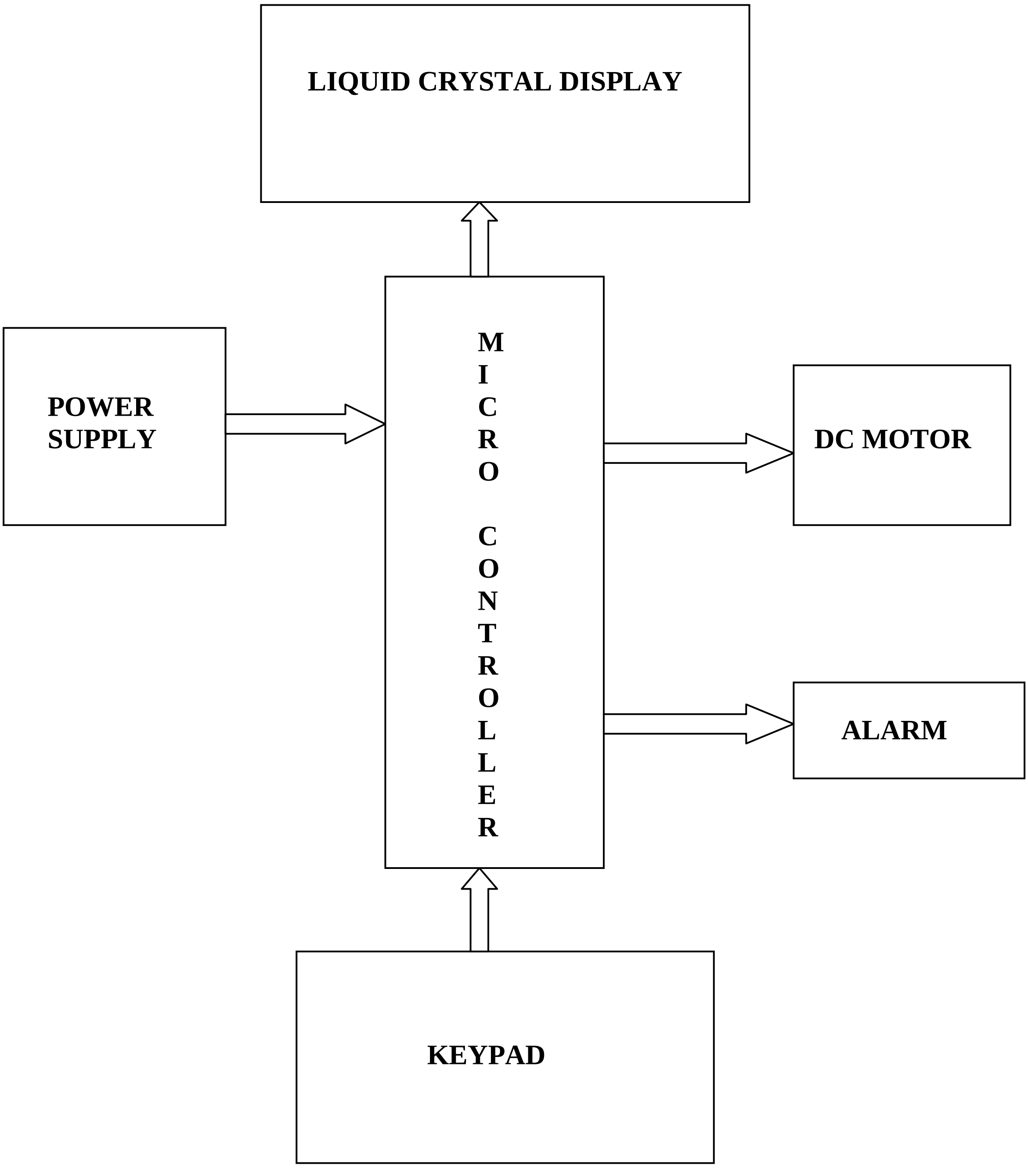


Fig. 2.1: Block Diagram

Description :

In this Password based door locking system, the input is given from the keypad. Before all the things to be considered, a power supply of 5V DC should be given to the controller and to the remaining components.The Microcontroller chosen is the IC P89C51 because of its flash memory of 8Kb and an extra timer.The password is given by the keypad which is a 4x4 matrix keypad. In this, all the displaying of the data, password, welcoming part is displayed in the Liquid Crystal Display(LCD) section.

A liquid crystal display (commonly abbreviated LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. It is often utilized in battery-powered electronic devices because it uses very small amounts of electric power.

The philips P89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 and 80C52 instruction set and pinout.

When the password given from the keypad is matched with the password stored in the memory, the DC motor gets on and rotates in the clockwise direction which opens the door. The Keypad has an interfacing IC MM74C922 which encodes the entered password. The DC Motor also has an interfacing IC L293DNE. When the password pressed in the keypad is not matched with the password in the program, the Alarm(buzzer) gets on for a few seconds and the red LED also gets on which indicates that the password entered is wrong. When password is wrong, it requests to reenter the password. When the entered password is correct, the Motor is on for 10 seconds and rotates in the anti-clockwise direction which closes the door.

2.2 CIRCUIT DIAGRAM:

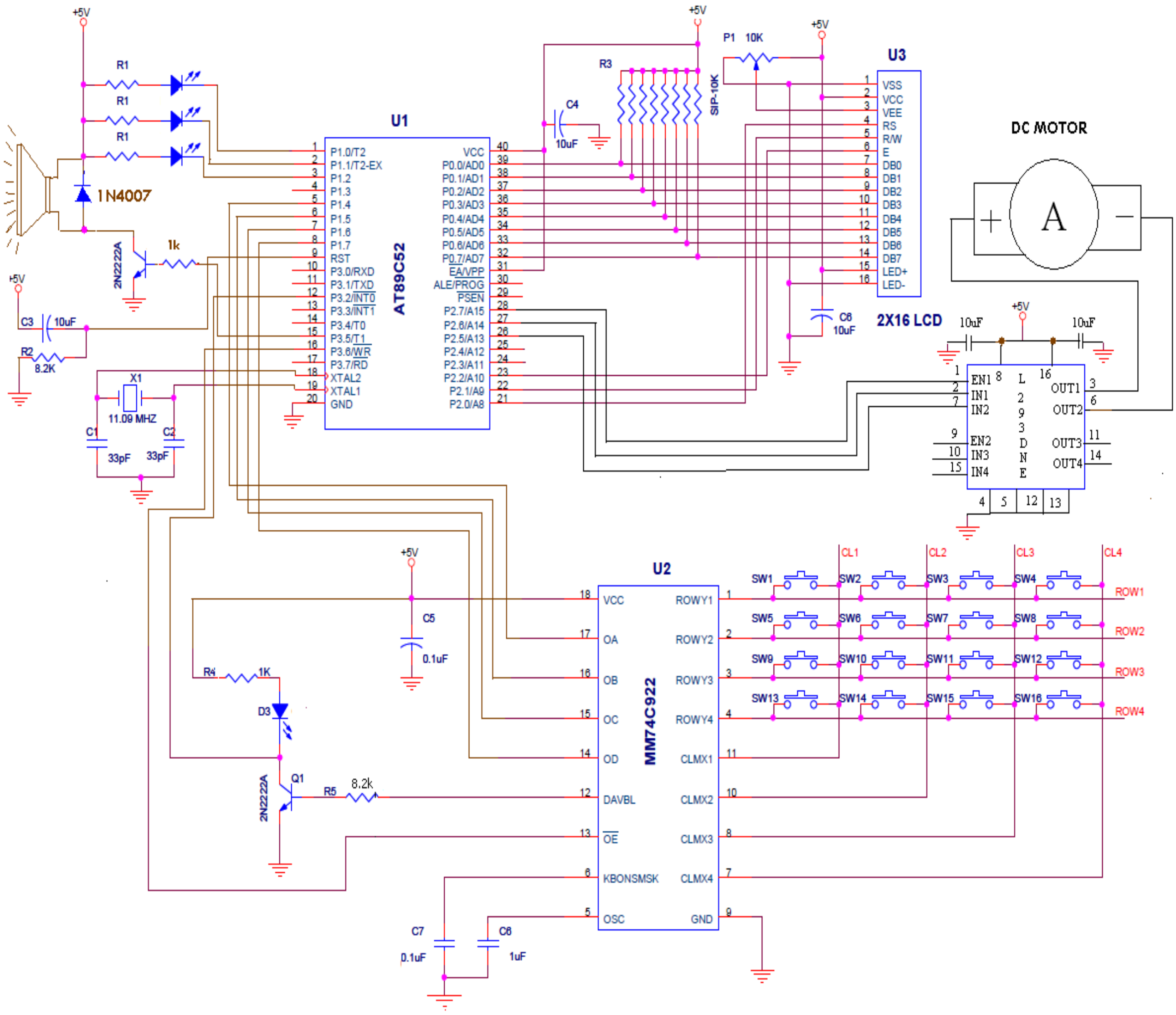


Fig. 2.2: Circuit Diagram

Description :

Password Based Door Locking System. In this, the major role is done by the Microcontroller. The entire circuit consists of a keypad through which the password is pressed, Liquid Crystal Display in which all the data and the entered password and all the things to be displayed are displayed, Microcontroller in which entered password is compared with the password which is already set in the program. The keypad can not be given to the microcontroller directly. So, an interfacing IC MM74C922 is placed before the keypad. The IC is an Encoder which encodes the entered password and redirect it to the controller.

The Microcontroller compares the entered password with the password which is already set in the program and gives the output according to it. When the entered password is wrong, the red LED flashes and the alarm(buzzer) gets ON. After a few seconds the buzzer gets OFF.

When the entered password is correct, the green LED glows and the door is opened. In the first, all the data is displayed and then there is an another option built up. The LOCK or UNLOCK is asked by the controller. When the 'lock' is pressed, It asks to enter the password and when the entered password matches with the password which is already in the program, the door is opened. Else, when the password is wrong, red led flashes and the bzzer gets ON.

In this, the door is opened with a DC motor. We can not directly connect the motor to the controller. So, an interfacing IC L293DNE is placed. The L293DNE is an Quadruple Half-H driver which is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V.In this circuit, we are using only a single motor. So, only few ports are connected and the remaining are not connected. Whenever a key is pressed, a blue LED glows which indicates the entered key is read. An orange LED is placed in the circuit which indicates the supply is running in the circuit. A Potentiometer is placed at the LCD which controls the contrast of the LCD.

CHAPTER 3: HARDWARE DESCRIPTION

MICROCONTROLLER (PHILIPS MICRO CONTROLLER):

Introduction:

The P89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable and erasable read only memory (EPROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 and 80C52 instruction set and pinout. 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 Flash on a monolithic chip, the PHILIPS P89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Features:

- Compatible with MCS-51 Products.
- 8K Bytes of In-System re-programmable flash memory.
- Endurance: 1,000 Write/Erase Cycles.
- Fully Static Operation: 0 Hz to 24 MHz.
- Three level Program Memory Lock.
- 256 x 8-bit Internal RAM.
- 32 Programmable I/O Lines.
- Three 16-bit Timer/Counters.
- Eight Interrupt Sources.
- Programmable Serial Channel.
- Low-power Idle and Power-down Modes.

PIN CONFIGURATION:

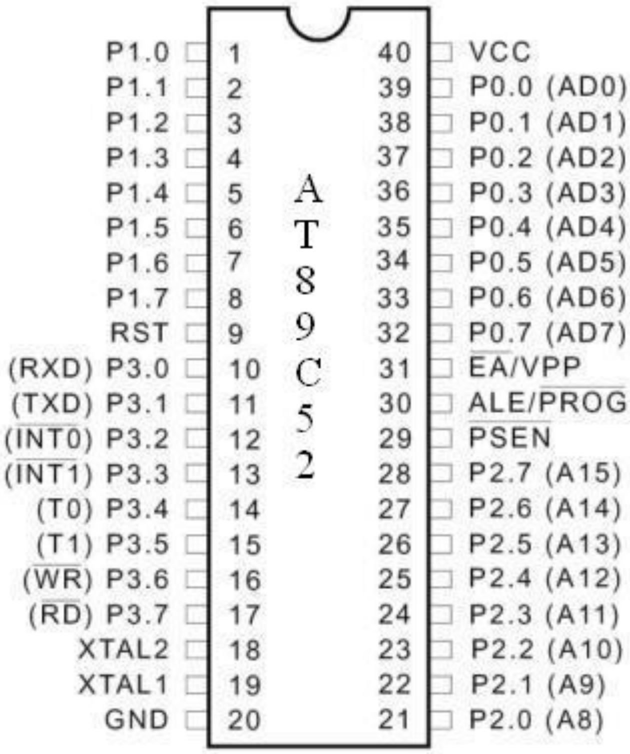


Fig3.1

BLOCK DIAGRAM OF P89C51 :

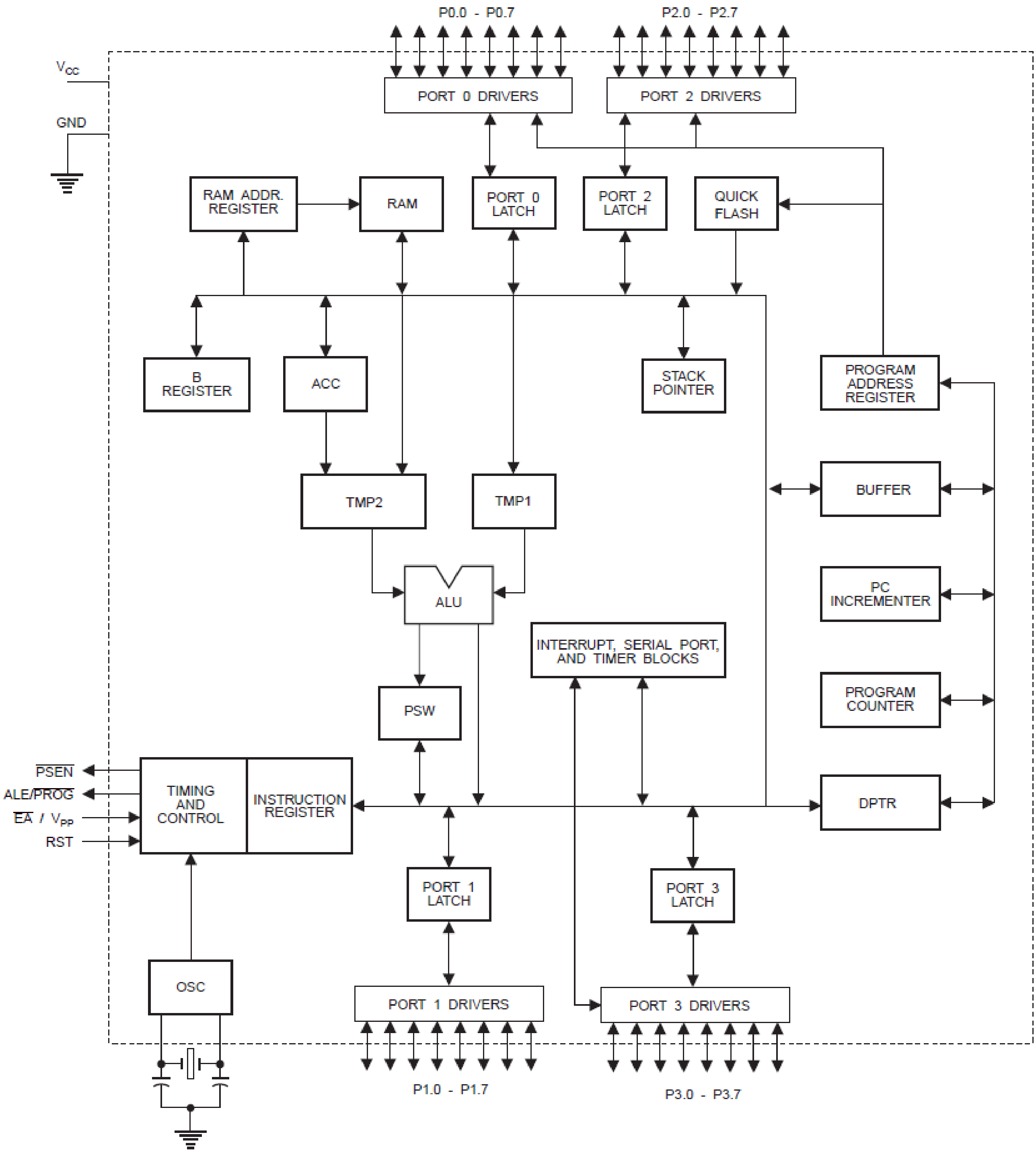


Fig3.2

Pin Description :

Vcc:

Supply voltage.

Gnd:

Ground.

Port 0:

Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high impedance inputs. Port 0 can also be configured to be the multiplexed low order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull ups are required during program verification.

Port 1:

Port 1 is an 8-bit bi-directional I/O port with internal pull-ups The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull ups. In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table. Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)

Port 2:

Port 2 is an 8-bit bi-directional I/O port with internal pull ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3:

Port 3 is an 8-bit bi-directional I/O port with internal pull ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull ups. Port 3 also serves the functions of various special features of the AT89C51, as shown in the following table. Port 3 also receives some control signals for Flash programming and verification.

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

Rst:

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

Ale/Prog:

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory. If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE- disable bit has no effect if the microcontroller is in external execution mode.

Psen:

Program Store Enable is the read strobe to external program memory. When the AT89C52 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

Ea/Vpp:

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming when 12-volt programming is selected.

XTAL1:

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2:

Output from the inverting oscillator amplifier.

Timer 0 and 1:

Timer 0 and Timer 1 in the AT89C52 operate the same way as Timer 0 and Timer 1 in the AT89C51.

Timer 2:

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Interrupts:

The AT89C52 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6. Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once. Note that Table shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that enerated the interrupt, and that bit will have to be cleared in software. The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Interrupt Enable(IE) Register :

(MSB)							(LSB)
EA	–	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.
User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.		

Idle Mode:

In idle mode, the CPU puts itself to sleep while all the onchip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset. Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to

two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external
memory.

Power-down Mode:

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before VCC is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

3.2 Liquid Crystal Display(LCD) :

- Reflective twisted nematic liquid crystal display.
- Vertical filter film to polarize the light as it enters.
- Glass substrate with ITO electrodes. The shapes of these electrodes will determine the dark shapes that will appear when the LCD is turned on or off. Vertical ridges etched on the surface are smooth.
- Twisted pneumatic liquid crystals.
- Glass substrate with common electrode film (ITO) with horizontal ridges to line up with the horizontal filter.
- Horizontal filter film to block/allow through light.

- Reflective surface to send light back to viewer. (In a backlit LCD, this layer is replaced with a light source.)

A **liquid crystal display** (commonly abbreviated **LCD**) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. It is often utilized in battery-powered electronic devices because it uses very small amounts of electric power.

Overview:

Each pixel of an LCD typically consists of a layer of molecules aligned between two transparent electrodes, and two polarizing filters, the axes of transmission of which are (in most of the cases) perpendicular to each other. With no liquid crystal between the polarizing filters, light passing through the first filter would be blocked by the second (crossed) polarizer.

The surface of the electrodes that are in contact with the liquid crystal material are treated so as to align the liquid crystal molecules in a particular direction. This treatment typically consists of a thin polymer layer that is unidirectionally rubbed using, for example, a cloth..

When a large number of pixels is required in a display, it is not feasible to drive each directly since then each pixel would require independent electrodes. Instead, the display is multiplexed. In a multiplexed display, electrodes on one side of the display are grouped and wired together (typically in columns), and each group gets its own voltage source. On the other side, the electrodes are also grouped (typically in rows), with each group getting a voltage sink. The groups are designed so each pixel has a unique, unshared combination of source and sink.

Specifications:

Important factors to consider when evaluating an LCD monitor:

- Resolution: The horizontal and vertical size expressed in pixels (e.g., 1024x768). Unlike CRT monitors, LCD monitors have a native-supported resolution for best display effect.
- Dot pitch: The distance between the centers of two adjacent pixels. The smaller the dot pitch size, the less granularity is present, resulting in a sharper image. Dot pitch may be the same both vertically and horizontally, or different (less common).
- Viewable size: The size of an LCD panel measured on the diagonal (more specifically known as active display area).
- Response time: The minimum time necessary to change a pixel's color or brightness. Response time is also divided into rise and fall time.
- Matrix type: Active or Passive.
- Viewing angle: (coll., more specifically known as viewing direction).
- Color support: How many types of colors are supported (coll., more specifically known as color gamut).
- Brightness: The amount of light emitted from the display (coll., more specifically known as luminance).
- Contrast ratio: The ratio of the intensity of the brightest bright to the darkest dark.
- Aspect ratio: The ratio of the width to the height (for example, 4:3, 16:9 or 16:10).
- Input ports (e.g., DVI, VGA, LVDS, or even S-Video and HDMI).

Color displays:

In color LCDs each individual pixel is divided into three cells, or sub pixels, which are colored red, green, and blue, respectively, by additional filters (pigment filters, dye filters and metal oxide filters). Each subpixel can be controlled independently to yield thousands or millions of possible colors for each pixel. Older CRT monitors employ a similar 'subpixel' structures via the use of phosphors, although the analog electron beam employed in CRTs do not hit exact 'subpixels'.

Color components may be arrayed in various pixel geometries, depending on the monitor's usage. If software knows which type of geometry is being used in a given LCD, this can be used to increase the apparent resolution of the monitor through subpixel rendering. This technique is especially useful for text anti-aliasing.

Drawbacks:

- Laptop LCD screen viewed at an extreme angle.
- LCD technology still has a few drawbacks in comparison to some other display technologies:

3.3 ENCODER-MM74C922:

Introduction:

The MM74C922 and MM74C923 CMOS key encoders provide all the necessary logic to fully encode an array of SPST switches. The keyboard scan can be implemented by either an external clock or external capacitor. These encoders also have on-chip pull-up devices which permit switches with up to 50 kΩ on resistance to be used. No diodes in the switch array are needed to eliminate ghost switches. The internal denounce circuit needs only a single external capacitor and can be defeated by omitting the capacitor. A Data Available output goes to a high level when a valid keyboard entry has been made. The Data Available output returns to a low level when the entered key is released, even if another key is depressed. The Data Available will return high to indicate acceptance of the new key after a normal debounce period; this two-key roll-over is provided between any two switches. An internal register remembers the last key pressed even after the key is released. The 3-STATE outputs provide for easy expansion and bus operation and are LPTTL compatible.

Features:

* 50 kΩ maximum switch on resistance
* On or off chip clock
* On-chip row pull-up devices
* Keybounce elimination with single capacitor
* Last key register at outputs

* 3-STATE output LPTTL compatible
* Wide supply range: 3V to 15V
* Low power consumption

Pin Configuration :

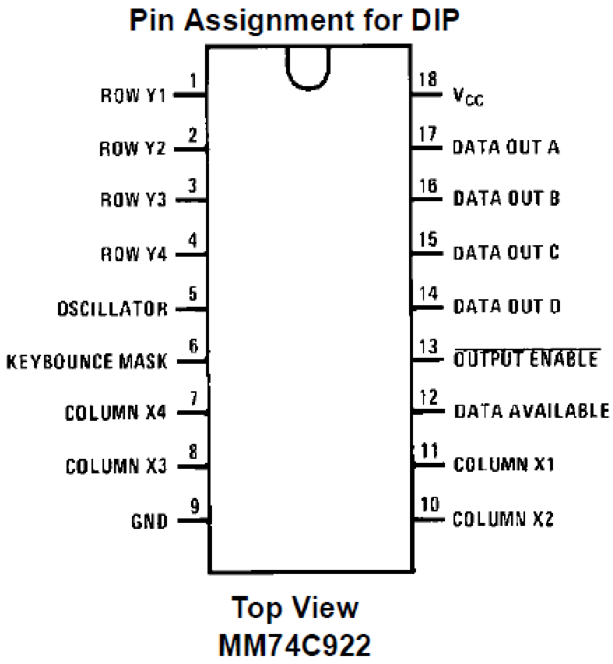


Fig3.3

Truth Tables

Switch Position	0 Y1, X1	1 Y1, X2	2 Y1, X3	3 Y1, X4	4 Y2, X1	5 Y2, X2	6 Y2, X3	7 Y2, X4	8 Y3, X1	9 Y3, X2	10 Y3, X3	11 Y3, X4
A	0	1	0	1	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1	0	0	1	1
C	0	0	0	0	1	1	1	1	0	0	0	0
D	0	0	0	0	0	0	0	0	1	1	1	1

Switch Position	12 Y4, X1	13 Y4, X2	14 Y4, X3	15 Y4, X4
A	0	1	0	1
B	0	0	1	1
C	1	1	1	1
D	1	1	1	1

Block Diagram

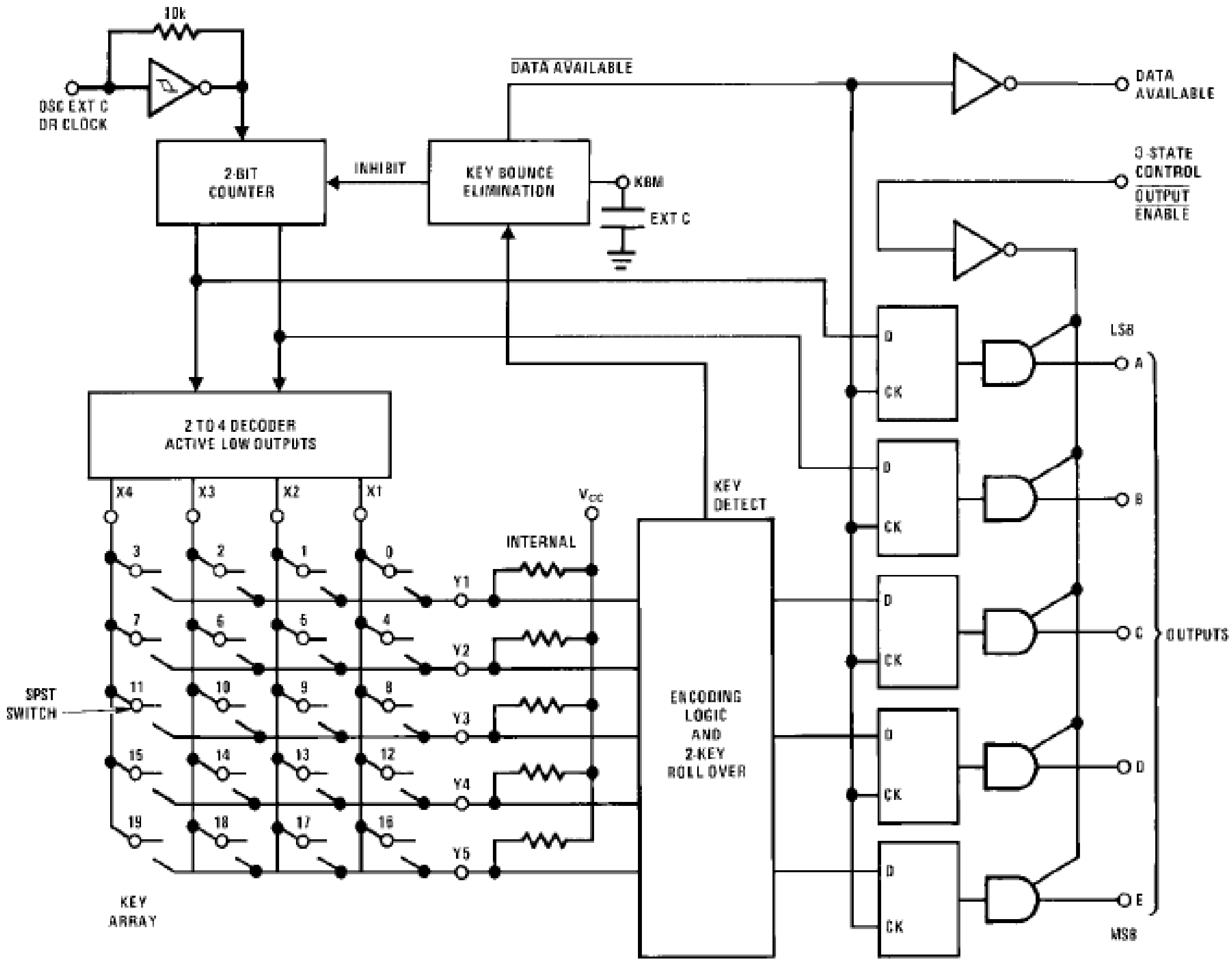


Fig3.4

Theory of Operation:

The MM74C922/MM74C923 Keyboard Encoders implement all the logic necessary to interface a 16 or 20 SPST key switch matrix to a digital system. The encoder will convert a key switch closer to a 4(MM74C922) or 5(MM74C923) bit nibble. The designer can control both the keyboard scan rate and the key debounce period by altering the oscillator capacitor, COSE, and the key bounce mask capacitor, CMSK. Thus, the MM74C922/MM74C923's performance can be optimized for many keyboards. The keyboard encoders connect to a switch matrix that is 4 rows by 4 columns (MM74C922) or 5 rows by 4 columns (MM74C923). When no keys are depressed, the row inputs are pulled high by internal pull-ups and the column outputs sequentially output a logic "0". These outputs are open drain and are therefore low for 25% of the time and otherwise off. The column scan rate is controlled by the oscillator input, which consists of a Schmitt trigger oscillator, a 2- bit counter, and a 2–4-bit decoder. When a key is depressed, key 0, for example, nothing will happen when the X1 input is off, since Y1 will remain high. When the X1 column is scanned, X1 goes low and Y1 will go low. This disables the counter and keeps X1 low. Y1 going low also initiates the key bounce circuit timing and locks out the other Y inputs.

The key code to be output is a combination of the frozen counter value and the decoded Y inputs. Once the key bounce circuit times out, the data is latched, and the Data Available (DAV) output goes high. If, during the key closure the switch bounces, Y1 input will go high again, restarting the scan and resetting the key bounce circuitry. The key may bounce several times, but as soon as the switch stays low for a debounce period, the closure is assumed valid and the data is latched. A key may also bounce when it is released. To ensure that the encoder does not recognize this bounce as another key closure, the debounce circuit must time out before another closure is recognized. The two-key roll-over feature can be illustrated by assuming a key is depressed, and then a second key is depressed. Since all scanning has stopped, and all other Y inputs are disabled, the second key is not recognized until the first key is lifted and the key bounce circuitry has reset. The output latches feed 3-STATE, which is enabled when the Output Enable (OE) input is taken low.

3.4 QUADRUPLE HALF-H DRIVER – L293DNE:

Features:

* 600-mA Output Current Capability per Driver.
* Pulsed Current 1.2 A per driver.
* Output clamp diodes for inductive transient suppression.
* Wide supply voltage range 4.5V to 36V.
* Separate input logic supply.
* Thermal shutdown.
* Internal ESD Protection.
* High Noise immunity inputs.
* Functional replacement for SGS L293D.

Pin Configuration:

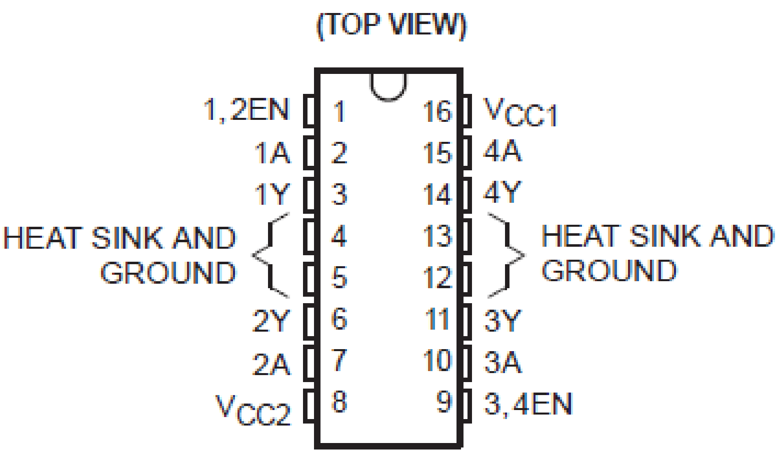


Fig3.5

Description:

The L293DNE is a quadruple high-current half-H driver designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. It is designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. All inputs are TTL-compatible. Each output is a complete totem-pole drive circuit with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs with drivers 1 and 2 enabled by 1,2EN and drivers 3 and enabled by 3,4 EN.When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs.

External high-speed output clamp diodes should be used for inductive transient suppression. When the enable input is low, those drivers are disabled, and their outputs are off and in a high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications. A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation. The L293DNE is designed for operation from 0°C to 70°C.

logic diagram

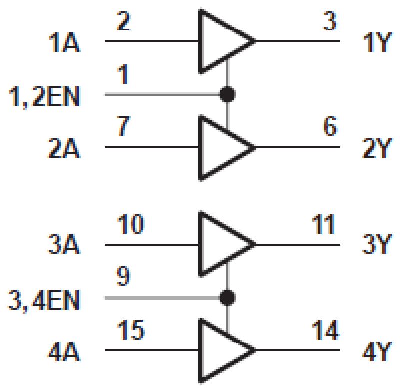


Fig3.6

FUNCTION TABLE
(each driver)

INPUTS‡		OUTPUT
A	EN	Y
H	H	H
L	H	L
X	L	Z

H = high-level, L = low level,
X = irrelevant, Z = high-impedance (off)
‡ In the thermal shutdown mode, the output is in the high-impedance state regardless of the input levels.

Block Diagram :

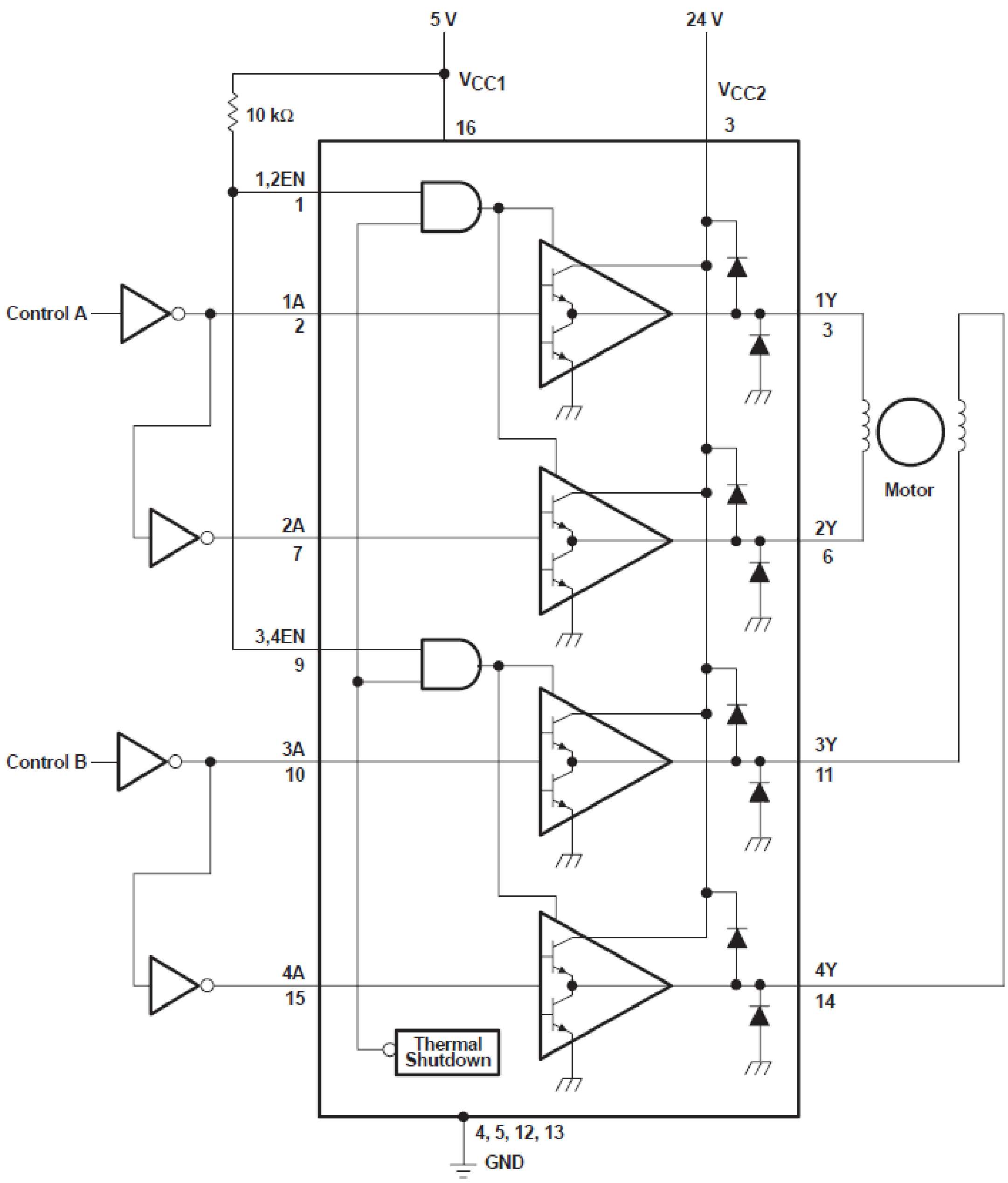
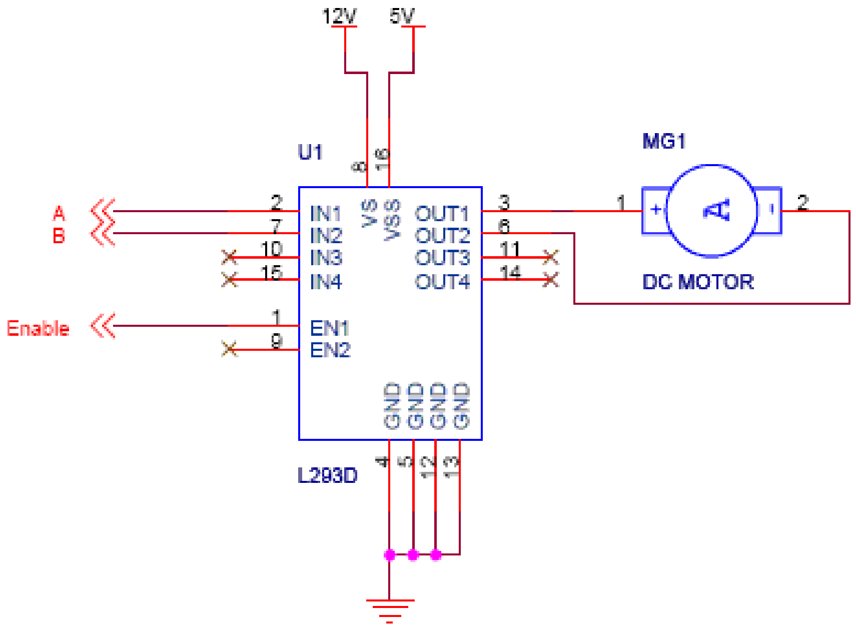


Fig3.7 :Two-phase motor driver

A simple schematic for interfacing a DC motor using L293DNE is shown below :



Truth Table

A	B	Description
0	0	Motor stops or Breaks
0	1	Motor Runs Anti-Cloclwise
1	0	Motor Runs Clockwise
1	1	Motor Stops or Breaks

For above truth table, the Enable has to be Set (1). Motor Power is mentioned 12V, but you can connect power according to your motors.

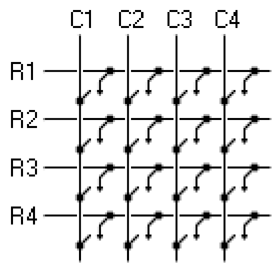
3.5 4 x 4 KEYPAD :

Introduction:

Keypads are a part of HMI or Human Machine Interface and play really important role in a small embedded system where human interaction or human input is needed. Martix keypads are well known for their simple architecture and ease of interfacing with any microcontroller.

Construction of matrix keypad:

Construction of a keypad is really simple. As per the outline shown in the figure below we have four rows and four columns. In between each overlapping row and column line there is a key.



So keeping this outline we can construct a keypad using simple SPST Switches as shown below:

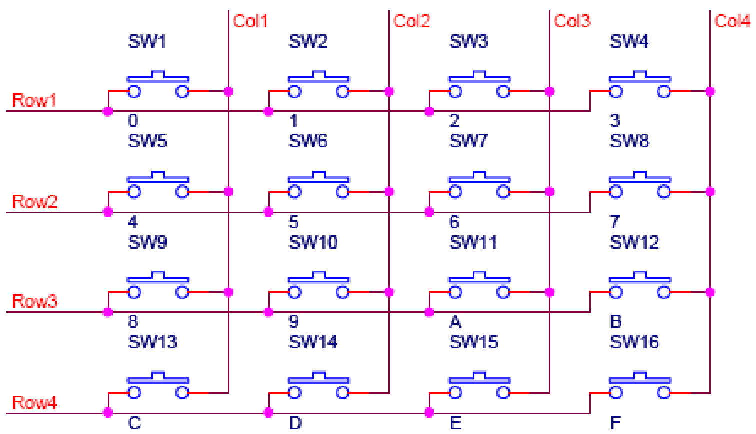


Fig3.8

Now our keypad is ready, all we have to do is connect the rows and columns to a port of microcontroller and program the controller to read the input.

.

3.6 DC MOTOR:

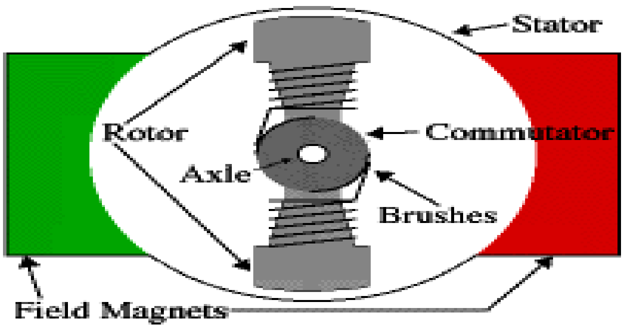
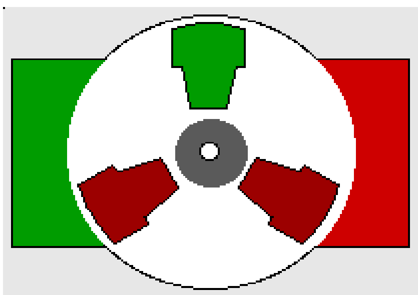


Fig3.9

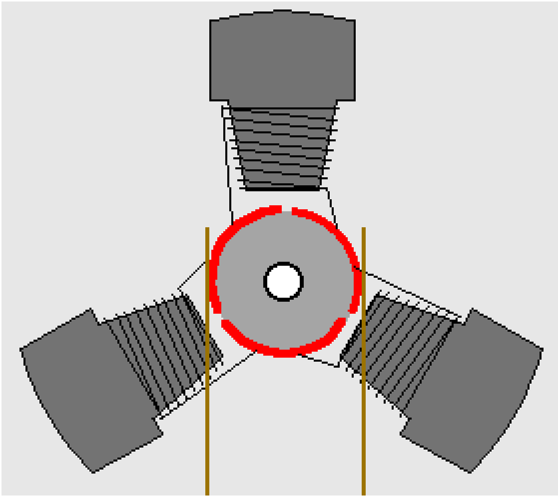
Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that BEAMers will see), the external magnetic field is produced by high-strength permanent magnets[1]. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces.The rotor (together with the axle and attached commutator) rotate with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

The geometry of the brushes, commutator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commutator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.

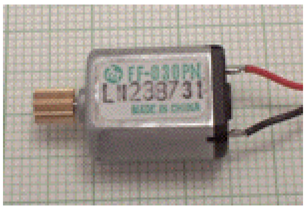


In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can (perfectlyaligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

So since most small DC motors are of a three-pole design, let's tinker with the workings of



You'll notice a few things from this -- namely, one pole is fully energized at a time (but two others are "partially" energized). As each brush transitions from one commutator contact to the next, one coil's field will rapidly collapse, as the next coil's field will rapidly charge up (this occurs within a few microsecond). We'll see more about the effects of this later, but in the meantime you can see that this is a direct result of the coil windings' series wiring:



There's probably no better way to see how an average DC motor is put together, than by just opening one up. Unfortunately this is tedious work, as well as requiring the destruction of a perfectly good motor.

The use of an iron core armature (as in the Mabuchi, above) is quite common, and has a number of advantages[2]. First off, the iron core provides a strong, rigid support for the windings -- a particularly important consideration for high-torque motors. The core also conducts heat away from the rotor windings, allowing the motor to be driven harder than might otherwise be the case. Iron core construction is also relatively inexpensive compared with other construction types.

But iron core construction also has several disadvantages. The iron armature has a relatively high inertia which limits motor acceleration. This construction also results in high winding inductances which limit brush and commutator life.

In small motors, an alternative design is often used which features a 'coreless' armature winding. This design depends upon the coil wire itself for structural integrity. As a result, the armature is hollow, and the permanent magnet can be mounted **inside** the rotor coil. Coreless DC motors have much lower armature inductance than iron-core motors of comparable size, extending brush and commutator life.

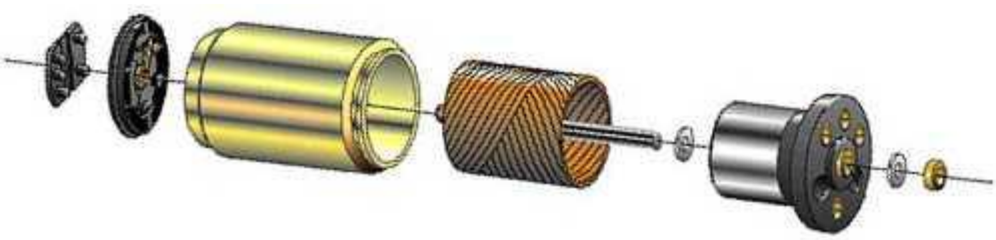


fig3.10

The coreless design also allows manufacturers to build smaller motors; meanwhile, due to the lack of iron in their rotors, coreless motors are somewhat prone to overheating. As a result, this design is generally used just in small, low-power motors. BEAMers will most often see coreless DC motors in the form of pager motors.

3.7 BUZZER:

These are mechanical devices which produce sound via magnetized arm repeatedly striking a diaphram. These devices operate with a DC voltage and current reqirement is small, generally in the region of 10mA. Buzzers generate a buzzing noise(single tone) in the frequency range of 300- 500 Hz. Buzzers are small devices, they can be either panel mounted or PCB mounted.

Circuit Symbol:

Fig3.11

Specifications:

Rated Voltage: 5VDC

Operating voltage : 4~7VDC

Output S.P.L: ≥85dB

Rated current: ≤30mA

Generated Freq.: 2300±500Hz

Operating Temp.: -20~+70 ℃

Storage Temp.: -30~+85 ℃

Weight: 2g

P max : 0.5W.

Working principle:

The working principle of passive electromagnetic **buzzer**passive electromagnetic **buzzer**: Ac signal through the bypass line in the stent in the stent core package produce a column of alternating magnetic flux, the alternating magnetic flux and magnetic flux for a constant stack, so that films of molybdenum to a given The exchange of signals with the frequency of vibration and sound resonator. Products of the frequency response and sound pressure curve and the value gap, molybdenum films inherent vibration frequency (which can be rough refraction.For small film thickness of molybdenum), Shell (Tune Helmholtz resonance) frequency of the magnetometer magnetic wire is directly related to the diameter. Composed of electromagnetic buzzer by electromagnetic oscillator, the electromagnetic coil, magnet, vibration, such as the composition of membrane and shell.Access to power, the audio signal generated by oscillator current through the electromagnetic coil to generate magnetic fields of electromagnetic coils.Membrane vibration in the electromagnetic coil and magnet interaction, the buzzer sound vibration periodically.

Choosing a suitable NPN transistor

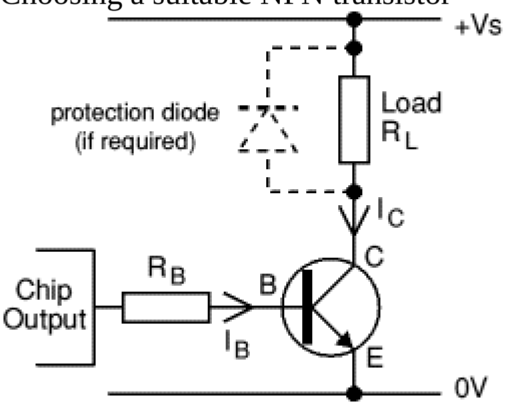


Fig3.12

(load is on when IC output is high)

The circuit diagram shows how to connect an **NPN transistor**, this will switch on the load when the IC output is **high**. If you need the opposite action, with the load switched on when the IC output is **low** (0V) please see the circuit for a PNP transistor below.

The procedure below explains how to choose a suitable switching transistor.

- The transistor's maximum collector current Ic(max) must be greater than the load current Ic.

$$\text{load current } I_c = \frac{\text{supply voltage } V_s}{\text{load resistance } R_L}$$

- The transistor's minimum current gain $h_{FE}(min)$ must be at least **five** times the load current Ic divided by the maximum output current from the IC.

$$h_{FE}(min) > 5 \times \frac{\text{load current } I_c}{\text{max. IC current}}$$

Choose a transistor which meets these requirements and make a note of its properties:
There is a table showing technical data for some popular transistors on the transistors page.

- Calculate an approximate value for the base resistor:

$$R_B = \frac{V_c \times h_{FE}}{5 \times I_c}$$ where V_c = IC supply voltage (in a simple circuit with one supply this is Vs)

- For a simple circuit where the IC and the load share the same power supply ($V_c = V_s$) you may prefer to use: $R_B = 0.2 \times R_L \times h_{FE}$
- Then choose the nearest standard value for the base resistor.
- Finally, remember that if the load is a motor or relay coil a protection diode is required.

3.8 PROTECTION DIODE:

If the load is a **motor**, relay or **solenoid** (or any other device with a coil) a diode must be connected across the load to protect the transistor from the brief high voltage produced when the load is switched off. The diagram shows how a protection diode is connected 'backwards' across the load, in this case a relay coil.

Current flowing through a coil creates a magnetic field which collapses suddenly when the current is switched off. The sudden collapse of the magnetic field induces a brief high voltage across the coil which is very likely to damage transistors and ICs. The protection diode allows the induced voltage to drive a brief current through the coil (and diode) so the magnetic field dies away quickly rather than instantly. This prevents the induced voltage becoming high enough to cause damage to transistors and ICs.

A resistor R_B is required to limit the current flowing into the base of the transistor and prevent it being damaged. However, R_B must be sufficiently low to ensure that the transistor is thoroughly saturated to prevent it overheating

CHAPTER 4: SOFTWARE

4.1 INTELLIGENT UNIVERSAL PROGRAMMER:

Fig4.1

LabTool-48UXP USB:

Introduction:

The LabTool-48UXP USB is a universal device programmer, which works through your PC's parallel port. It features a 48-pin universal pin driver and an expandable TTL pin driver. It supports over 5600 different devices (including memory, logic and single chip) that's over 11000 devices including different package forms in current software release, new chips are added to the support lists every few months.

Universal pin driver--True universal programming:

The LabTool-48UXP features universal pin driver, each pin can supply four different voltage, ground, it also can be configurable as TTL high/low levels with pull-high/pull-low, high-speed clock and high impedance. This advanced pin design lets you program any DIL device of up to 48 pins without needing an adapter.

Unbeatable programming:

The LabTool-48UXP's on-board intelligence reduces system overhead to a minimum. The LabTool-48UXP has 100% more performance then its predecessor product in program the high density flash chip, it can program a Intel 32 M bit flash chip in less then 60 seconds. Following is the programming speed example of labtool-48XP

	Intel 28F320C3B	AMD 29DL323DB
Blank check	18.6 sec	18.9 sec
Program	57.5 sec	76.2 sec
Verify	32.5 sec	33.0 sec
Total	108.6 sec	128.1 sec

The LabTool-48XP is much faster than its competitors, making it much more productive with today's high density, multi-megabit memory devices.

The LabTool-48UXP performs device insertion and contact checks before it programs each device. It can detect poor pin contact and devices inserted upside down or in the wrong position. This function protects your pocketbook by preventing expensive chip damage due to operator error.

Many EPROM and Flash memories have a build-in device ID and manufacturer ID. The LabTool-48UXP can read the device's ID by press hot key to detect the ID and compare its database and determined the chip correct vendor and product number. This feature is especially useful with secondhand chips and devices that have had their part number accidentally (or intentionally) removed (this function only applied to 28 pin or 32 pin EPROM and Flash).

To meet mass-production requirements the LabTool-48UXP has implemented new patented technology in both its hardware and software. After entering the Mass-production Mode, the production line operator inserts a device into the ZIF socket. An LED on the LabTool-48UXP indicates the device is ready and the operator simply removes it and replaces it with a new one.

No formal training is necessary adding flexibility and saving time and money. In addition, the LabTool-48UXP's auto-sensing feature ensures the device has been inserted correctly and then automatically programs the device. Furthermore, in the mass-production mode the system keyboard is automatically disabled preventing the operator from making any inadvertent mistakes.

If your memory devices need individual serial numbers, the LabTool-48 has an Auto Increment function,This simply increments the serial numbers in the buffer each time a new device is inserted. This saves time and money. The LabTool-48 has everything you need in one package: 48-pin driver, socket and complete device support library. You don't need to order separate device libraries.

The LabTool-48XP lets you select the verify voltage after you have programmed the device, e.g., Vcc, Vcc ?%,Vcc ?0%.The Vcc voltage can be 2.0V to 6.5V. This feature ensures that your device has been programmed properly, preventing failures due to programming errors and ensuring data retention.

The programming software's easy-to-use graphical interface eliminates your learning curve--you don't need the manual! Like MS-Windows, the software offers pop-up dialog boxes and on-line help. You can simply type the device vendor's name and part number. The software will select the proper driver automatically; you don't need to search one by one through long lists of devices.

If your memory devices need individual serial numbers with different increment sequence and initial value, the LabTool-48XP has an Auto Increment function. This simply increments the serial numbers in the buffer each time a new device is inserted. This saves time and money.

User can create and save a project file which contains device selection, buffer data and all the programming set-up options, this project file can be called upon at any time for future use without having to go through the setting up procedure again, your design file can easily pass to production department without making any mistake.

The LabTool-48UXP has additional safety features such as a built-in current limit and pin continuity check function. This prevents damage from faulty chips during the programming cycle. As a result we have provided a 3 year warranty on parts and labor for the Lab.

Universal 44-pin PLCC, QFP, PSOP and TQFP adapter supports 44-pin chips:

The universal pin driver on the LabTool-48UXP enables you to support all 44-pin chips such as EPLD, EPROM, Flash, EPLD and Micro-controllers. You need only one 44-pin universal adapter which eliminates the need to buy multiple adapters and saves money. Moreover, different package devices within 48 pins will only require one adapter.

The LabTool-48UXP support 2.0V ~ 5.0V logic level input/output, and it can supply 2.0V~21V analog voltage (such as VCC). Support lower then 2.7V low voltage chip such as 1.8V also possible by special low voltage adapter on top of its 48 pin ZIF socket

4.2 SOURCE CODE :

PROGAMMING :

```
#include <reg51.h>
sbit  r0=P1^0;  //keypad rows
sbit  r1=P1^1;
sbit  r2=P1^2;
sbit  r3=P1^3;

sbit  c0=P1^4;  //keypad columns
sbit  c1=P1^5;
sbit  c2=P1^6;
sbit  c3=P1^7;


sfr ldata=0x80;   // PORT0 for SFR address(LCD data pins)

sbit rs=P2^5;             //RS set for Giving the Command or Data
sbit rw=P2^6;               // Read/write command for the data in the  LCD RAM
sbit en=P2^7;           // Enable Bit of the LCD
sbit busy=P0^7;         //busy pin of  the controller

sbit Motor_In1=P2^4;
sbit Motor_In2=P2^3;
sbit en1=P2^2;
void lcdcmd (unsigned char);
void lcddata (unsigned char);
void lcdready(void);
void MSDelay(unsigned int);
void WriteString(unsigned char,unsigned char *);
unsigned char KeyTest(void);
void Start(void);
void LCDClear(void);
void MainMenu(void);
void UpdateService(void);
unsigned char MY_PWD[4]={'1','2','3','4'};

void lcdcmd (unsigned char value)
 {
  lcdready();
  ldata=value;
  rs=0;
  rw=0;
  en=1;
  MSDelay(1);
  en=0;
```

```
}
void lcddata (unsigned char value)
 {
  lcdready();
  ldata=value;
  rs=1;
  rw=0;
  en=1;
  MSDelay(1);
  en=0;
 }
void lcdready(void)
 {
  busy=1;
  rs=0;
  rw=1;
  while(busy==1)
   {
        en=0;
        MSDelay(1);
        en=1;
       }
 }
void MSDelay(unsigned int Iter)
 {
  unsigned int i, j;
  for(i=0;i<Iter;i++)
   {
        for(j=0;j<1275;j++);
       }
 }
void WriteString(unsigned char count,unsigned char *MSG)
{
 unsigned char i;
 for(i=0;i<count;i++)
 {
   lcddata(MSG[i]);
 }
}
 unsigned char KeyTest(void)
{
P1=0xF0;
while (1)
{
while(P1!=0xF0)
 {
 r0=0; r1=1;r2=1;r3=1;
 if(c0==0)
 return '1';
```

```
else if(c1==0)
  return '2';
 else if(c2==0)
  return '3';
 else if(c3==0)
  return 'A';

 r1=0; r0=1;r2=1;r3=1;
 if(c0==0)
  return '4';
 else if(c1==0)
  return '5';
 else if(c2==0)
  return '6';
 else if(c3==0)
  return 'B';

 r2=0; r0=1;r1=1;r3=1;
 if(c0==0)
  return '7';
 else if(c1==0)
  return '8';
 else if(c2==0)
  return '9';
 else if(c3==0)
  return 'C';   // Down Arrow

 r3=0; r0=1;r1=1;r2=1;
 if(c0==0)
  return 'E';
 else if(c1==0)
  return '0';
 else if(c2==0)
  return 'F';   // Redail
 else if(c3==0)
  return 'D';   // Enter
  }
  }
}
void MainMenu(void)
{
 unsigned char key;
 // Start our project
  key=0;
 LCDClear();
 WriteString(7,"1. open");
 lcdcmd(0xC0);

WriteString(7,"2. close");
```

```
do
  {
   key=KeyTest();
  }while((key!='1')&&(key!='2'));



switch(key)
  {
    case '1':
                                        Motor_In1=0;
                                        MSDelay(100);
                                        Motor_In2=1;
                                        LCDClear();
                                        WriteString(12,"Door is open");
                                        MSDelay(500);
                                        Motor_In1=1;
                                        Motor_In2=1;
                                        LCDClear();
                                        WriteString(8,"Press 1:");
                                         do
                                        {
                                  key=KeyTest();
                                }while((key!='1')&&(key!='2'));
                                MainMenu();
                      break;
        case '2':

                                        Motor_In1=1;
                                        MSDelay(100);
                                        Motor_In2=0;

                                        LCDClear();
                                        WriteString(13,"Dorr is close");
                                        MSDelay(100);
                                        Motor_In1=1;
                                        Motor_In2=1;
                                        LCDClear();
                                       WriteString(8,"Press 1:");
                                          {
                                  key=KeyTest();
                                }while(key!='1');
                                MainMenu();

                        break;
  }
}
```

```
void Start(void)
 {
  unsigned char key;
  unsigned char uid[4],loop,pwd[4];

  key=0x0;
// wait for enter.

do
         {
           key=KeyTest();
         }while(key!='E');
        LCDClear();
    WriteString(8,"User ID:");
        loop=0;
        do
         {
          key=KeyTest();
          if(key>='0' && key<='9')
           {
                uid[loop]=key;
                lcddata(key);
                loop++;
                MSDelay(100);
         //LCDClear();
         //WriteString(13,"Its My World!");
              }
         }while(loop!=4);
        lcdcmd(0xC0);
    WriteString(9,"Password:");
        loop=0;
        do
         {
          key=KeyTest();
          if(key>='0' && key<='9')
           {
                pwd[loop]=key;
                //lcddata(pwd[loop]);
                lcddata('*');
                MSDelay(100);
                loop++;
              }
         }while(key!='E');
    LCDClear();
        if((pwd[0]=MY_PWD[0])&&
           (pwd[1]=MY_PWD[1])&&
               (pwd[2]=MY_PWD[2])&&
              (pwd[3]=MY_PWD[3]))
 {
          WriteString(15,"Authenticated!!");
```

```
key=0;
       do
        {
          key=KeyTest();
        }while(key!='E')
MainMenu();

}
       else
        {
         WriteString(16,"Access Denied!!!");
         Start();
        }
}

void LCDClear(void)
{
  lcdcmd(0x0E);
  lcdcmd(0x01);
  lcdcmd(0x06);
  lcdcmd(0x80);
}
void main(void)
 {
  en1=1;
   Motor_In1=1;
   Motor_In2=1;
  lcdcmd(0x38);
  lcdcmd(0x0E);
  lcdcmd(0x01);
  lcdcmd(0x06);
  lcdcmd(0x80);
  WriteString(16,"Welcome ECE@KGRCET");
  lcdcmd(0xC0);
  WriteString(16,"Home security system Proj");
  Start();
 }
```

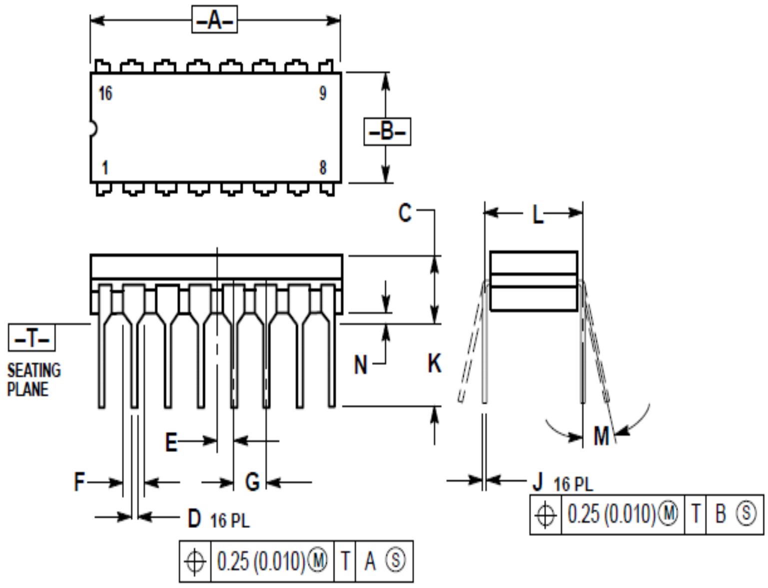
APPENDIX

ELECTRICAL CHARACTERISTICS (Voltages Referenced to V_{SS})

Characteristic	Symbol	V_{DD} Vdc	– 55°C Min	– 55°C Max	+ 25°C Min	+ 25°C Typ[1]	+ 25°C Max	+ 125°C Min	+ 125°C Max	Unit
Output Voltage "0" Level $V_{in} = V_{DD}$	V_{OL}	5.0 10 15	— — —	0.05 0.05 0.05	— — —	0 0 0	0.05 0.05 0.05	— — —	0.05 0.05 0.05	Vdc
"1" Level $V_{in} = 0$	V_{OH}	5.0 10 15	4.95 9.95 14.95	— — —	4.95 9.95 14.95	5.0 10 15	— — —	4.95 9.95 14.95	— — —	Vdc
Input Voltage "0" Level (V_O = 4.5 Vdc) (V_O = 9.0 Vdc) (V_O = 13.5 Vdc)	V_{IL}	5.0 10 15	— — —	1.5 3.0 4.0	— — —	2.25 4.50 6.75	1.5 3.0 4.0	— — —	1.5 3.0 4.0	Vdc
"1" Level (V_O = 0.5 Vdc) (V_O = 1.0 Vdc) (V_O = 1.5 Vdc)	V_{IH}	5.0 10 15	3.5 7.0 11	— — —	3.5 7.0 11	2.75 5.50 8.25	— — —	3.5 7.0 11	— — —	Vdc
Output Drive Current (V_{OH} = 2.5 Vdc) Source (V_{OH} = 9.5 Vdc) (V_{OH} = 13.5 Vdc)	I_{OH}	5.0 10 15	– 1.6 – 1.6 – 4.7	— — —	– 1.25 – 1.30 – 3.75	– 2.5 – 2.6 – 10	— — —	– 1.0 – 1.0 – 3.0	— — —	mAdc
(V_{OL} = 0.4 Vdc) Sink (V_{OL} = 0.5 Vdc) (V_{OL} = 1.5 Vdc)	I_{OL}	5.0 10 15	3.75 10 30	— — —	3.2 8.0 24	6.0 16 40	— —	2.6 6.6 19	— — —	mAdc
Input Current	I_{in}	15	—	± 0.1	—	±0.00001	± 0.1	—	± 1.0	µAdc
Input Capacitance (V_{in} = 0)	C_{in}	—	—	—	—	10	20	—	—	pF
Quiescent Current (Per Package)	I_{DD}	5.0 10 15	— — —	1.0 2.0 4.0	— — —	0.002 0.004 0.006	1.0 2.0 4.0	— — —	30 60 120	µAdc
Total Supply Current[2,3] (Dynamic plus Quiescent, per package) (C_L = 50 pF on all outputs, all buffers switching	I_T	5.0 10 15	I_T = (1.8 µA/kHz) f + I_{DD} I_T = (3.5 µA/kHz) f + I_{DD} I_T = (5.3 µA/kHz) f + I_{DD}							µAdc

OUTLINE DIMENSIONS

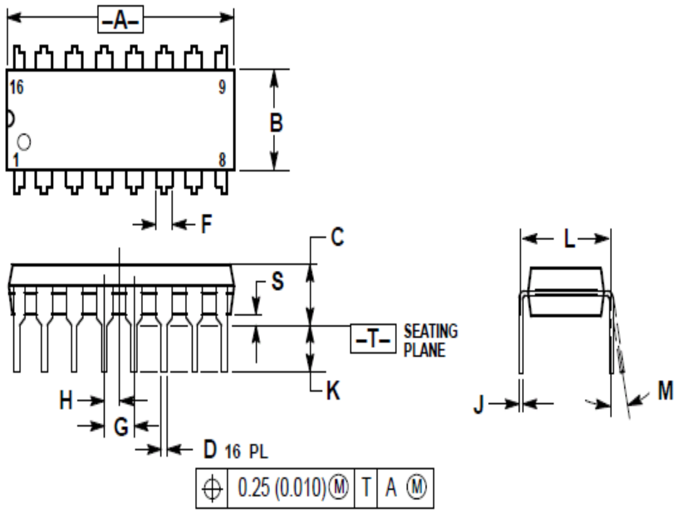
L SUFFIX
CERAMIC DIP PACKAGE
CASE 620–10
ISSUE V



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSION F MAY NARROW TO 0.76 (0.030) WHERE THE LEAD ENTERS THE CERAMIC BODY.

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	0.750	0.785	19.05	19.93
B	0.240	0.295	6.10	7.49
C	—	0.200	—	5.08
D	0.015	0.020	0.39	0.50
E	0.050 BSC		1.27 BSC	
F	0.055	0.065	1.40	1.65
G	0.100 BSC		2.54 BSC	
H	0.008	0.015	0.21	0.38
K	0.125	0.170	3.18	4.31
L	0.300 BSC		7.62 BSC	
M	0°	15°	0°	15°
N	0.020	0.040	0.51	1.01

P SUFFIX
PLASTIC DIP PACKAGE
CASE 648–08
ISSUE R



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
4. DIMENSION B DOES NOT INCLUDE MOLD FLASH.
5. ROUNDED CORNERS OPTIONAL.

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	0.740	0.770	18.80	19.55
B	0.250	0.270	6.35	6.85
C	0.145	0.175	3.69	4.44
D	0.015	0.021	0.39	0.53
F	0.040	0.70	1.02	1.77
G	0.100 BSC		2.54 BSC	
H	0.050 BSC		1.27 BSC	
J	0.008	0.015	0.21	0.38
K	0.110	0.130	2.80	3.30
L	0.295	0.305	7.50	7.74
M	0°	10°	0°	10°
S	0.020	0.040	0.51	1.01

CONCLUSION

The project work micro controller based electronic code door lock system is successfully developed and designed and fabricated on demonstrated unit. This project work is designed to control four parameters; it can be further expanded to many other parameters. In addition to control the door either locking the door and unlocking the door.

BIBLIOGRAPHY

The following is the list of study material referred while designing developing and fabricating the project work. The study materials include the textbooks and websites. As the technology is growing at a rapid rate in this direction, tremendous potential is available for the entrepreneur's and hobbyists to take up new assignments and new projects. Hence lot of study materials. The listing are given below.

REFERENCE BOOKS

- The 8051/8052 Microcontroller by: Craig Steiner
- Embedded Systems by Dr. K.V.K.K Prasad
- Mohammad ali mazidi and robin D Mckinlay,"The 8051 microcontroller and embedded system using assembly and c", Pearson Education.
- Designing Embedded Hardware :John Catsaulis by Shroff Pub.
- The 8051 Microcontroller Architecture Programming and Applications by: Kenneth J. Ayala, West Publishing company.

WEBSITES:

- http://www.datasheetcatalog.com/datasheets_pdf/K/A/7/8/KA78XX.shtm
- http://www.dxportal.com/stats/content/250368-IC-4049.html
- http://www.mashhadkit.com/2004/Learning/images/IC4049.gif
- http://www.bitsbox.co.uk/images/7805.jpg
- www.8051projects.info/datasheets/**BC548**.pdf
- www.datasheetpro.com/496229_view_**BC558**_datasheet.html
- http://www.electronic-circuits-diagrams.com/alarmsimages/alarmsckt6.shtml
- http://www.sapconinstruments.com/conductivity/water-level-switch.html
- www.electronicstutotials.com/oscillators/crystal- oscillators.htm