VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

J Component report

Programme: MCA

Internet of Things (ITA6010)

***Slot*: E2**

LPG LEAKAGE DETECTOR WITH BUZZER INDICATION USING MICROCONTROLLER

‘

Submitted in partial fulfillment for the course

by

NIKHIL SARAOGI (21MCA1080), LAKSHAY (21MCA1056)

NIKHIL TEOTIA (21MCA1024), JOY BHATTACHARJEE (21MCA1015)

ALIN PAUL (21MCA1054), VINAYAK SINGH (21MCA1064)

June, 2022

FACULTY NAME	**FACULTY SIGN**
DR. M. VERGIN RAJA SAROBIN	

ABSTRACT

Liquefied Petroleum Gas (LPG) is a main source of fuel, especially in urban areas because it is clean compared to firewood and charcoal. Gas leakage is a major problem in the industrial sector, residential premises, etc. Nowadays, home security has become a major issue because of increasing gas leakage. Gas leakage is a source of great anxiety with ateliers, residential areas and vehicles like Compressed Natural Gas (CNG), buses, and cars which are run on gas power. One of the preventive methods to stop accidents associated with the gas leakage is to install a gas leakage detection kit at vulnerable places. The aim of our project is to propose and discuss a design of a gas leakage detection system that can automatically detect, alert and control gas leakage. This proposed system also includes an alerting system for the users. The system is based on a sensor that easily detects a gas leakage.

INDEX

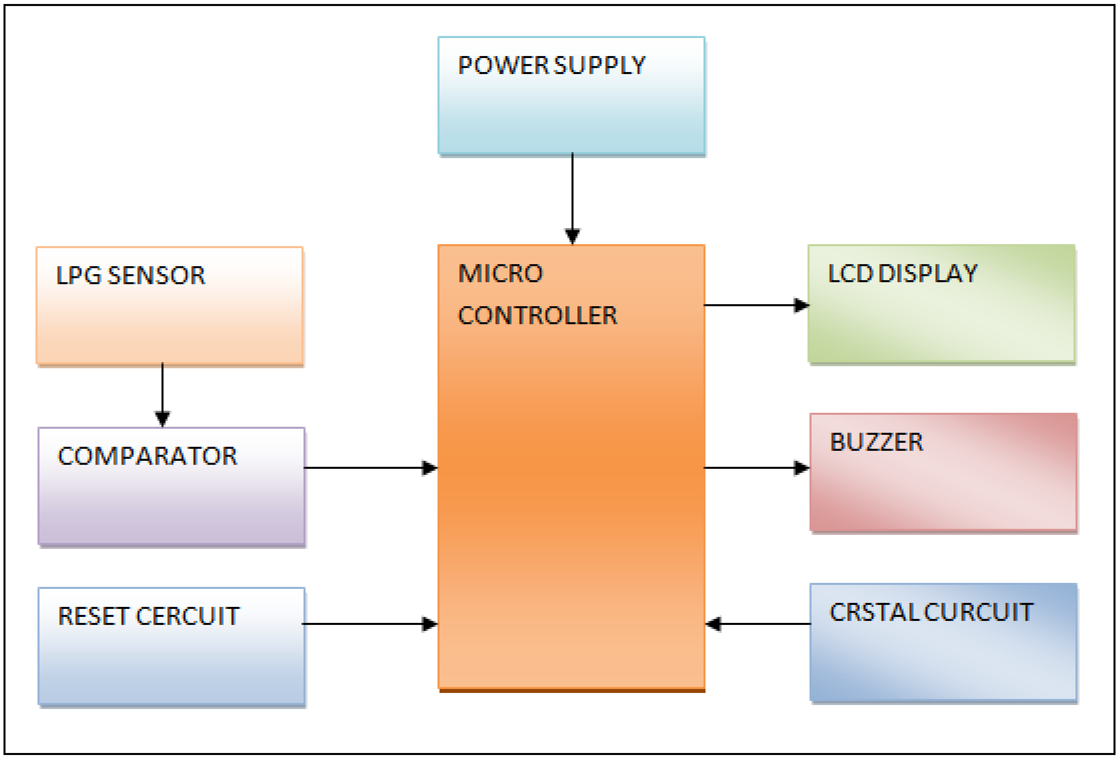
INTRODUCTION

Gas leakage is a serious problem and nowadays it is observed in many places like residences, industries, and vehicles like Compressed Natural Gas (CNG), buses, cars, etc. It is noticed that due to gas leakage, dangerous accidents occur. The Liquefied petroleum gas (LPG), or propane, is a flammable mixture of hydrocarbon gases used as fuel in many applications like homes, hostels, industries, automobiles, and vehicles because of its desirable properties which include high calorific value, less smoke, less soot, and meager harm to the environment. Liquid petroleum gas (LPG) is highly inflammable and can burn even at some distance from the source of leakage. This energy source is primarily composed of propane and butane which are highly flammable chemical compounds. These gases can catch fire easily. In homes, LPG is used mainly for cooking purposes. When a leak occurs, the leaked gases may lead to an explosion. Gas leakage leads to various accidents resulting in both material loss and human injuries. Home fires have been occurring frequently and the threat to human lives and properties has been growing in recent years. The risks of explosion, fire, suffocation are based on their physical properties such toxicity, flammability, etc. The number of deaths due to the explosion of gas cylinders has been increasing in recent years. The Bhopal gas tragedy is an example of accidents due to gas leakage.

There are various projects based on sensors. And this is a low cost Microcontroller based project. Microcontroller based LPG gas leakage system using MQ6 (LPG Gas) sensor with Buzzer indication has applications in various areas including Industrial application, domestic application. This system is very useful in hotels, home, car, LPG agencies. This system has advance safety standard and most important this helps prevent accident occurred because of fire. It is also useful in protecting human life, wealth and property.

MQ6 (LPG Gas) sensor is used to detect LPG leakage this sensor has quick response time it responding very short period of time. Output of MQ6 sensor is given to Comparator. And output of comparator IC is given to microcontroller 89s51. Liquid crystal display – LCD display shows a message when MQ6 crosses threshold level. And also a Buzzer is turned on to give Alert indication. We have provided a potentiometer which is used to vary the threshold level of comparator which decides the threshold level of leakage condition.

BLOCK DIAGRAM



BLOCK DIAGRAM DESCRIPTION

1) LPG gas sensor:-

LPG sensor is the sensor that measures the amount of gas in the surrounding environment. It gives voltage output. The output is in the form of analog voltage. This output varies from 0 to 5 volts.

2)MICRO-CONTROLLER (8051):-

It is the major part of the system. The 8051 has one serial port that receives and transmits data. Transmission and reception can take place simultaneously. The four communication modes possible with 8051 present the system designer and programmer with opportunities to conduct very sophisticated data communication network. It is the heart of the system which controls all the inputs and the controlling action to be taken at the output. Microcontroller used here is the AT89S51.

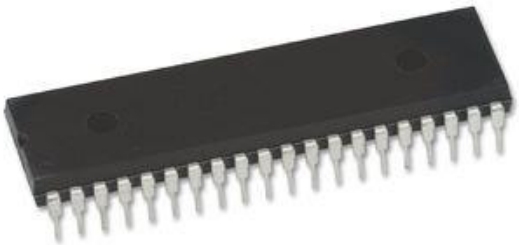


Figure: Photograph of an 89s51 microcontroller.

3) LCD DISPLAY:-

Liquid Crystal Display which is commonly known as LCD is an Alphanumeric Display it means that it can display Alphabets, Numbers as well as special symbols thus LCD is a user friendly Display device which can be used for displaying various messages unlike seven segment display which can display only numbers and some of the alphabets. The only disadvantage of LCD over seven segment is that seven segment is robust display and be visualized from a longer distance as compared to LCD. Here we have used 16 x 2 Alphanumeric Display which means on this display we can display two lines with maximum of 16 characters in one line.

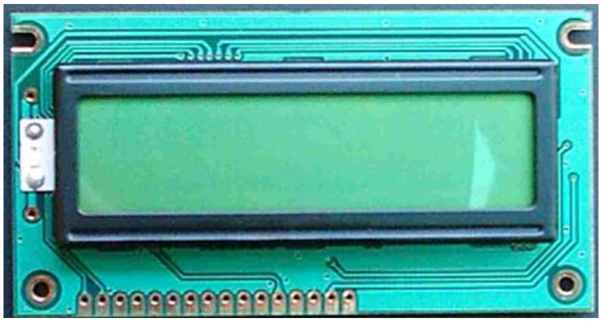


Figure: Photograph of A typical 16 by 2 alphanumeric LCD display

4) Comparator:

The output voltage of LPG sensor is in analog form. And the 89s51 microcontroller can not read analog voltage. That’s why we have used comparator in our project. The output voltage from sensor and voltage divider is given to the comparator. This comparator compares the voltage from sensor and gives output to the microcontroller.

5) Buzzer:

We have used a piezoelectric buzzer in our project. It is used for alert indication. The buzzer will be turned on when sensor value goes above the desired value.

HARDWARE DESCRIPTION

MICROCONTROLLER

Design specification and Features of Microcontroller 89S51

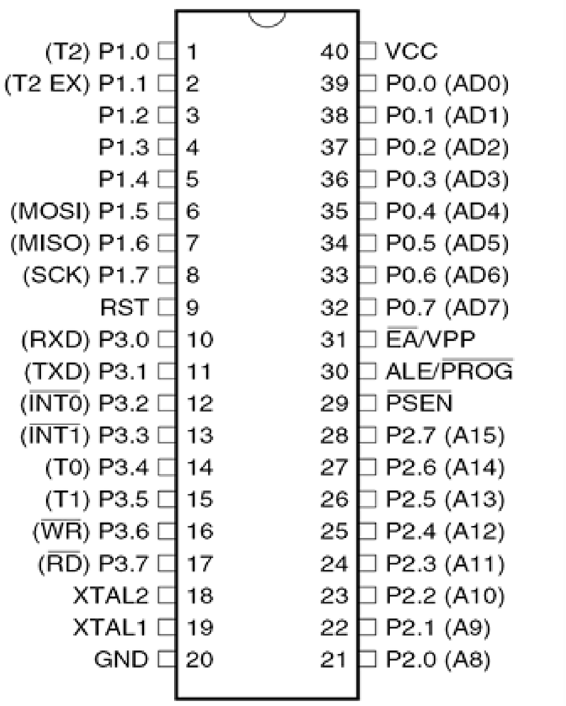
- Compatible with MCS-51 (8051 series microcontroller) products.
- 4k bytes of I system reprogrammable flash memory.
- Endurance: 1000 write/erase cycles.
- Fully static operation: 0Hz to 24 MHz
- Three level program memory lock.
- 128*8 bit internal ram.
- 32 programmable I/O lines.
- Two 16 bit timers/counters.
- Six interrupt sources.
- Programmable serial channel.
- Low power idle and power down modes.

Description

The AT89s52 is a low power, high performance CMOS 8 bit microcomputer with 4k bytes of flash programmable and erasable read only memory (EEPROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51 instruction set and pin out. The on chip flash allows 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 Atmel AT89S51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89S52 provides the following standard feature: 4k bytes of flash, 128 bytes of ram, 32 I/O lines, two 16 bit timers/counters, five vector two-level interrupt architecture, a full duplex serial port, and on-chip oscillator and clock circuitry. In addition the AT89S52 is designed with a static logic for operation down to zero frequency and supports two selectable power saving modes. The IDLE mode stops the CPU while allowing the ram, timer/counter, serial port and interrupts system to continue functioning. The power down mode saves the ram contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

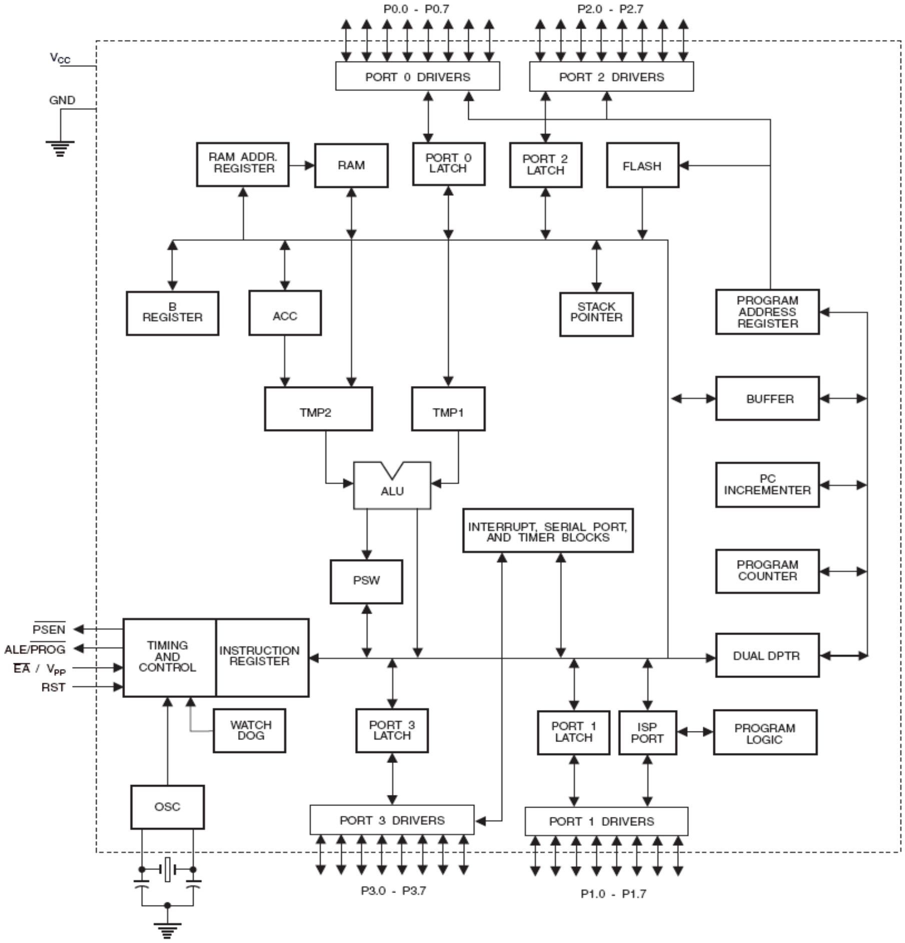


PIN CONFIGURATION 40-lead PDIP

Pin description

1) VCC : Supply voltage : Pin no 40
2) GND : Ground : Pin no 20
3) XTAL1 : Crystal terminal 1 : Pin no 18
4) XTAL2 : Crystal terminal 2 : Pin no 19
5) RST : Reset Pin : Pin no 9

ARCHITECTURE:



Port0:

Port 0 is an 8 bit bi-directional I/O port. As an output port each pin can sink eight TTL input. When 1's are written to port 0 pins, the pins can be used as high impedance inputs. Port 0 may also be configured to be the multiplexed low order address/data bus during the access to external program memory. In this mode p0 has internal pull ups. Port 0 also receives the code bytes during the flash programming, and outputs the code bytes during program verification. External pull ups are required during the program verification.

Port1:

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. Port 1 also receives the low-order address bytes during Flash programming and verification.

Port2:

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 uses 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 uses 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.

Port3:

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 inter- Port Pin Alternate Functions P1.5 MOSI (used for In-System Programming) P1.6 MISO (used for In-System Programming) P1.7 SCK (used for In-System Programming)5 2487D–MICRO–6/08 AT89S51 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 receives some control signals for Flash programming and verification. Port 3 also serves the functions of various special features of the AT89S51, as shown in the following table.

- P3.0: RXD (serial input port)
- P3.1: TXD (serial output port)
- P3.2: INT0 (external interrupt 0)
- P3.3: INT1 (external interrupt 1)
- P3.4: T0 (timer0 external input)

- P3.5: T1 (timer1 external input)
- P3.6: WR (external data memory write strobe)
- P3.7: RD (external data memory read strobe)

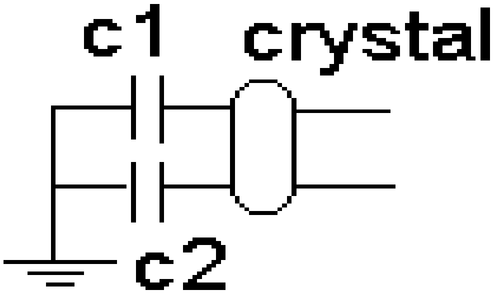
BASIC CIRCUIT OPERATION: - The main circuitries involved in it are: -

<u>1) CRYSTAL CIRCUIT</u>: - This circuit gives the required clock pulses to the microcontroller to give it the sense of the reference time.

XTAL1 - Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2 - Output from the inverting oscillator amplifier

Diagram: -



Working: - The circuit consists of one crystal and two capacitors. The crystal is used to give the microcontroller the required periodic pulses to make it function properly. The crystal used in the project is of 12 MHz The two capacitors are connected to two pins of the crystal and are grounded at the other ends

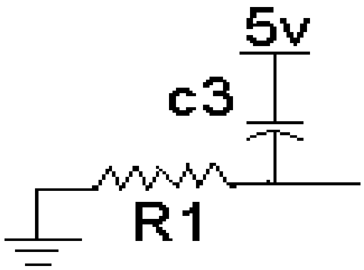
Oscillator characteristics:

XTAL1 and XTAL2 are the input and output, respectively of an inverting amplifier which can be configured for use as an on chip oscillator. Either a quartz crystal or a ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown. There are no requirements on the duty cycle of the external clock signal, since the input to the external clock circuitry is through a divide by two flip flops, but minimum & maximum voltage and high & low time specification must be observed.

<u>2) RESET CIRCUIT</u>: - This circuit gives microcontroller the starting pulse required to start the operation from the start. Unless this pulse is given, microcontroller doesn't start functioning.

A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 98 oscillator periods after the Watchdog times out. The DIS-RTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

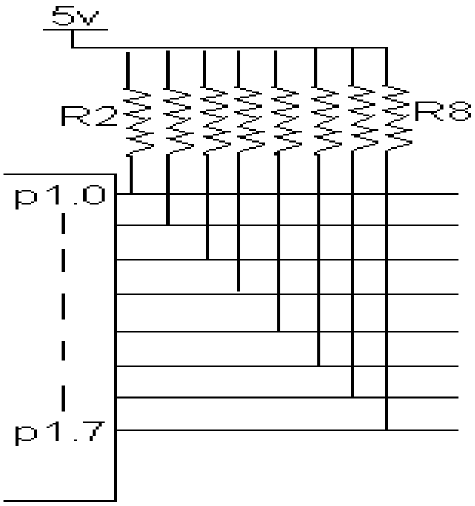
Diagram: -



Working: - The circuit gives the required starting pulse to the microcontroller to start the operation from the very beginning. The 89S51 microcontroller requires the active high reset pulse. So the capacitor is connected to positive supply and the resistor is grounded.

<u>3) Pull up Resistors:</u>The pull-up resistors are required to source the required current to the 7-segment display, which the microcontroller alone is not capable of.

<u>Diagram</u>: -



Working: - The microcontroller pins cannot be connected to the LCD directly because the microcontroller cannot supply all the required current. So the required remaining current is provided through the pull-up resistors. They are designed to supply just the enough current to the motor driver circuit.

4) ALE/PROG

Address Latch Enable (ALE) 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.

5) PSEN:

Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S51 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.

6) EA/VPP:

EA means 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. Port Pin Alternate Functions P3.0 RXD (serial input port) P3.1 TXD (serial output port) P3.2 INT0 (external interrupt 0) P3.3 INT1 (external interrupt 1) P3.4 T0 (timer 0 external input) P3.5 T1 (timer 1 external input) P3.6 WR (external data memory write strobe) P3.7 RD (external data memory read strobe) 6 2487D–MICRO–6/08 AT89S51 EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming.

Idle mode:

In idle mode, the CPU puts itself to sleep while all the on chip peripherals remain active. The mode is invoked by software.

Power down mode:

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed.

POWER SUPPLY MODULE

Block Diagram

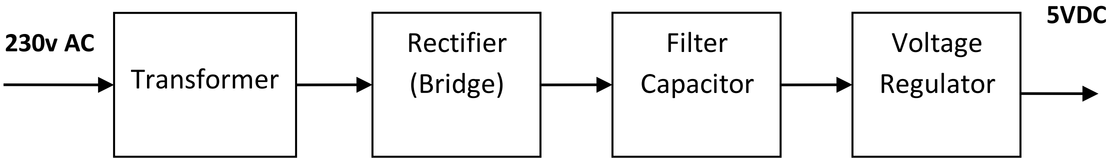


Fig: Block diagram of Power Supply Module

The 230 AC mains supply is given to the transformer primary to get the required voltage at the secondary. Then it is applied to the bridge rectifier, which converts the sinusoidal input into full wave rectified output. The output of the rectifier contains some ripple voltage. To remove this voltage filter circuit is used. A ripple voltage is nothing but a small value of AC over DC signal. Then a pure DC is given to the regulator. The function of the regulator is to give the constant or stable output DC in spite of changes in the load current.

The reasons for choosing IC regulator is that they are versatile in operation and relatively inexpensive with features like programmable output, current/voltage boosting, internal short circuit current limiting, thermal shutdown. The 78XX are popularly known for regulation has been used. The 78XX series is a 3-terminal positive voltage regulator and 79XX series is a 3-terminal negative voltage regulator. As name suggests it transforms the voltage level from one level to another. Transformer used is the step down transformer to step 230 V to +9 V. It provides isolation too from the mains.

1) TRANSFORMER: Transformer is the main component of the power supply module. There are two types of transformer namely Step up and Step Down. We have used Step down transformer as we have to generate 5 volts and 12 volts DC supply from the 230 volts input AC supply so we have used 15 volts / 500 mA transformers which mean its output will be 15 volts AC with current rating of 500 mA.

Types of Transformer:

1) Core type Transformer
2) Shell type Transformer
3) Berry type Transformer

2) RECTIFIER:

Rectifier is used to rectify the negative half cycles of the output signal of the secondary of the transformer. So at the input of the rectifier We have AC signal with both positive and negative cycles and at the output of the rectifier We have signal with only positive cycles but as this signal is also AC We have to use capacitor to filter out the AC of the output signal. There are mainly three types of rectifiers namely half wave, Full wave and Bridge rectifier. Out of these three we have used Bridge rectifier since it give more efficiency. A full wave rectifier converts the whole of the input waveform to one of constant polarity (positive or negative) at its output by reversing the negative (or positive) portions of the alternating current waveform .The positive (negative) portions thus combine with the reversed negative (positive) portions to produce an entirely positive (negative) voltage/current waveform. For single phase AC is center tapped, and then two diodes back to back (i.e. anode to anode or cathode to cathode) form a full wave rectifier.

Rectifier designing 1N4007 diodes are used to build circuit of full wave bridge rectifier

- Surge overload rating - 50 amperes peak
- Ideal for printed circuit board
- Reliable low cost construction utilizing molded plastic technique results in inexpensive product
- Mounting Position: Any

For diode design:-

PIV = Vm

Vm = E0 max + 2 Vf

= 10.7 + 1.4 V

= 12.1 V

I0 = Il/2

= 116.2 mA/ 2

= 58.1 mA

Peak repetitive current

Ifm = [Il (t1+t2)]/t2

T2 = time for 90° - time for θ1

= 5ms - 3.4ms

= 1.2ms

Ifm = 116.2mA (8.6ms+1.2ms) /1.2ms

= 833mA

From above specification diode 1N4007 is selected.

3) FILTER CAPACITOR:

As mentioned above we have to use filter capacitor to remove the AC signal from the output of rectifier. Filter capacitor is used in order to remove ripples from the pulsating DC and convert it to unregulated DC.

A capacitor is an electrical device that can store energy in the electric field between a pair of closely spaced conductors (called ‘plates’). When voltage is applied to the capacitor, electric charges of equal magnitude, but opposite polarity, build up on the plate.

Capacitors are used in electrical circuits as energy storage devices. They can also be used to differentiate between high frequency and low frequency signals and this makes them useful in electronic filters. These small deviations from the ideal behavior of the device can become significant when it is operating under certain conditions, i.e. high frequency, high current, or temperature extremes.

PIV = 100V

I = 1A

For filter capacitor design:-

C = (I1 * t1)/Vr

Vr = ripple voltage

IL = load current

T1 = time during which the capacitor being discharge by load current

Vr = ripple voltage 10% of output voltage

Vr = 1.0 V

Frequency 50 HZ

T1 = 1/50 = 20 ms

T for 360° = 20ms

For 180° = 10ms

For 60° = 20ms * (60°/360)

= 3.4ms

For bridge:-

T1 = [time for 90° + time for θ1]

= 5ms + 3.4ms

= 8.4ms

I1 = load current supplied to various IC

I1 = current required for LCD + o/p current of 89S51 +o/p current of max232 + current required for LM35 + Current required for heart beat sensor + current required

For GPS SR-87

= 3mA + 40mA + 8mA + 0.060mA + 22mA + 40mA

= 113.06Ma

C = I1 * T1/ Vr

= 113.06 * 8.4 * 10^-6 / 1

= 949.704 µF

Thus this 949.704 µF value can be approximated to 1000 µF. Thus we will use 1000 µF capacitor before IC 7805, which is used for improving Frequency Response.

4) VOLTAGE REGULATOR:

Two separate voltage regulators are used after the filter capacitor so as to generate constant DC voltage supply of 5 volts and 12 volts. We have used 7805 and 7812 as a voltage regulator. Both of them are three pin IC which are namely input, ground and output. We have to give output of filter capacitor to the input of regulator, and we get 5 volts and 12 volts supply at the output pin of the respective regulator.

Transformer selection:we require 12V for min input for IC 7805

= Drop across IC 7805 + Required Output voltage

= 3 V+ 5V

= 8 V

So at Input of 7805 we required 8 V with margin

Consider drop across diode 0.7V so 2 diodeconducts drop is 1.4 V

= 1.4 V +8 V

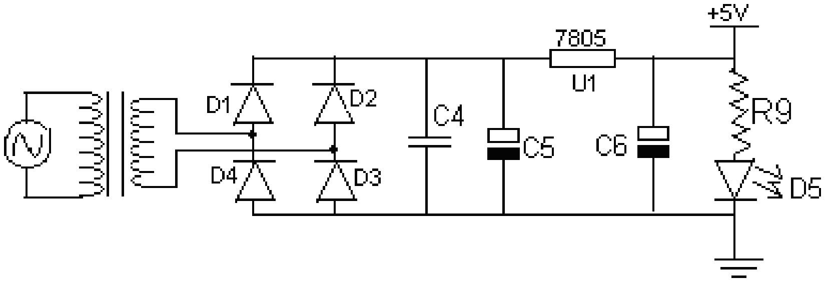
= 9.4 V

So at secondary we required 10 V

Fixed voltage regulator IC 7805 produces +5V regulated output voltage with respect to the ground.

- Output Current in Excess of 1.0 A
- No External Components Required
- Internal Thermal Overload Protection
- Internal Short Circuit Current Limiting
- Output Transistor Safe–Area Compensation
- Output Voltage Offered in 2% and 4% Tolerance
- Available in Surface MountD2PAK, DPAK and Standard 3–Lead
- Transistor Packages

Power supply Diagram: -



Working: - The power supply gives +5v and +12v supply to the circuit. The power supply consists of four stages namely transformer, rectifier, filter, and regulator. Transformer is a step-down transformer taking input of 230v AC and giving output of 15v at the secondary. This 15v AC is rectified by bridge rectifier consisting of four diodes, which converts the AC wave into fully rectified wave. The next stage is the filter stage consisting of capacitor, which converts the fully rectified wave into the DC wave with some ripple. Last stage is the regulator stage. Regulator removes the entire ripple and gives pure DC. The LED is connected to indicate that power supply is ON.

DETAILED DESCRIPTION OF 16X2 LCD DISPLAY

LCD stands for Liquid Crystal Display. The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters.

Pin Description

Most LCDs with two controllers has 16 Pins. Pin description is shown in the table below.

Pin No.	**Name**	**Description**
Pin no. 1	GND	Power supply (GND)
Pin no. 2	VCC	Power supply (+5V)
Pin no. 3	VEE	Contrast adjust
Pin no. 4	RS	0 = Instruction input
		1 = Data input
Pin no. 5	R/W	0 = Write to LCD module
		1 = Read from LCD module
Pin no. 6	EN	Enable LCD
Pin no. 7	D0	Data bus line 0 (LSB)
Pin no. 8	D1	Data bus line 1
Pin no. 9	D2	Data bus line 2
Pin no. 10	D3	Data bus line 3
Pin no. 11	D4	Data bus line 4
Pin no. 12	D5	Data bus line 5
Pin no. 13	D6	Data bus line 6
Pin no. 14	D7	Data bus line 7 (MSB)
Pin no. 15	VCC	VCC Pin for Backlight
Pin no. 16	GND	GND Pin for Backlight

Table No.4.1: Pin description of the LCD

4.2 DDRAM - Display Data RAM

Display data RAM (DDRAM) stores display data represented in 8-bit character codes. Its extended capacity is 80 X 8 bits, or 80 characters. The area in display data RAM (DDRAM) that is not used for display can be used as general data RAM. So whatever you send on the DDRAM is actually displayed on the LCD.

4.3 BF - Busy Flag

Busy Flag is a status indicator flag for LCD. When we send a command or data to the LCD for processing, this flag is set (i.e. BF =1) and as soon as the instruction is executed successfully this flag is cleared (BF = 0). This is helpful in producing and exact amount of delay. To read Busy Flag, the condition RS = 0 and R/W = 1 must be met and The MSB of the LCD data bus (D7) act as busy flag. When BF = 1 means LCD is busy and will not accept next command or data and BF = 0 means LCD is ready for the next command or data to process.

4.4 Instruction Register (IR)andData Register (DR)

There are two 8-bit registerscontroller Instruction and Data register. Instruction register corresponds to the register where you send commands to LCD e.g. LCD shift command, LCD clear, LCD address etc. and Data register is used for storing data which is to be displayed on LCD. When send the enable signal of the LCD is asserted, the data on the pins is latched in to the data register and data is then moved automatically to the DDRAM and hence is displayed on the LCD.

4.5 Commands and Instruction set:

Only the instruction register (IR) and the data register (DR) of the LCD can be controlled by the MCU. Before starting the internal operation of the LCD, control information is temporarily stored into these registers to allow interfacing with various MCUs, which operate at different speeds, or various peripheral control devices. The internal operation of the LCD is determined by signals sent from the MCU.

4.6 Sending Commands to LCD

To send commands we simply need to select the command register. Everything is same as we have done in the initialization routine. But we will summarize the common steps and put them in a single subroutine.

Following are the steps:

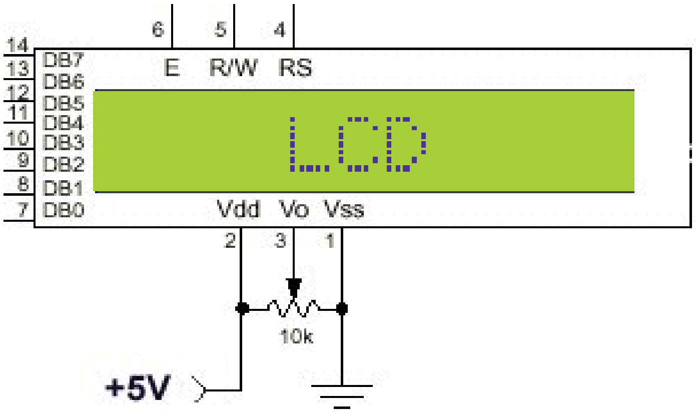
- Move data to LCD port
- Select command register
- Select write operation
- Send enable signal
- Wait for LCD to process the command

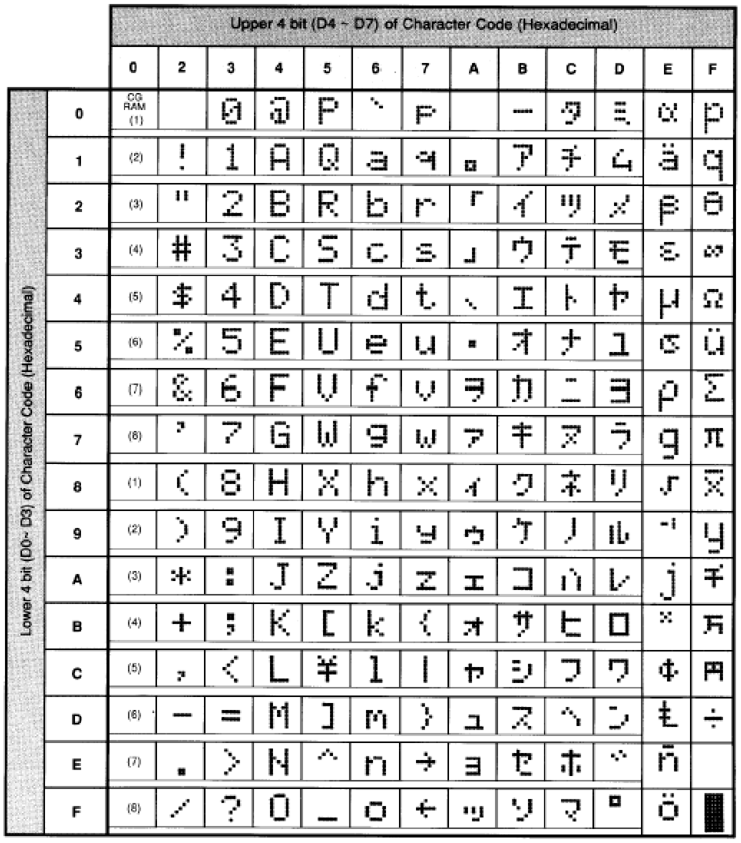
LCD DISPLAY INTERFACING

In market various displays are available like 7-segment,5*7 matrix LED and LCD, bar graph, LCD, etc. It's important for deciding the required display set for our project. Selection of display depends on various factors like power consumption, ambient light conditions, surrounding temperature, visibility from long distance, total information to be display, cost of display, circuit/lines required for display interfacing, etc.

The most common way to accomplish this is with the LCD(Liquid Crystal Display).LCDs have become a cheap and easy way to get text display for an embedded system.

Common LCD displays are set up as 16 to 20 characters by 1 to 4 lines and noted as 16*2, 20*2, 16*4, 20*4, etc. Following figure shows the basic pin diagram of 16*2 LCD display.



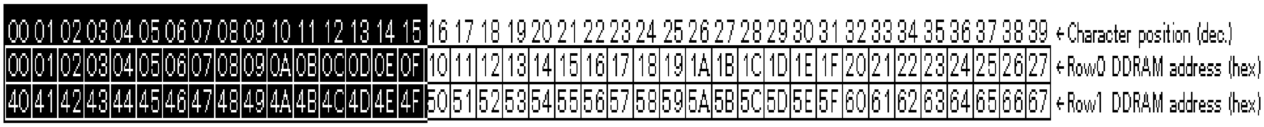


		Upper 4 bit (D4 ~ D7) of Character Code (Hexadecimal)												
		0	2	3	4	5	6	7	A	B	C	D	E	F
Lower 4 bit (D0~ D3) of Character Code (Hexadecimal)	0	CG RAM (1)		0	@	P	`	p		ー	タ	ミ	α	p
	1	(2)	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
	2	(3)	"	2	B	R	b	r	「	イ	ツ	メ	β	θ
	3	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
	4	(5)	$	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
	5	(6)	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
	6	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
	7	(8)	'	7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
	8	(1)	(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
	9	(2)	)	9	I	Y	i	y	ゥ	ケ	ノ	ル	⁻¹	y
	A	(3)	*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
	B	(4)	+	;	K	[	k	{	ォ	サ	ヒ	ロ	ˣ	万
	C	(5)	,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
	D	(6)	-	=	M	]	m	}	ュ	ス	ヘ	ン	Ł	÷
	E	(7)	.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
	F	(8)	/	?	O	_	o	←	ッ	ソ	マ	゜	ö	█

Figure: ASCII code table, showing hex code for all characters, numbers and alphabets

Display Data RAM (DDRAM):

Display data RAM (DDRAM) is where you send the characters (ASCII code) you want to see on the LCD screen. It stores display data represented in 8-bit character codes. Its capacity is 80 characters (bytes).Below you see DD RAM address layout of a 2*16 LCD.



00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	←Character position (dec.)
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	←Row0 DDRAM address (hex)
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	←Row1 DDRAM address (hex)

Figure 3.5DDRAM Address Layouts

In the above memory map, the area shaded in black is the visible display (For 16x2 displays). For first line addresses for first 15 characters is from 00h to 0Fh. But for second line address of first character is 40h and so on up to 4Fh for the 16th character. So if you want to display the text at specific positions of LCD, we require to manipulate address and then to set cursor position accordingly.

REGISTERS:

It has two 8-bit registers, an instruction register (IR) and a data register (DR). The IR stores instruction codes. The DR temporarily stores data to be written into DDRAM or CGRAM and temporarily stores data to be read from DDRAM or CGRAM. Data written into the DR is automatically written into DDRAM or CGRAM by an internal operation. These two registers can be selected by the register selector (RS) signal. See the table below:

Register Selection

RS	R/W	Operation
0	0	IR write as an internal operation (display clear, etc.)
0	1	Read busy flag (DB7) and address counter (DB0 to DB6)
1	0	DR write as an internal operation (DR to DDRAM or CGRAM)
1	1	DR read as an internal operation (DDRAM or CGRAM to DR)

Table 3.1: Register Selection

Busy Flag (BF):

When the busy flag is 1, the LCDis in the internal operation mode, and the next instruction will not be accepted. When RS = 0 and R/W = 1 (see the table above), the busy flag is output to DB7 (MSB of LCD data bus). The next instruction must be written after ensuring that the busy flag is 0.

LCD Commands: The LCD’s internal controller accepts several commands and modifies the display accordingly. These commands would be things like:

– Clear screen– Return home

– Shift display right/left

Instruction	Decimal	HEX
Function set (8-bit interface, 2 lines, 5*7 Pixels)	56	38
Function set (8-bit interface, 1 line, 5*7 Pixels)	48	30
Function set (4-bit interface, 2 lines, 5*7 Pixels)	40	28
Function set (4-bit interface, 1 line, 5*7 Pixels)	32	20
Entry mode set	See Below	See Below
Scroll display one character right (all lines)	28	1E
Scroll display one character left (all lines)	24	18

Home (move cursor to top/left character position)	2	2
Move cursor one character left	16	10
Move cursor one character right	20	14
Turn on visible underline cursor	14	0E
Turn on visible blinking-block cursor	15	0F
Make cursor invisible	12	0C
Blank the display (without clearing)	8	08
Restore the display (with cursor hidden)	12	0C
Clear Screen	1	01
Set cursor position (DDRAM address)	128 + addr	80+ addr
Set pointer in character-generator RAM (CG RAM address)	64 + addr	40+ addr

Table 3.2: LCD Commands

LCD Display PINOUT and Description:

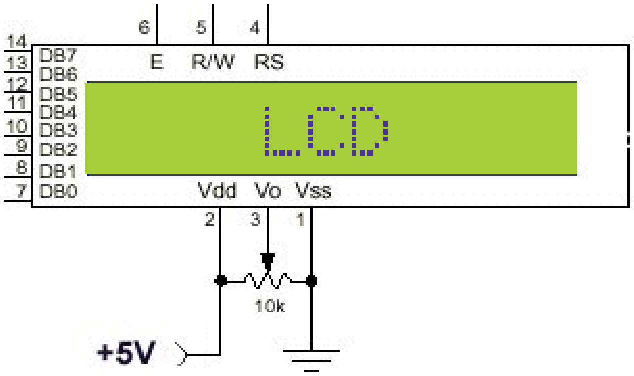


Fig 3.6 LCD

Data pins D7:D0: Pins 7-14:

Bi-directionaldata/command pins.Alphanumeric characters are sent in ASCII format. As shown in figure lcd1.0, there are 8 pins, Pin No.7-14 used for data lines.

RS:Register Select: Pin No.4:

RS = 0; Command Register is selected

RS = 1; Data Register is selected

R/W: Read or Write: Pin No.5:

R/W=0; Write.

R/W= 1; Read

E: Enable (Latch data): Pin No.6:

It is used to latch the data present on the data pins.

A high-to-low edge is needed to latch the data.

Vo: contrast control: Pin No.2:

It is used to control contrast of LCD.

NOTE:

When writing to the display, data is transferred only on the high to low transition of this signal. However, when reading from the display, data will become available shortly after the low to high transition and remain available until the signal falls low again.

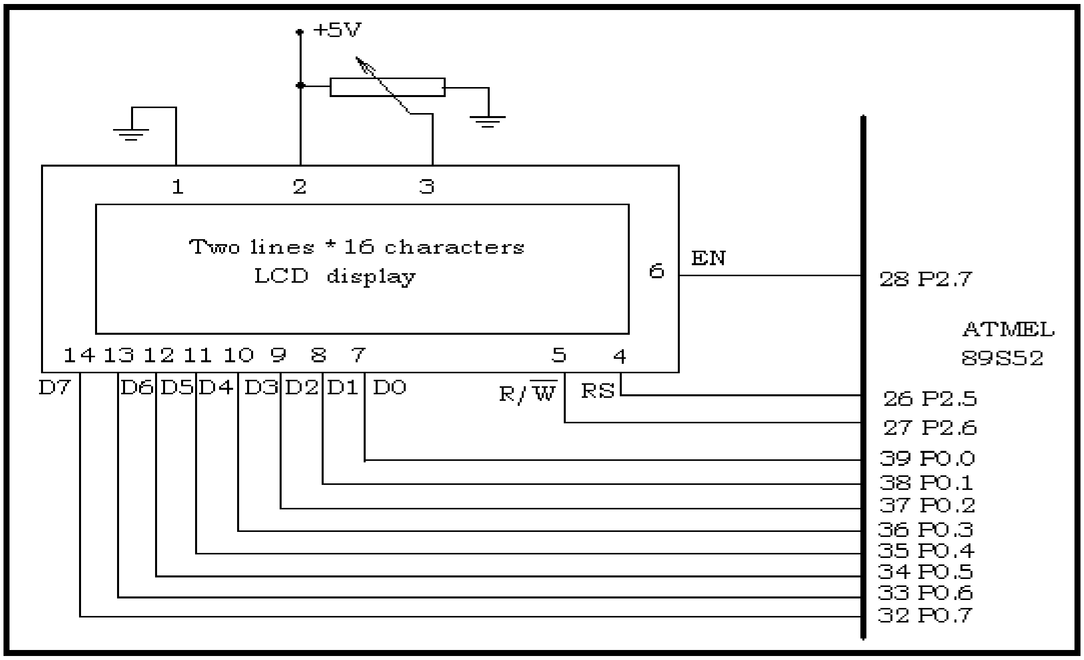
The standard LCD Display requires 3 control lines as well as either 8 I/O lines for the data bus. The three control lines are**EN**, **RS**, and **RW**. Note that the EN line must be raised/lowered before/after each instruction sent to the LCD regardless of whether that instruction is read or write text or instruction. In short, you must always manipulate EN when communicating with the LCD. EN is the LCD's way of knowing that you are talking to it. If you don't raise/lower EN, the LCD doesn't know you're talking to it on the other lines.

Checking the Busy Flag:

We can use subroutine for checking busy flag or just a big (and safe) delay.

1. Set R/W Pin of the LCD HIGH(read from the LCD)
2. Select the instruction register by setting RS pin LOW
3. Enable the LCD by Setting the enable pin HIGH
4. The most significant bit of the LCD data bus is the state of the busy flag (1=Busy, 0=ready to accept instructions/data). The other bits hold the current value of the address counter.

If the LCD never comes out from"busy" status because of some problems, the program will"hang," waiting for DB7 to go low. So in real applications it would be wise to put some kind of time limit on the delay--for example, a maximum of 100 attempts to wait for the busy signal to go low. This would guarantee that even if the LCD hardware fails, the program would not lock up.

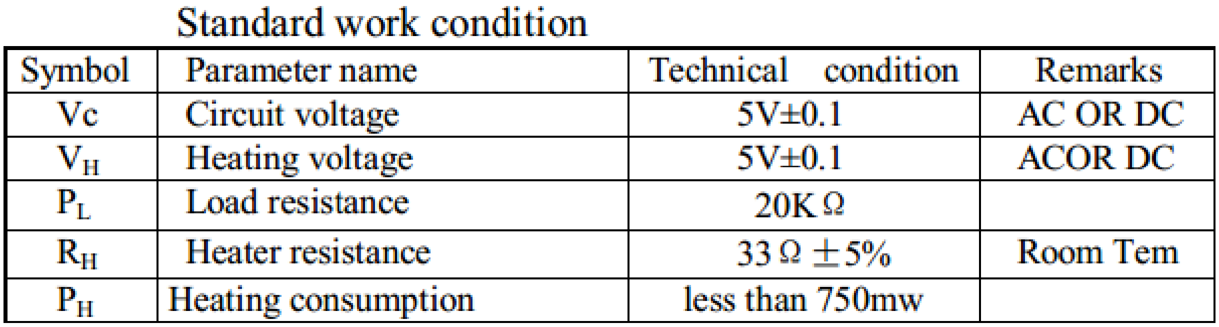

+5V
1
2
3
Two lines * 16 characters
LCD display
6
EN
28 P2.7
ATMEL
89S52
14 13 12 11 10 9 8 7
5
4
D7
D6
D5
D4
D3
D2
D1
D0
R/$\overline{W}$
RS
26 P2.5
27 P2.6
39 P0.0
38 P0.1
37 P0.2
36 P0.3
35 P0.4
34 P0.5
33 P0.6
32 P0.7

LPG GAS SENSOR

This is a simple-to-use liquefied petroleum gas (LPG) sensor, suitable for sensing LPG (composed of mostly propane and butane) concentrations in the air. The MQ-6 can detect gas concentrations anywhere from 200 to 10000ppm.

This sensor has a high sensitivity and fast response time. The sensor's output is an analog voltage.

Connections: Connecting five volts across the heating (H) pins keeps the sensor hot enough to function correctly. Connecting five volts at either the A or B pins causes the sensor to emit an analog voltage on the other pins. A resistive load between the output pins and ground sets the sensitivity of the detector. Please note that the picture in the datasheet for the top configuration is wrong. Both configurations have the same pinout consistent with the bottom configuration. The resistive load should be calibrated for your particular application using the equations in the datasheet, but a good starting value for the resistor is 20 kΩ.



Standard work condition

Symbol	Parameter name	Technical condition	Remarks
Vc	Circuit voltage	5V±0.1	AC OR DC
V_H	Heating voltage	5V±0.1	ACOR DC
P_L	Load resistance	20KΩ	
R_H	Heater resistance	33Ω ±5%	Room Tem
P_H	Heating consumption	less than 750mw	

This sensor is used in gas leakage detecting equipment for detecting of LPG, iso-butane, propane, LNG combustible gases. The sensor does not get trigger with the noise of alcohol, cooking fumes and cigarette smoke. LPG gas sensor has various applications which includes Gas leak detection system, Fire/Safety detection system, Gas leak alarm / Gas detector

Features of LPG gas sensor are:

- It gives simple analog output
- It has high sensitivity to LPG, iso-butane, propane
- It has very small sensitivity to alcohol, smoke
- It has very Fast response and wide detection range
- It gives stable performance and long life
- It requires Power supply of 5V

Specification:

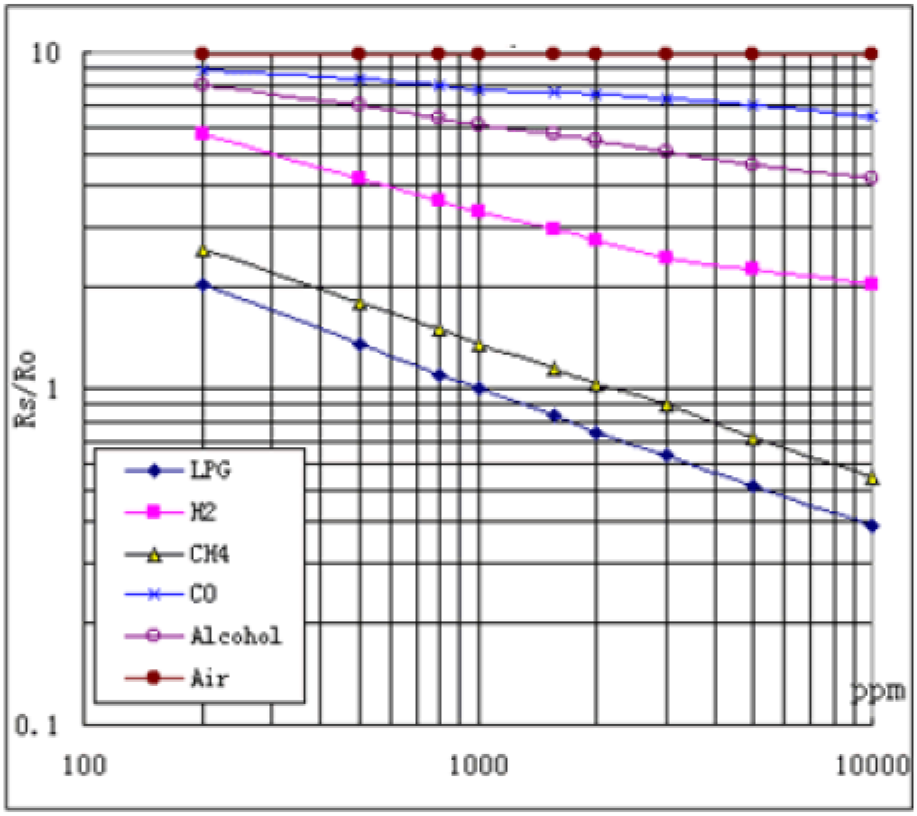
- Interface type: Analog
- Fast response
- Stable and long life
- Simple drive circuit
- Size: 40x20mm

Parameter	**Value**
Target Gas	iso-butane, Propane, LPG
Detection Range	100 to 10000 PPM PPM (part per millions)
Output Voltage Range	0 to 5 VDC
Working Voltage	5 VDC
Current Consumption	≤180 mA
Warmup Time	10 Minutes
Calibrated Gas	1000ppm iso-butane
Response Time	≤10s Seconds
Resume Time	≤30s Seconds
Standard Working Condition	Temperature:-10 to 65 deg C. Humidity: ≤95%RH
Storage Condition	Temperature: -20-70 deg C Hum: ≤ 70%RH

Pin outs

#	Pin	Details
1	GND	Power Supply Ground
2	A.OUT	Analog Voltage Out
3	+5V	Supply voltage DC +5V regulated

Sensitivity



Typical Sensitivity Characteristics of sensor for several gases in their

Temp: 20 deg C

Humidity: 65%

Oxygen concentration: 21%

RL = 10K Ohm

Ro = Sensor resistance at 1000 ppm of LPG in clean air

Rs = Sensor resistance at various concentrations of gases

Deriving Gas concentration from Output Voltage

Here is a the equation which convert analog output to PPM gas concentration

PPM = Analog Voltage in mV x 2

Example: Gas sensor voltage is giving output as

2500mV(2.5V) so the gas concentration in PPM = 2500x2 = 5000 PPM

Here is the table for your reference

PPM:	200	300	500	1000	2000	3000	5000	9000	10000
Voltage(mV)	100	150	250	500	1000	1500	2500	4500	5000

Warm up Time

The sensor needs 10 minutes of warm up time after first power is applied. After 10 minutes you can take its readings. During warm up time the output analog voltage would go up from 4.5V to 0.5V in variation down gradually. During this warm up time the sensor reading should be ignored.

Using the Sensor

The sensor needs 5V to operate, Give regulated +5V DC supply, the sensor will take around 180mA supply. The sensor will heat a little bit since it has internal heater that heats the sensing element.

Testing the sensor

Measure the output voltage through multi-meter between A.OUT and Ground pins or Use a microcontroller to measure the voltage output. Take the sensor near combustible gas place like cooking gas stove with flame off or near bottle of after shave liquid or cigarette light with flame off. You will notice sudden jump in analog voltage output since the gas concentration will increase.

BUZZER

A buzzer 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.In simplest terms, a piezo buzzer is a type of electronic device that’s used to produce a tone, alarm or sound. It’s lightweight with a simple construction, and it’s typically a low-cost product. Yet at the same time, depending on the piezo ceramic buzzer specifications, it’s also reliable and can be constructed in a wide range of sizes that work across varying frequencies to produce different sound outputs.

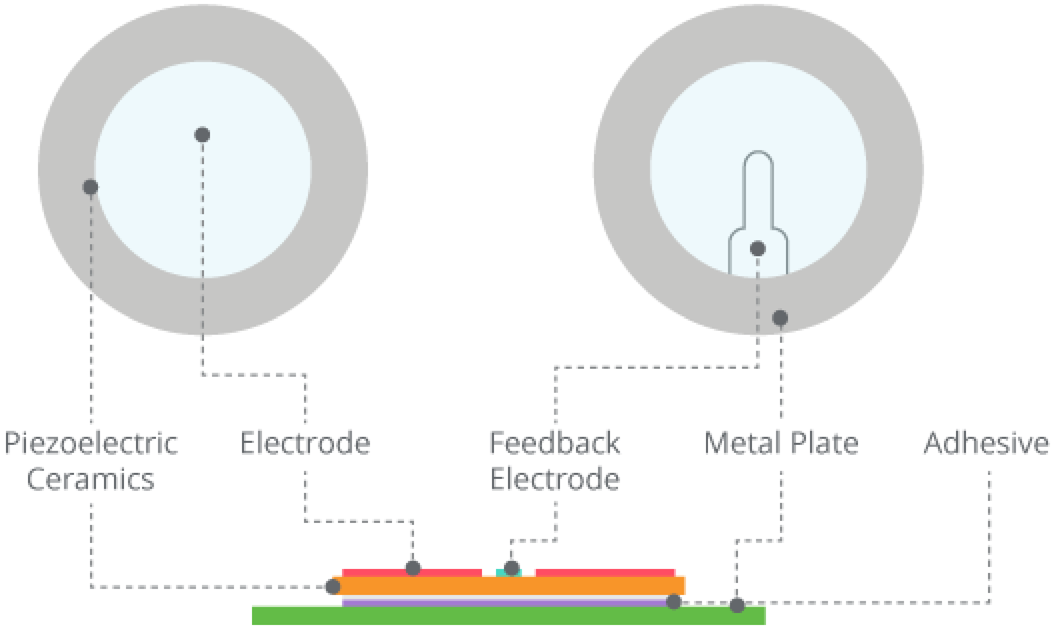
The use of the piezo ceramic buzzer was discovered thanks to an inversion of the piezoelectricity principle that was discovered by Jacques and Pierre Curie back in 1880. They found that electricity could be generated when a mechanical pressure was applied to particular materials — and the inverse was true as well.

We have used piezoelectric type of Buzzer. A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep.

It is a small 30mm round speaker that operates around the audible 2kHz range. We can use these speakers to create simple music or user interfaces.

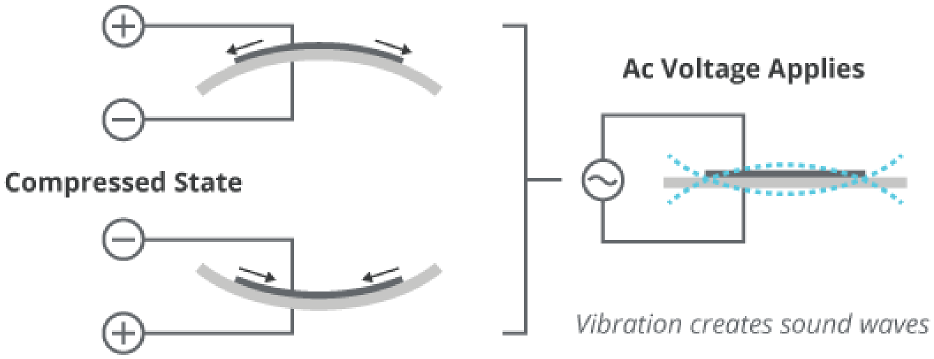
It requires an operating voltage of 3 to 12 V with a mean current of 50mAmp. These speakers also have a typical sound output of 95 dBA and a coil resistance of 42 ±6.3 ohms.

Piezo-ceramic Element Structure



At the heart of all piezo-type buzzers is the piezoelectric element. The piezoelectric element is composed of a piezoelectric ceramic and a metal plate held together with adhesive. Both sides of the piezoelectric ceramic plate contain an electrode for electrical conduction. Piezo materials exhibit a specific phenomenon known as the piezoelectric effect and the reverse piezoelectric effect. Exposure to mechanical strain will cause the material to develop an electric field, and vice versa.

Working Principle of Piezo Buzzers



When an alternating voltage is applied to the piezoceramic element, the element extends and shrinks diametrically. This characteristic of piezoelectric material is utilized to make the ceramic plate vibrate rapidly to generate sound waves.

There are two types of piezo buzzers - transducers and indicators. Transducers consist of a casing, a piezoceramic element and a terminal. In order to operate a transducer, the user must send a square wave signal to the buzzer. Indicators consist of a casing, a piezoceramic element, a circuit board and a terminal. In order to operate an indicator, the user must send the buzzer a specified dc voltage.

Specifications

- Operating Voltage 3 - 24V
- Rated Voltage 12V DC
- Buzzer Type Piezoelectric
- Frequency 3900±500Hz
- Termination Lead Wire
- Drive Method Drive Circuit Built-in
- Sound Pressure Level 95dB
- Max. Current Rating 50mA
- low cost / high sound outputs
- can be mounted in printed circuit boards

Typical Applications of a Piezo Buzzer

Buzzers are typically used for identification and alarm purposes across many major industries. The major application categories that utilize buzzers for indication or alert purposes include: home appliances, automotive electronics, medical, safety and security, industrial, and office automation.

- Alarms / warning devices / automobile alarms
- Pest deterrents
- Computer devices
- Telephones
- Toys / games

RESISTOR

Resistor is an electrical component used to reduce current in a circuit. Its resistance is given by the ratio of voltage applied across its terminals to the current passing through it. Thus a particular value of resistor, for fixed voltage, limits the current through it. They are omnipresent in electronic circuits. The resistor's ability to reduce the current is called resistance and is measured in units of ohms (symbol: Ω)

The different value of resistances are used to limit the currents or get the desired voltage drop according to the current-voltage rating of the device to be connected in the circuit. For example, if an LED of rating 2.3V and 6mA is to be connected with a supply of 5V, a voltage drop of 2.7V (5V-2.3V) and limiting current of 6mA is required. This can be achieved by providing a resistor of 450 connected in series with the LED.

Resistors can be either fixed or variable. The low power resistors are comparatively smaller in size than high power resistors. The resistance of a resistor can be estimated by their color codes or can be measured by a multimeter. There are some nonlinear resistors also whose resistance changes with temperature or light. Negative temperature coefficient (NTC), positive temperature coefficient (PTC) and light dependent resistor (LDR) are some such resistors. These special resistors are commonly used as sensors. Read and learn about internal structure and working of a resistor.

Carbon film resistors:

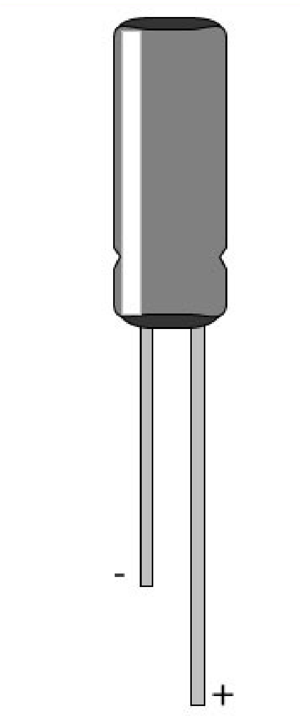
This resistor is formed by depositing a carbon film layer on an insulating substrate. Helical cuts are then made through the carbon film to trace a long and helical resistive path. The resistance can be varied by using different resistivity carbon material and modifying the shape of the resistor. The helical resistive path make these resistors highly inductive and of little use for RF applications.

They exhibit a temperature coefficient between -100 and -900 ppm/ °C. The carbon film is protected either by a conformal epoxy coating or a ceramic tube. The operation of these resistors requires high pulse stability.

Metal Film resistor:

These resistors are made from small rods of ceramic coated with metal (such as a nickel alloy) or metal oxide (such as tin oxide). The value of resistance is controlled mainly by the thickness of the coating layer (the thicker the layer, the lower is the value of resistance). A fine spiral groove can be cut along the rod using a laser to split the carbon or metal coating effectively into a long and spiral strip, which forms the resistor.

CAPACITOR

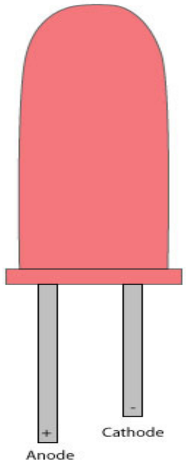


Capacitor is a passive component used to store charge. This component consists of two conductors which are separated by a dielectric medium. The charge (q) stored in a capacitor is the product of its capacitance (C) value and the voltage (V) applied to it. Capacitors offer infinite reactance to zero frequency so they are used for blocking DC components or bypassing the AC signals. The capacitor undergoes through a recursive cycle of charging and discharging in AC circuits where the voltage and current across it depends on the RC time constant. For this reason, capacitors are used for smoothing power supply variations. Other uses include, coupling the various stages of audio system, tuning in radio circuits etc. These are used to store energy like in a camera flash. The capacitance or the potential storage by the **capacitor** is measured in Farads which is symbolized as ‘F’.

Capacitors may be non-polarized/polarized and fixed/variable. Electrolytic capacitors are polarized while ceramic and paper capacitors are examples of non-polarized capacitors. Since capacitors store charge, they must be carefully discharged before troubleshooting the circuits. The maximum voltage rating of the capacitors used must always be greater than the supply voltage.

There are different types of capacitors; they can be fixed or variable. They are categorized into two groups, polarized or non-polarized. Electrolytic capacitors are polarized. Most of the low value capacitors are non-polarized.

LED



Light emitting diodes (LEDs) are semiconductor light sources. The light emitted from LEDs varies from visible to infrared and ultraviolet regions. They operate on low voltage and power. LEDs are one of the most common electronic components and are mostly used as indicators in circuits. They are also used for luminance and optoelectronic applications.

Based on semiconductor diode, LEDs emit photons when electrons recombine with holes on forward biasing. The two terminals of LEDs are anode (+) and cathode (-) and can be identified by their size. The longer leg is the positive terminal or anode and shorter one is negative terminal.

The forward voltage of LED (1.7V-2.2V) is lower than the voltage supplied (5V) to drive it in a circuit. Using an LED as such would burn it because a high current would destroy its p-n gate. Therefore a current limiting resistor is used in series with LED. Without this resistor, either low input voltage (equal to forward voltage) or PWM (pulse width modulation) is used to drive the LED.

When a PN junction diode is forward biased, the electrons and holes move in opposite directions. During this free movement, an electron may fall into a hole which exists at a lower energy level than electron, because of this an electron loses some energy which is released in the form photon and hence light is emitted. This phenomenon is termed as Electroluminescence.

LED (Light Emitting Diode) is a special diode that emits light when an electric voltage is applied to it. It is common electronic component that is being used in devices like TV, computer, etc. generally for indicating purpose. They are available in various colors like red, yellow, green etc.

CRYSTAL OSCILLATOR

A quartz crystal resonator plays a vital role in electronics oscillator circuitry. Sometimes mispronounced as crystal oscillator, it is rather a very important part of the feedback network of the oscillator circuitry. Electronics oscillators are used in frequency control application finding their usage in almost every industry ranging from small chips to aerospace.

A quartz crystal is the heart of such type of resonators. Their characteristics like high quality factor (Q), stability, small size and low cost make them superior over other resonators like LC circuit, turning forks, ceramic resonator etc.

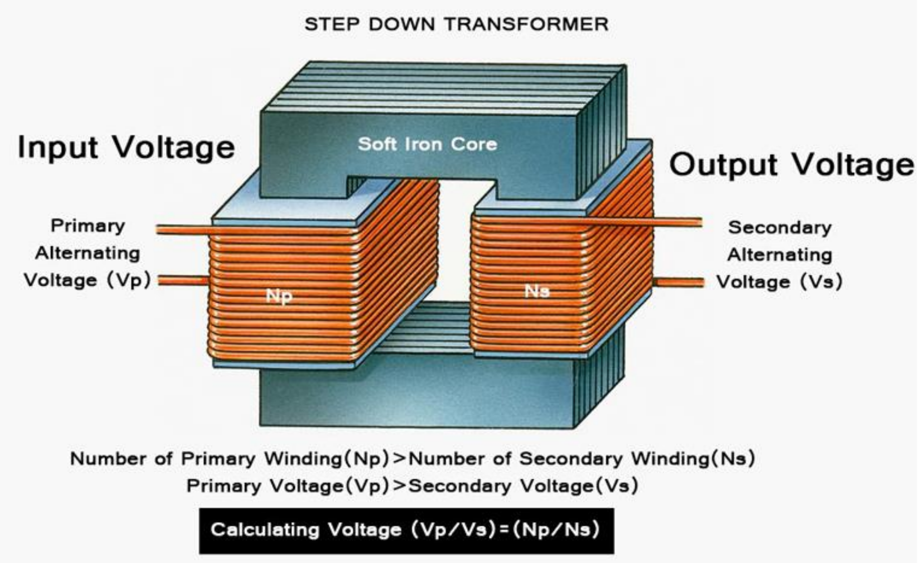
The basic phenomenon behind working of a quartz crystal oscillator is the inverse piezo electric effect i.e., when electric field is applied across certain materials they start producing mechanical deformation. These mechanical deformation/movements are dependent on the elementary structure of the quartz crystal. Quartz is one of the naturally occurring materials which show the phenomena of piezo electricity, however for the purpose of resonator it is artificially developed since processing the naturally occurring quartz is difficult and costly process.

TRANSFORMER

Transformers are static electrical devices with no moving parts, transforming electrical power from one voltage and current setting to another. The frequency of the electrical current remains constant.

Transformers are classified by their function, which is either step up or step down. Step-up transformers increase the voltage of the incoming current, while step-down transformers decrease the incoming current's voltage. The incoming voltage is referred to as the primary voltage, while the outgoing stream is referred to as secondary.

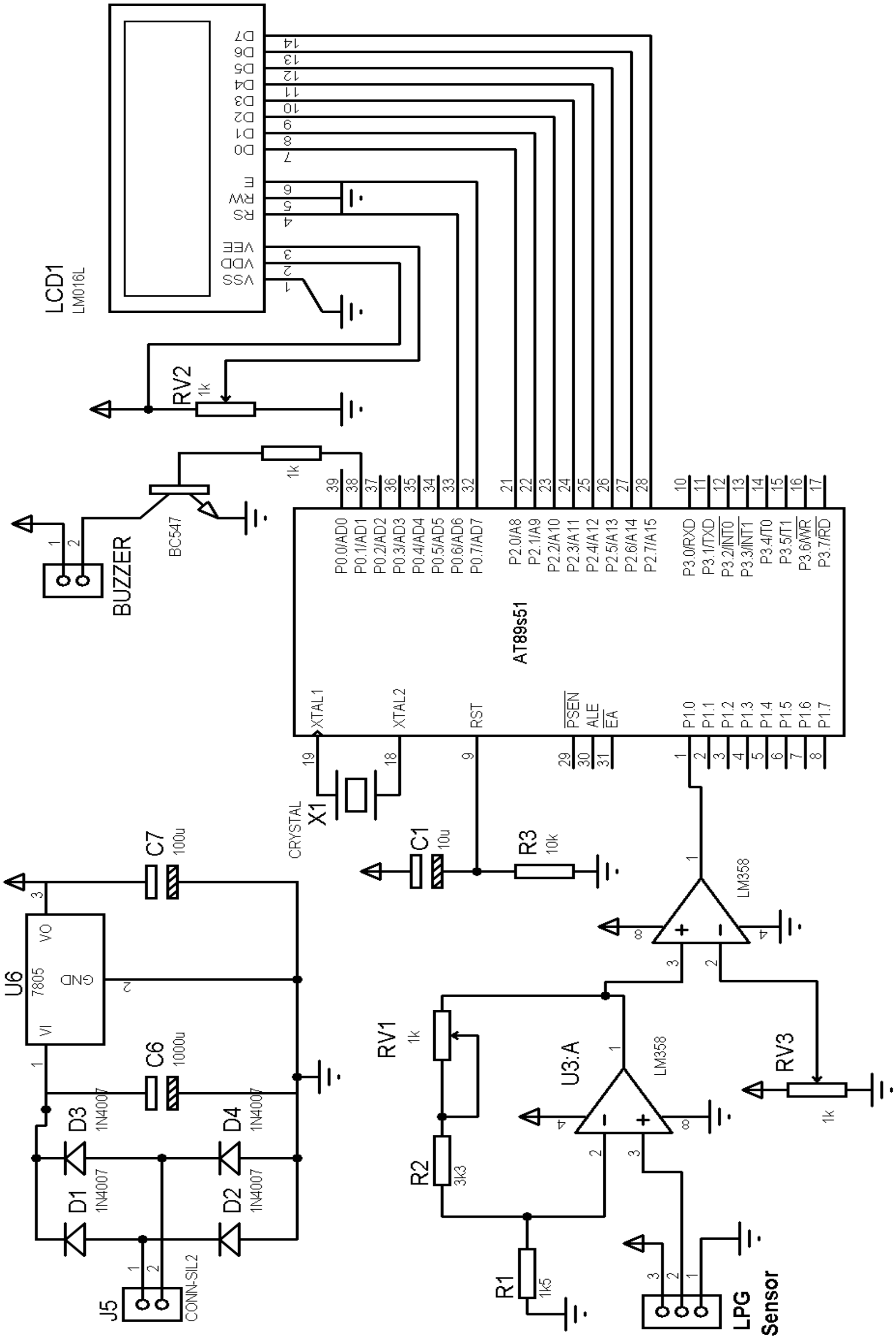
Generally, step-up transformers are located at power generation plants, stepping up the voltage flowing from the power plant to long-distance distribution networks. Step-down transformers, on the other hand, decrease the voltage of power streams received at the local distribution level. The long-distance stream is first stepped down to a level acceptable for local distribution, and then stepped down again at each consumer node (residential homes and offices).


STEP DOWN TRANSFORMER
Input Voltage
Soft Iron Core
Output Voltage
Primary Alternating Voltage (Vp)
Np
Ns
Secondary Alternating Voltage (Vs)
Number of Primary Winding(Np)>Number of Secondary Winding(Ns)
Primary Voltage(Vp)>Secondary Voltage(Vs)
Calculating Voltage (Vp/Vs)=(Np/Ns)

Transformer Operation

Transformers work on the principle of mutual induction. A changing magnetic field in one loop of wire induces an electromotive force (EMF) in an adjacent loop of wire, inductively coupled to the first. In very basic terms, a transformer consists of two coils of wire with high mutual inductance. These coils are electrically separated while they share a common magnetic circuit.

CIRCUIT DIAGRAM



EAGLE PCB DESIGN SOFTWARE

The name EAGLE is a short form, which stands for Easily Applicable Graphical Layout Editor. This software offers user friendly and powerful solutions for PCB design, including Schematic Capture, Board Layout and Auto router. This software is developed by Cad soft. EAGLE is popular among hobbyists because of its freeware license and rich availability of component libraries on the web. EAGLE has following 2 sections:

Schematic capture: EAGLE contains a schematic editor, for designing circuit diagrams. Parts can be placed on many sheets and connected together through ports.

PCB layout: The PCB layout editor allows back annotation to the schematic and auto-routing to automatically connect traces based on the connections defined in the schematic.

How to use EAGLE Software

1) Create new project

Start up the Eagle control panel, and right-click on "projects" to create a new project. You'll get to name it whatever you want

2) Create new schematic in the project

Once you have created the new project, it will be "opened" automatically (which doesn't do much other than tell EAGLE that "this is the current project".) Right click on the new project and follow the popup menus to create a new schematic.

3) Find and place ("add") components

Components are added to a schematic from the ADD dialog, which you get to by clicking the ADD button over on the GUI menu.

4) Neaten things up

We can make things a bit neater by moving some of the components, junctions, and wires around just a BIT using the Move button. When you select an object after clicking the move button, the object gets attached to your mouse pointer and behaves like when you were adding objects with respect to the other mouse buttons.

5) Re-zoom the drawing

Since we have all the components added, we can use the zoom-to-fit button to fit things better in our window.

6) Wiring components

Now it's time to make wire between components, find "Wire" to start. When you finish making wire between two components, hit "Escape" in the keyboard to move to the next one

7) Do Rule Check!

The button shown does an electrical rule check. It will check whether the pins designated outputs are connected to inputs, whether there are obvious missing junctions, and stuff like that.

8) Fix Errors and Warnings

When we run it on our schematics so far, we get a warning that the junction are missing. We could fix that with the name command, or just leave it as is.

9) Apply component values

When we plopped down the components during the "add" phase, we didn't assign specific values to any of them. Some of the components have inherent values that don't need to change. But the resistors, capacitors, and diodes should all have their values filled in appropriately.

Values are assigned using the "Value" button. After selecting the button, click on each component near its origin (little "+" sign), and you should be presented with an opportunity to change the value.

10) Transform the schematic into board

After finishing and saving your schematic, now we can move to the next step which is to move to board drawing by using "Board" as shown in the picture.

11) Rearrange your components

Use "move" and the right-button of the mouse to rearrange your circuit on the board

12) Auto router

After rearranging your circuit, just go to "Tools / Auto" Then hit "OK"

The job is done. Your circuit is ready, save your work as usual and print it if you want

KEIL MICROCONTROLLER PROGRAMMING SOFTWARE

Keil has compiler designed specifically for the 8051 microcontroller. Keil provides a broad range of development tools like IDE (Integrated Development environment), Project Manager, Simulator, Debugger, and C Cross Compiler.Compilers are programs used to convert a High Level Language source code (written in assembly language or C language) into its object code.Then a linker is used to create an absolute object module suitable for your circuit.

8051 project development cycle: - these are the steps to develop 8051 project using keil

1. Create source files in C or assembly.
2. Compile or assemble source files.
3. Correct errors in source files.
4. Link object files from compiler and assembler.
5. Test linked application.

KEIL Software Programming Procedure

Step-1) Install Keil MicroVision in your PC, Then after Click on that “Keil UVision” icon. After opening the window go to toolbar and select Project Tab then close previous project.

Step-2) Next select New Project from Project Tab.it will open “Create New Project” window

Step-3)Select the path where you want to save project and edit project name.

Step-4) Next it opens “Select Device for Target” window, it shows list of companies and here you can select the device manufacturer company.

Step-5) For an example, for your project purpose you can select the chip as 89c51/52 from Atmel Group. Next Click OK Button, it appears empty window here you can observe left side a small window i.e., “Project Window”. Next create a new file.

Step-6) From the Main tool bar Menu select “File” Tab and go to New, then it will open a window, there you can edit the program.

Step-7) You can edit the program of either Assembly language or C language

Step-8) After editing the program save the file with extension as “.c” or “.asm”, if you write a program in Assembly Language save as “.asm” or if you write a program in C Language save as “.c” in the selected path. Take an example and save the file as “test.c”.

Step-9) Then after saving the file, compile the program. For compilation go to project window select “source group” and right click on that and go to “Add files to Group”.

Step-9) Here it will ask which file has to Add. E.g. you can add “test.c” as you saved before.

Step-9) After adding the file, again go to Project Window and right click on your “c file” then select “Build target” for compilation. If there is any “Errors or Warnings” in your program you can check in “Output Window” that is shown bottom of the Keil window.

Step-10) Here in this step you can observe the output window for “errors and warnings”.

Step-11) If you make any mistake in your program you can check in this slide for which error and where the error is by clicking on that error.

Step-12) After compilation then next go to Debug Session. In Tool Bar menu go to “Debug” tab and select “Start/Stop Debug Session”.

Step-13) Here a simple program for “Leds Blinking”. LEDS are connected to PORT-1. You can observe the output in that port.

Step-14) To see the Ports and other Peripheral Features go to main toolbar menu and select peripherals.

Step-15) In this slide see the selected port i.e., PORT-1.

Step-16) Start to trace the program in sequence manner i.e., step by step execution and observe the output in port window.

Step-17) After completion of Debug Session Create an Hex file for Burning the Processor. Here to create an Hex file go to project window and right click on Target next select “Option for Target”.

Step-18) It appears one window; here in “target tab” modify the crystal frequency as you connected to your microcontroller.

Step-19) Next go to “Output’ tab. In that Output tab click on “Create HEX File” and then click OK. Finally Once again compile your program. The Created Hex File will appear in your path folder.

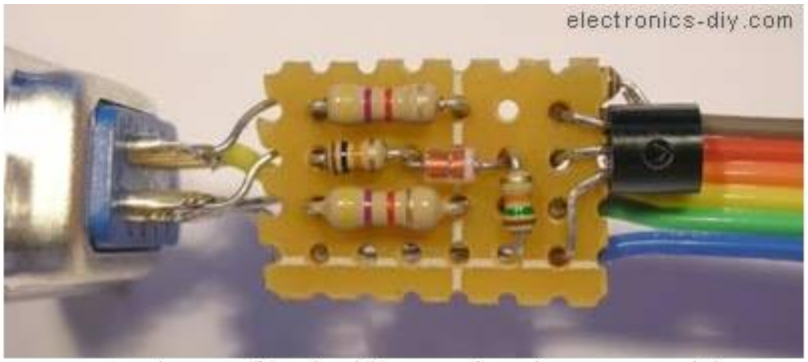
MICROCONTROLLER PROGRAMMER

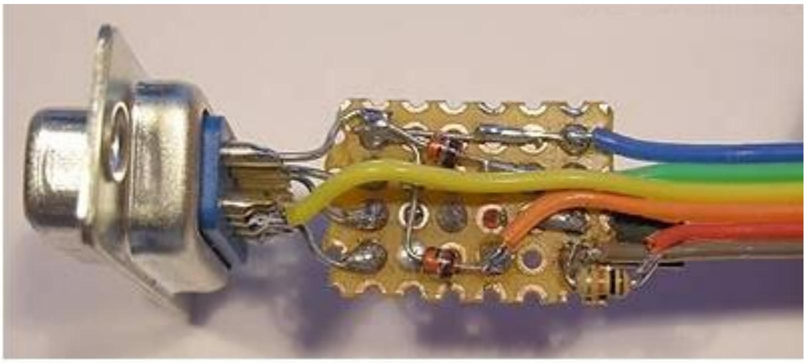
This Microcontroller Programmer we are using for our project. This simple Microcontroller Programmer will allow you to painlessly transfer hex programs to most ATMEL Microcontroller microcontrollers without sacrificing your budget and time. It is more reliable than most other simple Microcontroller programmers available out there and can be built in very short amount of time.

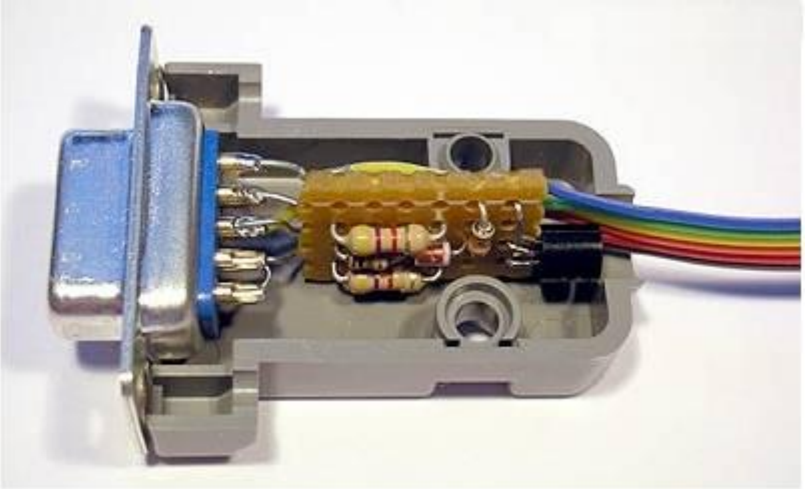
Microcontroller programmer consists of in-circuit serial programmer (dongle) and small PCB with a DIP socket where you can fit your microcontroller and have it quickly programmed.

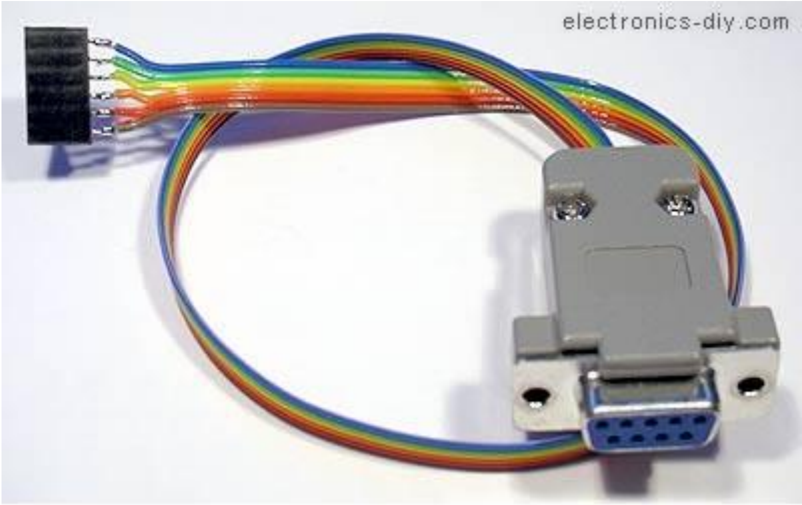
You may also use this programmer as a standalone in-circuit serial programmer that can be used to conveniently program Microcontroller microcontrollers without removing them from the target circuit.

Entire Microcontroller programmer has been built with using common parts and fits in the case of the serial connector. The socket PCB has been created to fit a 40-DIP Microcontroller 89s51 microcontroller, but you can build a socket PCB for any other Microcontroller out there. This Microcontroller programmer is compatible with popular PonyProg software that shows you a status bar of the programming progress.








electronics-diy.com

MICROCONTROLLER PROGRAM -- SOURCE CODE FORMAT

A typical line in assembly language program might be as follows:

LOOP: MOV R0, #80; initialize counter

This line will be assembled into a single instruction (in this case 11 0000 1000 0000 in binary, or 3080); the assembly language and the machine code correspond to each other.

It has four parts: - (not all are present in every line)

1) Label 2) Mnemonic 3) Operand 4) Comment

1) Label: The first part (LOOP in this example) is a label; this is a word, invented by programmer, it identifies this point in the program. It will be set equal to the value of the address where this instruction is stored. E.g. if later in the program there is a statement JMP LOOP, the assembler program replaces the label LOOP with the actual value of LOOP, it is the address at which this instruction is stored. (For the assembler to recognize this as a label, the label must have a colon ":" follows the label) So if the address at which the instruction is stored is 4F, LOOP takes on the value 4F. If LOOP is used later in the program, the assembler will give it the value 4F.

2) Mnemonic: The second part is the mnemonic .This corresponds to a particular kind of instruction (opcode sent by the Dispatch Unit). The intention is that the word chosen (by the manufacturers) for the mnemonic is easy to remember, and indicates what the instruction does. In this case, the instruction moves a literal value (one byte) into the register 0, hence MOV.B r0, #80

3) Operand: The third part of the line is an operand (there may be two); in this case the operand is the value 80 (in Hex) and the register r0.

4) Comment: The last part of the line is a comment. This does not affect the actual instruction at all; it is not part of the instruction, and is not assembled; instead it helps the programmer to remember what this part of the program does. The comment is preceded by a semi-colon.

When program is written an in assembly language, it actually consists of lots of ASCII characters; this would be stored in a file and called "source code".This then forms the input to a program called an "assembler" (MASM) which can translate the "source code" into machine code. Or object code. When the assembler has done its work, this line of source code will have been translated into a binary pattern, associated with a particular address in the program memory.

Assembly language directives

It is one of the important concepts. These statements are part of the source code which the assembler uses, but it does not correspond to actual instructions. E.g. ORG, EQU, END

ORG: The directive ORG does not get translated into any code; what it does is to set the address at which the next instruction will be stored.

Example:

```
ORG 0000H
GOTO 10
```

The instruction GOTO 10 is translated into the machine code 10 1000 0001 0000

But it will be placed at the address 0. ORG 0 does not get translated into any code; but it affects where the code for GOTO 10 is placed.

EQU: The directive EQU stands for "equals"; it allows you to give a numerical value to a label. In this example, the assembler would always replace START with 1016. Hence you could write:

Example:

```
START EQU 10
ORG START
```

The assembler would translate START as 10, and load the code at $10, as before. The use of labels often makes a program easier to understand and to modify.

For example, it is easier to remember the names of the special function registers than their addresses.

Example

```
P1.2 EQU IR_Sensor
STATUS EQU 3; status register
Port2 EQU LCD; Port A
```

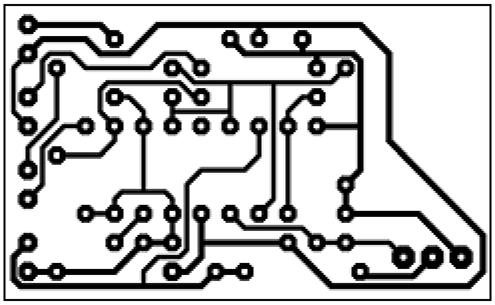
It is also easier to remember names than addresses of general purpose file registers. Suppose you want to use one of them as a counter; then you could define

END: This is not the end of the program that the 8051 runs

Such programs are usually endless loops; once the 8051 is switched on; it runs in the loop forever. END is an instruction or directive to assembler, to say "This is the end of my program; you can finish now!"

PCB LAYOUT

A printed circuit board, or PCB, is used to mechanically support and electrically connect electronic components using conductive pathways, tracks or traces etched from copper sheets laminated onto a non-conductive substrate. It is also referred to as printed wiring board (PWB) or etched wiring board. A PCB populated with electronic components is a printed circuit assembly (PCA), also known as a printed circuit board assembly (PCBA).

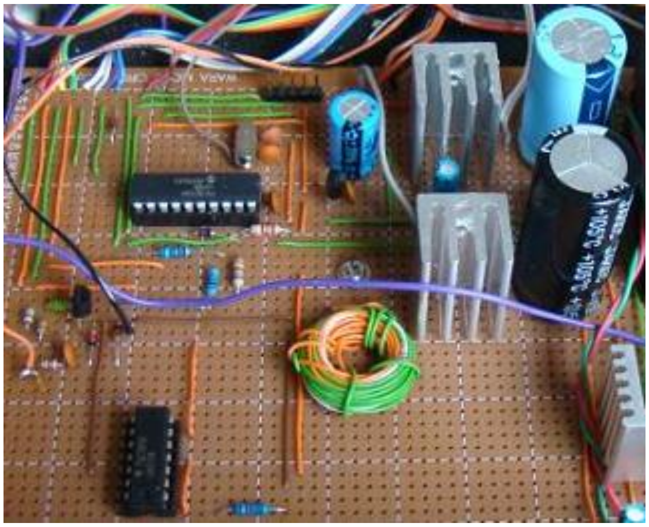


Printed Circuit Boards play a vital role here in determining the overall performance of the electronic equipment. A good PCB design ensures that the noise introduced as a result of component placement and track layout is held within limits while still providing components years of assembly maintenance and performance reliability.

Where and Why are PCB's used?

Printed circuits boards are used to route electric signals through copper track which are firmly bonded to an insulating base.

Advantages of PCB over common wiring are:



1. PCB's are necessary for connecting a large number of electronic components in a very small area with minimum parasitic effects.

2. PCB's are simulated with mass production with less chance of writing error.

3. Small components are easily mounted.

4. Servicing in simplified.

Base Materials used for PCB

The base materials used for PCB's are glass epoxy, epoxy paper, polyester etc. Copper foil used for copper clad is manufactured by the process of electronic deposition.

Preparation of Single Sided PCB

In single sided PCB conductor tracks are present on one side of copper clad board. So crossing of conductors is not allowed. It is mechanically & chemically cleansed. The photo resist is an organic solution which when exposed to light of proper wavelength, changes their solubility in developer but after exposure to light is not soluble.

Laminate coating of photo resist is done by: (I) Spray coating (ii) Dip coating (iii) Roller coating.

The coated copper clad and laminated film negative is kept in intimate contact with each other. The assembly is exposed to UV light and is rinsed in the developer tank. Proper developer has to be used for a particular photo resist and then the PCB is dyed in a tray. The dye reveals the flux to be used for a particular photo resist. Then the PCB is dyed in a tray.

FABRICATION

The required circuit is designed and the layout of the circuit is done on the component side as well as the copper clad side. Spaces are provided for holes to insert the respective components. Etch resistant ink coatings are given on the interconnecting marks.

ETCHING

The copper clad PCB is etched with ferrous chloride solution containing a small amount of Hydro Chloric Acid for increasing activeness of Ferric Chloride in etching. Wherever the varnish coating is there the copper remains. Then it is washed with water and Oxalic Acid.

The vast majority of printed circuit boards are made by bonding a layer of copper over the entire substrate, sometimes on both sides, (creating a "blank PCB") then removing unwanted copper after applying a temporary mask (e.g. by etching), leaving only the desired copper traces. A few PCBs are made by adding traces to the bare substrate (or a substrate with a very thin layer of copper) usually by a complex process of multiple electroplating steps.

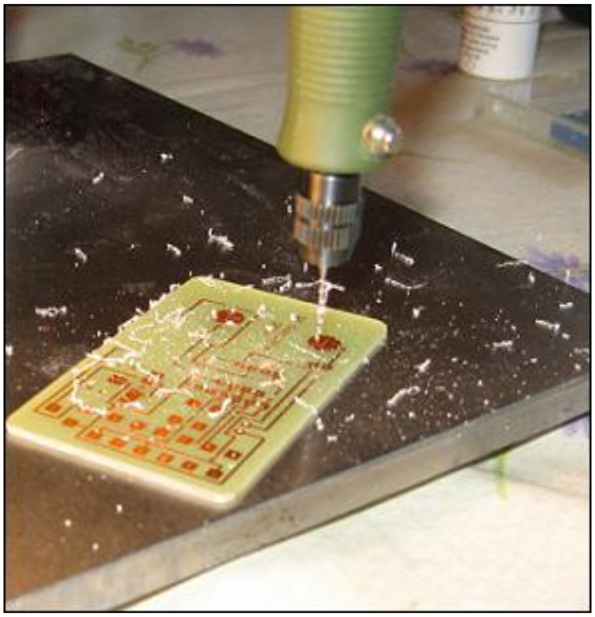
There are three common "subtractive" methods (methods that remove copper) used for the production of printed circuit boards:

1. Silk screen printing uses etch-resistant inks to protect the copper foil. Subsequent etching removes the unwanted copper. Alternatively, the ink may be conductive, printed on a blank (non-conductive) board. The latter technique is also used in the manufacture of hybrid circuits.

2. Photoengraving uses a photomask and chemical etching to remove the copper foil from the substrate. The photomask is usually prepared with a photoplotter from data produced by a technician using CAM, or computer-aided manufacturing software. Laser-printed transparencies are typically employed for phototools; however, direct laser imaging techniques are being employed to replace phototools for high-resolution requirements.

3. PCB milling uses a two or three-axis mechanical milling system to mill away the copper foil from the substrate. A PCB milling machine (referred to as a 'PCB Prototyper') operates in a similar way to a plotter, receiving commands from the host software that control the position of the milling head in the x, y, and (if relevant) z axis. Data to drive the Prototyper is extracted from files generated in PCB design software and stored in HPGL or Gerber file format.

DRILLING



Holes through a PCB are typically drilled with tiny drill bits made of solid tungsten carbide. The drilling is performed by automated drilling machines with placement controlled by a drill tape or drill file. These computer-generated files are also called numerically controlled drill (NCD) files or "Excellon files".

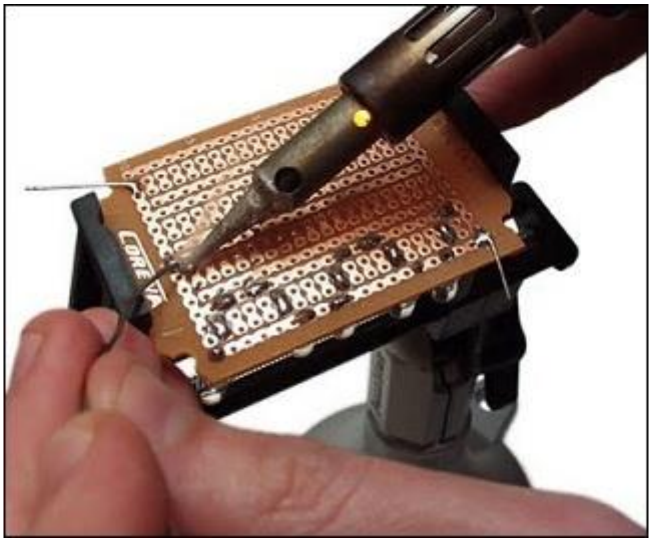
The drill file describes the location and size of each drilled hole. These holes are often filled with annular rings (hollow rivets) to create vias. Vias allow the electrical and thermal connection of conductors on opposite sides of the PCB.

Plating and Coating

PCBs are plated with Solder, Tin, or Gold over Nickel as a resist for etching (removal) away the (unneeded after plating) underlying copper. Matte solder is usually fused to provide a better bonding surface or stripped to bare copper.

Treatments, such as benzimidazolethiol, prevent surface oxidation of bare copper. The places to which components will be mounted are typically plated, because untreated bare copper oxidizes quickly, and therefore is not readily solderable. Traditionally, any exposed copper was coated with solder by hot air solder levelling (HASL). This solder was a tin-lead alloy, however new solder compounds are now used to achieve compliance with the RoHS directive in the EU and US, which restricts the use of lead. One of these lead-free compounds is SN100CL, made up of 99.3% tin, 0.7% copper, 0.05% nickel, and a nominal of 60ppm germanium.

SOLDERING



Soldering is the process of joining of two metals using an alloy solder consisting of Tin and Lead (Sn-Pb). Tin determines the melting whereas the Lead is used to reduce the cost. After the PCB fabrication is done, the various components are arranged at proper locations on the PCB and then the soldering is done. All liquids consist of particles which attract each other. The surface is always trying to shrink and this is because of surface tension. The principle behind soldering is that when liquid particles are brought in contact with the walls of the solid surface, it may happen that the solid attracts the liquid surface. This property is called adhesive property. Care must be taken that the melting point of solder is below that of the metal so that its surface is melted without melting without the metal.

NEED FOR FLUX

During the soldering process the flux acts as a medium for improving the degree of melting. The basic functions of flux are mentioned below:

1. Removes oxide from the surface.
2. It transfers heat from source to the joining & provide liquid cover including air gap.
3. Removal of residue after the completion of the soldering operation.

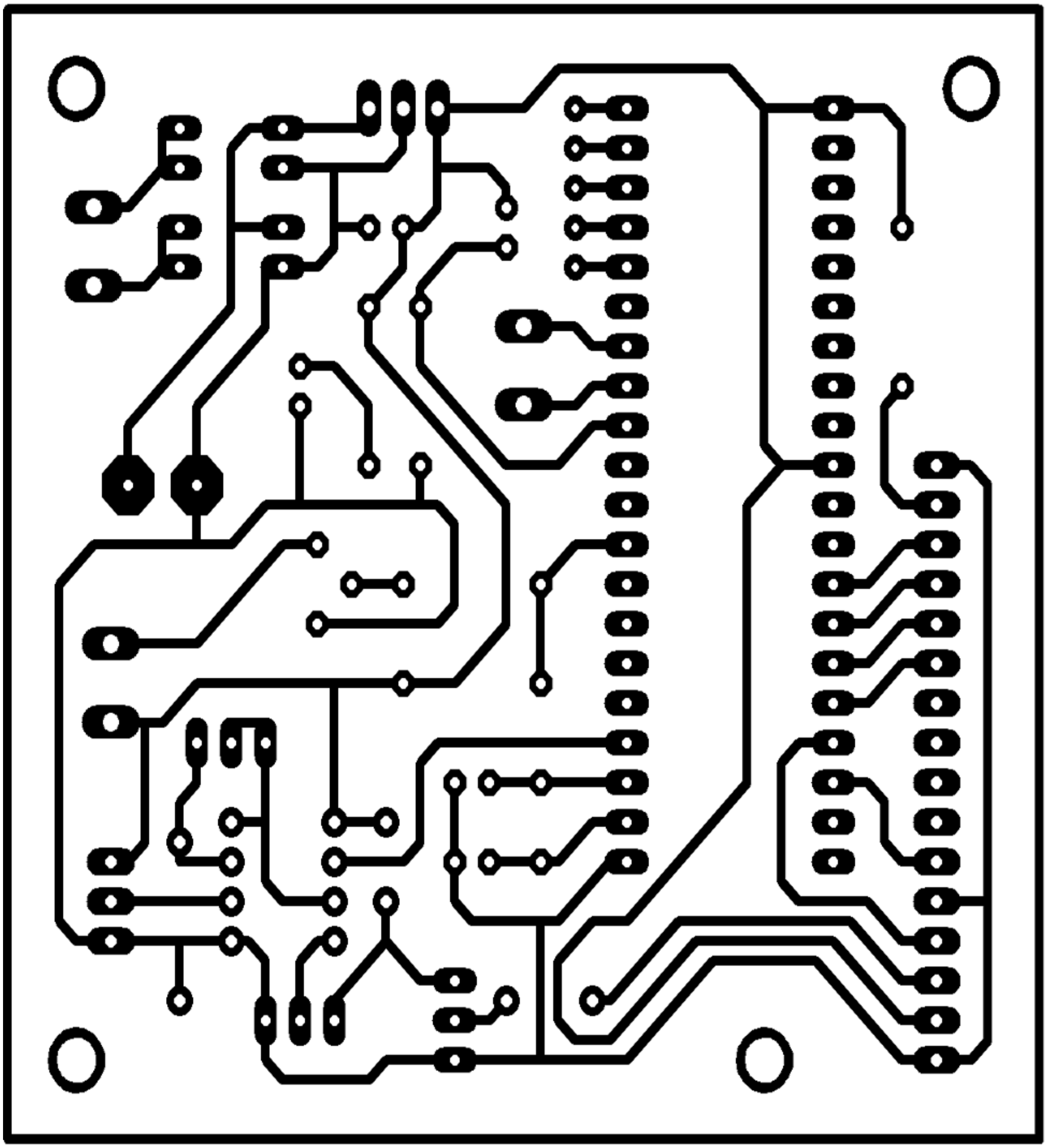
PCB LAYOUT

The layout can be done either by hand or by using PCB designing software mentioned below:

- EAGLE by Cadsoft
- DipTrace by Novarm
- FreePCB by Allan Wright (open-source Win2K/XP)
- FreeRouting by Alfons Wirtz (Java, Auto router with free angle support)
- Cadstar by Zuken
- gEDA, open-source PCB software project
- OrCAD by Cadence
- Allegro by Cadence
- TARGET 3001!
- Kicad, open-source suite
- PADS by Mentor Graphics
- PCB123 Design by Sunstone Circuits
- Proteus

Here we have used eagle software to design PCB layout

PCB Layout of our project



DEBUGGING & TESTING

We have used simulator kit of 89s51 for hardware testing and debugging. For the software testing and debugging we have used KEIL software.

We divided our debugging and testing in different modules, which are as follows:

a) First of all we developed assembly programming for KEIL software.
b) Then we developed programming of 89s51 to send the code of LCD, Serial communication.
c) Then we assign different ports for corresponding peripherals.
d) After connecting different devices to their corresponding ports weburn the 89s51 with the help of Flash-magic software.
e) Implementation of hardware is carried out by affixing the components on the PCB and check the continuity of PCB with the help of multi meter.
f) Before implementing the IC on the respective place we use IC tester to ensure that the IC is working properly or not.
g) The burnt 89s51 is placed in the circuit board.
h) Then we apply the power supply to our system to check whether the given code is working properly or not, message displayed on LCD.
i) We also used Rxd and Txd pin of 89s51 for serial communication. With the help of above given pin we connect max232 for making data in the format of pc compatibility.
j) After completion of all the connections and all the steps, we open HyperTerminal and check whether the message is displayed on HyperTerminal or not, if not, we check the code burned in 89s51 or check the connection between the systems.

1) Problem crop up

- Microcontroller at AT89C51 was not providing the results or was giving errors
- The pot used in the LCD circuit for adjusting the brightness/contract of LCD display was not giving the appropriate results.

2) How problems are rectified

- The microcontroller was replaced by the AT89S51 as the AT89C51 does not have In System Programmer.

- Potentiometer was adjusted to a certain level to get the required contrast on LCD display.

3) Testing procedure

Once the hardware has been assembled, it is necessary to verify that, the design is correct & the prototype is built to the design drawing. This verification of the design is done by writing several small programs, beginning with the most basic program & building on the demonstrated success of each.

a) Crystal test: The initial test is to ensure that, both, the crystal & the reset circuit are working. The micro-controller is inserted in the circuit, & the ALE pulse is checked, with an oscilloscope, to verify that the ALE frequency is 1/6th of the crystal frequency. Next, the reset button is pushed & all ports are checked to see that, they are in the high input state.

b) PCB testing: The PCB was tested, by tracing the tracks from the net list & the artwork of the PCB. The errors in the artwork were eliminated while testing & after that it was given for PCB manufacturing.

The PCB was tested using the DMM & the continuity of the tracks was tested using the DMM, in the diode mode. The positive terminal was connected to the terminals of the other IC's to show the negligible resistance, if the track is continuous.

c) VISUAL TESTING:

- Polarities of all the components like capacitors, connectors etc are checked.
- It is seen that all the IC sockets are soldered properly.

d) MULTIMETER TESTING:

- All the IC sockets and power supply are soldered and continuity is checked
- Also VCC and GND voltage are checked.
- Voltages at all the pins of the microcontroller are checked with respect to ground.
- Values of all possible components are checked on multimeter.

e) Software testing:

While designing the software for our project, we considered the following points:

Firstly in the software, all the IC's were initialized. After all the subroutines for each module were executed, they were displaying the proper results.

Component List

	ICs	
Sr no	**Component Name**	**Quantity**
1	89S51	1
2	IC Base 40 Pin	1
3	LM358	1
4	IC Base 8 pin	1
	Capacitors	
Sr no	**Component Name**	**Quantity**
1	102	2
2	103	2
3	1000microF	1
4	100microF	2
5	10microF	1
6	33pF	2
7	4.7microF	2
	Miscellaneous	
Sr no	**Component Name**	**Quantity**
1	Crystal 11.0592 MHz	1
2	1N4007	5
3	7805	1
4	7812	1
5		
6	BC547	1
7		
8	LCD	1

9	LED	5
10	PBT2	1
11	PBT3	1
12	Relay	1
13	Transformer	1
14		
	Resistor	
Sr no	**Component Name**	**Quantity**
1	100E	6
2	10k	4
3	15k	2
4	1k	2
5	27k	2
6	2k2	4
7	3k3	1
8	4k7	3
9	POT1 5k	1
10	POT2 10k	2
11	PULLUP	1
	Connector	
Sr no	**Component Name**	**Quantity**
1	02 Pin	4
2	03 Pin	2
3	16 Pin	1
4	ISP 06Pin	1

CONCLUSION

With the knowledge of new techniques in ‘Electronics’ we are able to make our life more comfortable. One such application of electronics is used in “LPG leakage detector with buzzer indication using Microcontroller” The approach we followed and which is explained in this project report is novel and has achieved the target of "LPG leakage detector with buzzer indication using Microcontroller" satisfying user needs and requirements.

The development of this project has shown how much hard work goes into the creation of a system.“LPG leakage detector with buzzer indication using Microcontroller” was a project based on microcontroller, due to which hardware requirement is reduced. Embarking of this project has helped us in developing a team spirit, patience and time management necessary for today's technical professionals.

Hence we can conclude that the required goals and objectives of our project have been achieved.

This project has built in us confidence that any problem can be solved with sheer determination, hard work and optimism. We feel that our product serves something good to this world and we like to present it before this prosperous world. By doing this project, we were better able to understand the various facets of doing an embedded system project which is emerging as one of the most 'in demand' technologies right now.

Microcontroller program

```
LCDdatabus              equ     0A0H
LCDrs                   equ     0A3H
LCDen                   equ     0A2H

Buzzer                   equ     0B2H

LPG_Sensor                    equ       0B7H


LCDreg                  equ     30h
LCDtempreg              equ     34H
Reg_LCD_swap1           EQU     35h
Reg_LCD_swap2           EQU     36h

                    org 0000h
                    call LCDinit
                    clr Buzzer
                    mov dptr,#msgwelcome
                    call LCDdisp

start:
                    jb LPG_Sensor,start
                    setb Buzzer
                    mov dptr,#MsgTempdetect
                    call LCDdisp
                    call delay2sec
                    clr Buzzer
                    jmp start
;=========================================
;subroutines start from here
;=======================================
LCDinit:
                call delayhalf
                mov LCDtempreg,#02h
                call LCDcmd
                mov LCDtempreg,#28h
                call LCDcmd
                mov LCDtempreg,#0Ch
                call LCDcmd
                mov LCDtempreg,#06h
                call LCDcmd
                mov LCDtempreg,#01h
                call LCDcmd
                ret
;====================================
LCDcmd:
                mov Reg_LCD_swap1,LCDtempreg

                mov Reg_LCD_swap2,Reg_LCD_swap1

                mov a,Reg_LCD_swap2
                anl a,#0F0H
                mov LCDdatabus,a

                clr LCDrs
                setb LCDen
                nop
                nop
                clr LCDen
                call LCDdelay

                mov a,Reg_LCD_swap2

                anl a,#0F0H
                mov LCDdatabus,a

                clr LCDrs
                setb LCDen
                nop
                nop
                clr LCDen
                call LCDdelay

                ret
;====================================
LCDdata:
                mov Reg_LCD_swap1,LCDtempreg

                mov Reg_LCD_swap2,Reg_LCD_swap1

                mov a,Reg_LCD_swap2
                anl a,#0F0H
                mov LCDdatabus,a

                setb LCDrs
                setb LCDen
                nop
                nop
                clr LCDen
                call LCDdelay
                mov a,Reg_LCD_swap2

                anl a,#0FH
                mov LCDdatabus,a


                setb LCDrs
                setb LCDen
                nop
                nop
                clr LCDen
                call LCDdelay

                ret
;========================================
LCDdelay:        mov 31H,#10             ;LCD
LCDdelay1:       mov 32H,#250
                djnz 31H,$
                djnz 32H,LCDdelay1
                ret
;========================================
LCDdisp:        mov LCDtempreg,#01h
                call LCDcmd
                mov LCDreg,#00h
LCDdisp2:        mov a,LCDreg
                movc a,@a+dptr
                cjne a,#'@',LCDdisp1
                mov LCDtempreg,#0c0h
                call LCDcmd
                inc LCDreg
                jmp LCDdisp2
LCDdisp1:        cjne a,#'$',LCDdisp3
                ret
LCDdisp3:        mov LCDtempreg,a
                call LCDdata
                inc LCDreg
                jmp LCDdisp2
;====================================
delayhalf:       mov 31H,#5
delhalf2:        mov 32H,#200
delhalf1:        mov 33H,#250
                djnz 31H,$
                djnz 32H,delhalf1
                djnz 33H,delhalf2
                ret
;========================================
delay1sec:        mov 31H,#10
del1sec2:         mov 32H,#200
del1sec1:         mov 33H,#250
                 djnz 31H,$
                 djnz 32H,del1sec1
                 djnz 33H,del1sec2
                 ret
;========================================
delay2sec:       call delay1sec
                 call delay1sec
                 ret
;========================================
; Messages to be displayed on LCD
;======================================
msgwelcome:              DB "* LPG *@*** DETECTOR ***$"
MsgTempdetect:           DB "LPG@INCREASED$"
;=========================================
;     Program ENDS here
;=========================================
                              END
```