HBeonLabs

-
Off. No. 46, 1st Floor, Kadamba Complex
Gamma-I, Greater Noida (India) - 201308

Contact us:

+91-120-4298000
+91-9212314779

info@hbeonlabs.com

training@hbeonlabs.com

www. hbeonlabs.com

TABLE OF CONTENTS

INTRODUCTION

Here is a circuit using MAX-232, which needs only a single power supply of 5V for level conversion. Fig. 1 shows the internal functional diagram of MAX232 IC. The communication over the short distance of 2 to 3 meters is established using infrared diodes as shown in Fig. 2. The range could be increased up to hundred meters, using a laser diode module in place of infrared LEDs.

The laser module used is easily available as laser pointer. It is to be used with its three battery cells removed and positive supply terminal soldered to the casing and 0V point to the contact inside the laser module.

Assemble the two prototypes on PCBs or breadboards and connect them to COM-1 (or COM-2) port of each PC. Point the laser beam of one module to fall on the photodiode of the module connected to the other PC, and vice versa.

Load PROCOMM or TELIX serial communication software and set the port parameters to 110 n 8 1 (here, 110 refers to the baud rate, n stands for parity-none, 8 represents bits per character, and 1 indicates number of stop bits) to establish the communication. File transfer is also possible. The software program for the purpose was written in ‘C’ language.

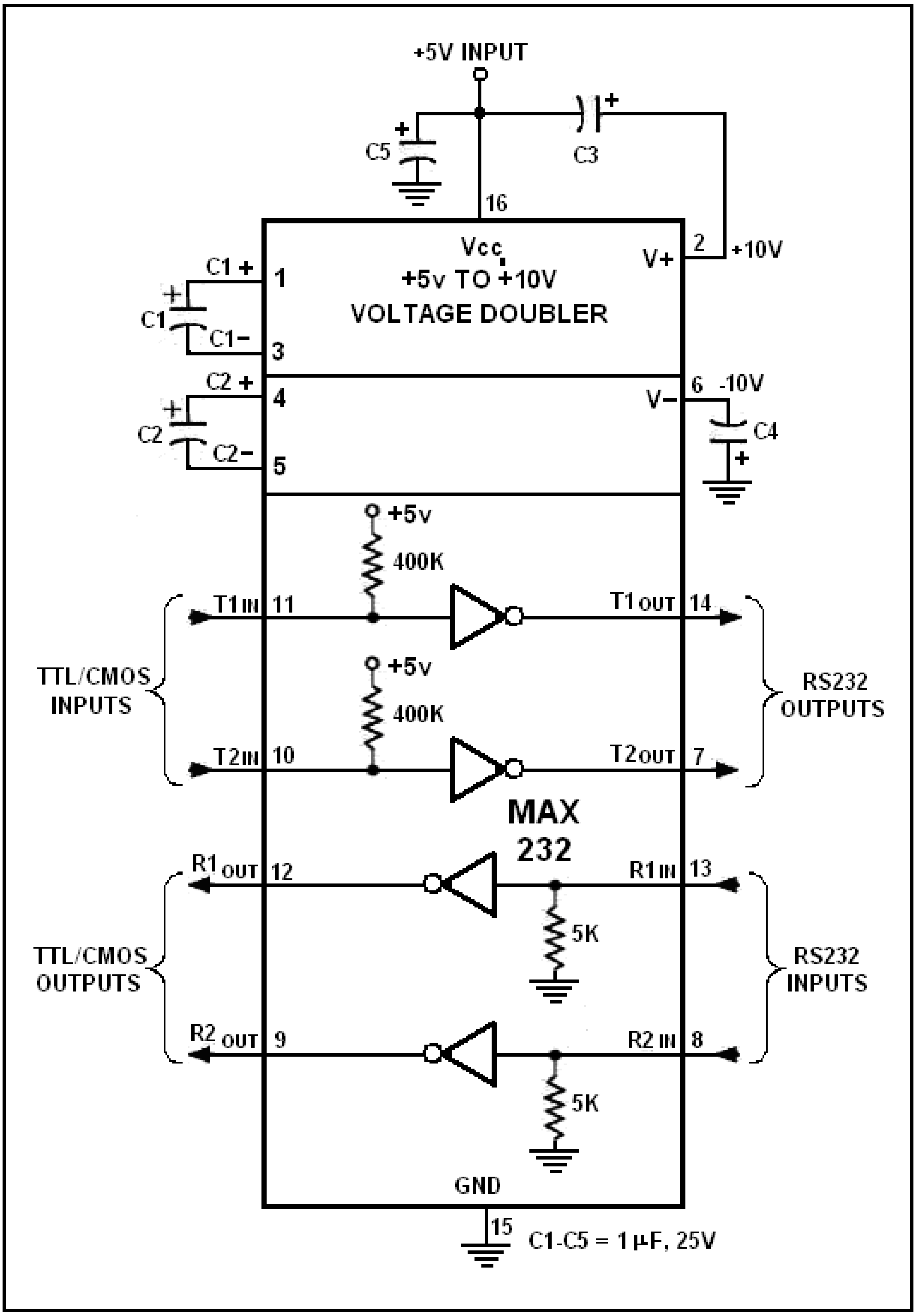


Fig.1 : Internal Functional Diagram of IC MAX232

MAX-232, which needs only a single power supply of 5V for level conversion. Fig. 1 shows the internal functional diagram of MAX232 IC. The communication over the short distance of 2 to 3 meters is established using infrared diodes as shown in Fig. 2. The range could be increased up to hundred meters, using a laser diode module in place of infrared LEDs.

The laser module used is easily available as laser pointer. It is to be used with its three battery cells removed and positive supply terminal soldered to the casing and 0V point to the contact inside the laser module.

Assemble the two prototypes on PCBs or breadboards and connect them to COM-1 (or COM-2) port of each PC. Point the laser beam of one module to fall on the photodiode of the module connected to the other PC, and vice versa.
Load PROCOMM or TELIX serial communication software and set the port parameters to 110 n 8 1 (here, 110 refers to the baud rate, n stands for parity-none, 8 represents bits per character, and 1 indicates number of stop bits) to establish the communication. File transfer is also possible. The software program for the purpose was written in ‘C’ language.

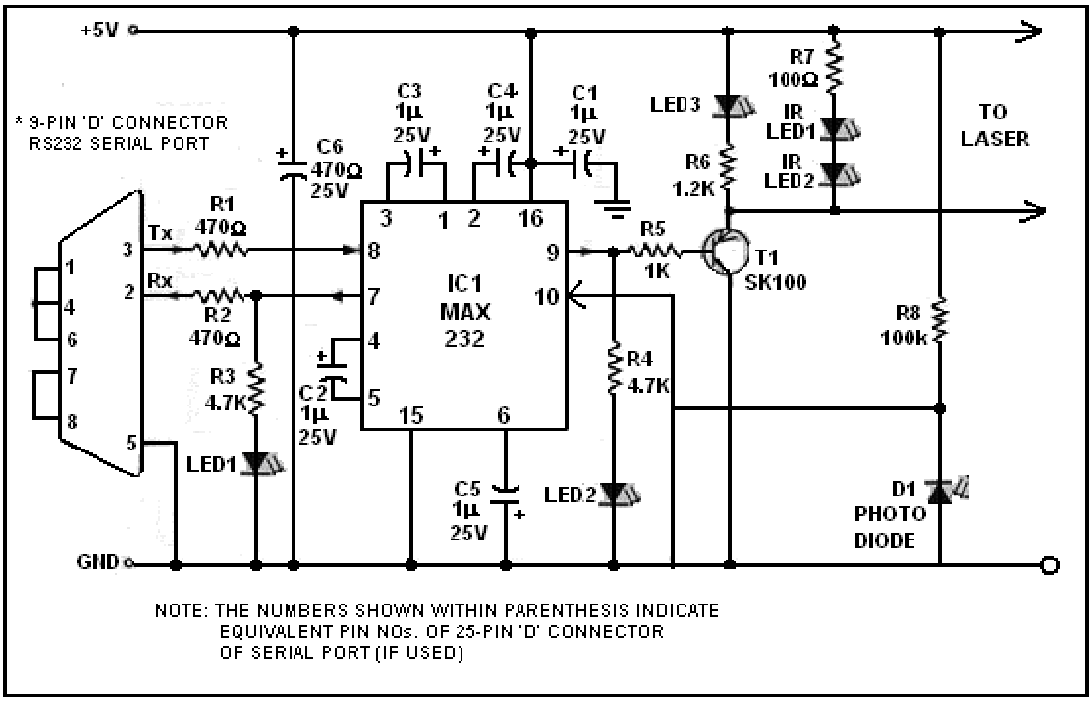


Fig.2: Communication between two PCs for a short range using IR diodes or longer distance using laser.

THEORY

COM PORT COMMUNICATION BASICS:

Data communication equipment (DCE) and data terminal equipment (DTE) use serial cables for communication. Examples of data communication equipment and data terminal equipment are modems and computers (or terminals), respectively.

To connect the two computers (DTEs) together, we use the null-modem configuration. The wiring diagram for the null-modem configuration is shown in Fig. 1.

COMPUTER A			COMPUTER B	
PIN	SIG.		PIN	SIG.
3	TD	→	RD	2
2	RD	←	TD	3
5	SG	↔	SG	5
4	DTR		DTR	4
6	DSR		DSR	6
1	CD		CD	1
7	RTS		RTS	7
8	CTS		CTS	8
9	RI		RI	9

Fig. 1 : Wiring for the null-modem configuration

In null-modem configuration, only three wires are required (TD, RD, and SG) to connect the two ports. Thus it is more cost-effective for long cable runs.

In this wiring configuration, a computer thinks that it is talking to a modem rather than another computer. Any data transmitted from one computer must be received by the other, so

TD pin of first PC is connected to RD pin of the other PC. The second PC also must have a similar set-up. Thus its TD pin is connected to RD pin of the former computer. Signal ground (SG) pins of both the PCs must be connected together; so that the ground forms the common reference for the RS-232 signals from/to each computer.

The data terminal ready (DTR) signal is looped back to data set ready (DSR) the configuration to which it is connected is that of a virtual modem in ready state. When the computer has to send data, it makes request-to-send (RTS) high. As this signal is looped back to the computer's own clear-to-send (CTS) signal pin, the computer immediately and carrier detected (CD) on both the computers. Hence, when DTR is active, DSR and CD immediately become active. The computer now thinks gets a reply that it's okay to send. We've not used the ring indicator (pin 9), as it is not required in this configuration.

COMPONENTS REQUIRED

SEMICONDUCTORS:

IC1, IC2 MAX232A +5V powered multichannel RS232 driver/ receiver

IC3 8051
IC4, IC5 MC74HC14A: HEX SCHMITT TRIGGER INVERTER

PUSH BUTTON 1-PUSH BUTTON 5
T1 ....BC558 pnp transistor

T2 ....BC548 npn transistor

LED1-LED2..........................Red LED

IRLED1, IRLED2 ....Infrared Light Emitting Diode

OSCILLATOR1……………..crystal oscillator

RESISTORS (All ¼ -watt, +5% carbon, unless stated otherwise):

R1, R2 ....47-ohm

R3, R4 ....4.7-kilo-ohm

R5, R9 ....1-kilo-ohm

R6..1.2-kilo-ohm

R7..10-ohm

R8..330-ohm

R10..2.2-kilo-ohm

R11..10-kilo-ohm

VR1.......................................4.7-kilo-ohm preset

CAPACITORS:

C1-C5....................................1□, 25V electrolytic

C6..470□, 25V electrolytic

C7, C8....................................0.01□ ceramic disk

MISCELLANEOUS

....9 pin ‘D’ connector (male/female)

THEORY OF COMPONENTS

PHOTO TRANSISTOR

A Germanium or silicon diode or transistor, which has a transparent encasing, can serve as a photodiode or transistor because the light photons can initiate conduction in the p-n- junction region. Early devices such as the OCP 71 were Ge-devices. Later, silicon types became available with lower leakage current and better light sensitivity. In a phototransistor, the base lead is not used; but, if a resistor is connected form base to emitter it reduced the light sensitivity. Darlington connected photo transistors (two transistors together in one case) such as the 2N5777 are very sensitive with a h_{FE} of 2.5K, a dark current of 100nA and a light current of 0.5-2.0mA for light flux density H=2mW/cm^2. The device is rated 200mW and voltage of 25V maximum.

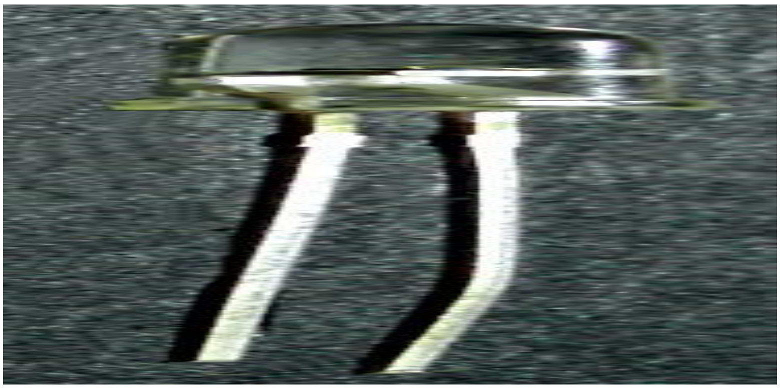
The switching speed of phototransistors far exceeds those of LDRs, made of CdS. The rise time for the 2N5777 is 75□s and fall time is 50□s. Maximum switching speed is 1KHz. Photo devices are useful in optical encoding, intrusion alarms, tape readers, level control, character recognition etc Nowadays packing containing an LED and a photodiode, called ‘opto-coupler’ is used for switching on power or control circuits. Because the light source (LED) and photodiode are physically kept separated (with 2mm) in the package, isolation upto 2500V can be had.

INFRARED LED'S

Gallium arsenide is a direct-gap semiconductor with an energy gap of 1.4eV at room temperature. A typical GaPs LED is made by solid-state impurity diffusion with zinc as the p-type impurity diffused into an n-type substate doped with tin, tellurium or silicon. The external efficiency at room temperature is typically 5 percent.

A GaAs diode can also be fabricated by liquid-phase epitaxy with silicon as both its n and p dopants. If a silicon atom replaces a Ga atom, it provides one additional electron, thus the resulting GaAs in as n-type. If a silicon atom replaces arsenic atoms, an electron is missing and the resulting GaAs is a p-type. In Si doped GaAs diode, the emission peak shifts down to 1.32eV. Since the emission is in infrared region, GaAs light sources are suitable for application such as the optical isolator. The high switching speed, with a recovery time between 2 and 10ns, makes them ideal for data transmission.

PHOTODIODE



A **photodiode** is a semiconductor diode that functions as a photodetector. Photodiodes are packaged with either a window or optical fibre connection, in order to let in the light to the sensitive part of the device. They may also be used without a window to detect vacuum UV or X-rays.

A **phototransistor** is in essence nothing more than a bipolar transistor that is encased in a transparent case so that light can reach the *base-collector junction*. The phototransistor works like a photodiode, but with a much higher sensitivity for light, because the electrons that are generated by photons in base-collector junction are injected into the base, this current is then amplified by the transistor operation. A phototransistor has a slower response time than a photodiode however.

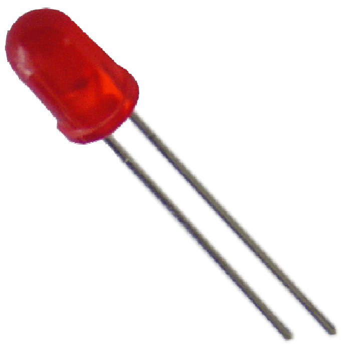
PRINCIPLE OF OPERATION

A **photodiode** is a p-n junction or p-i-n structure. When light with sufficient photon energy strikes a semiconductor, photons can be absorbed, resulting in generation of a mobile electron and electron hole. If the absorption occurs in the junction's depletion region, these carriers are swept from the junction by the *built-in field* of the depletion region, producing a *photocurrent*.

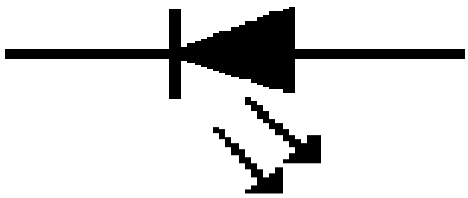
Photodiodes can be used in either zero bias or reverse bias. In zero bias, light falling on the diode causes a voltage to develop across the device, leading to a current in the forward bias direction. This is called the photovoltaic effect, and is the basis for solar cells — in fact a solar cell is just a large number of big, cheap photodiodes.

Diodes usually have extremely high resistance when reverse biased. This resistance is reduced when light of an appropriate frequency shines on the junction. Hence, a reverse biased diode can be used as a detector by monitoring the current running through it. Circuits based on this effect are more sensitive to light than ones based on the photovoltaic effect.

LIGHT EMITTING DIODE



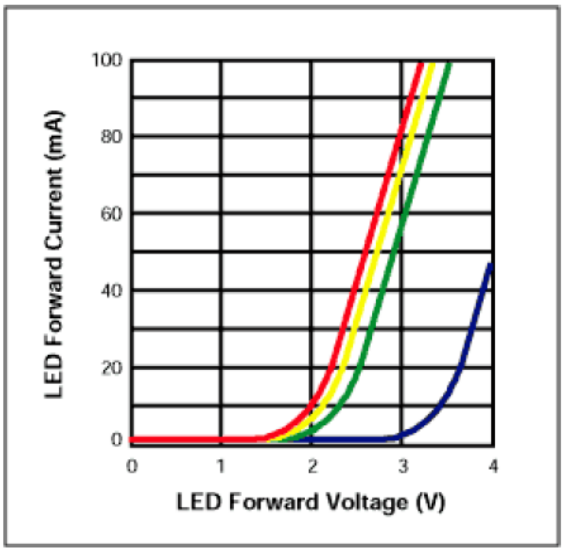
v



Light emitting diode (LED) is basically a P-N junction semiconductor diode particularly designed to emit visible light. There are infrared emitting LEDs which emit invisible light. The LEDs are now available in many colours red, green and yellow. A normal LED emits at 2.4V and consumes MA of current. The LEDs are made in the form of flat tiny P-N junction enclosed in a semi-spherical dome made up of clear coloured epoxy resin. The dome of a LED acts as a lens and diffuser of light. The diameter of the base is less than a quarter of an inch. The actual diameter varies somewhat with different makes. The common circuit symbols for the LED are shown in Fig. It is similar to the conventional rectifier diode symbol with two arrows pointing out. There are two leads- one for anode and the other for cathode.

CHARACTERISTICS OF LEDs

- Forward Voltage (VF) drop across LEDDiodes are current driven!
- Wavelength variationsCrystal and junction growth defects
- Brightness variationsCrystal defects resulting formation of phonons and non-radiation energy transfer
- Temperature Junction temperatureof the device affects each of the parameters above



TRANSISTOR

The name is transistor derived from ‘transfer resistors’ indicating a solid state Semiconductor device. In addition to conductor and insulators, there is a third class of material that exhibits proportion of both. Under some conditions, it acts as an insulator, and under other conditions it’s a conductor. This phenomenon is called Semi-conducting and allows a variable control over electron flow. So, the transistor is semi conductor device used in electronics for amplitude. Transistor has three terminals, one is the collector, one is the base and other is the emitter, (each lead must be connected in the circuit correctly and only then the transistor will function). Electrons are emitted via one terminal and collected on another terminal, while the third terminal acts as a control element. Each transistor has a number marked on its body. Every number has its own specifications.

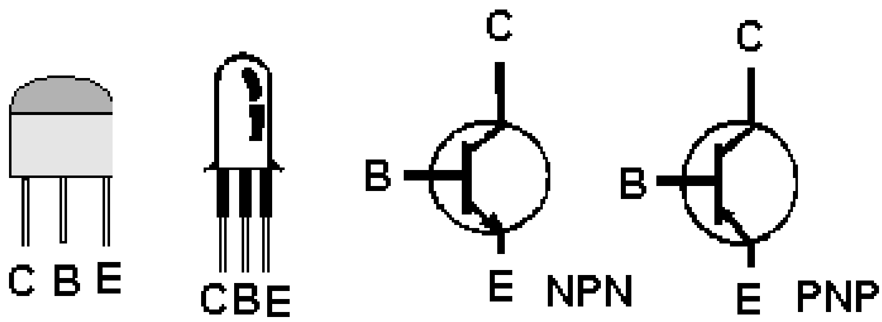
There are mainly two types of transistor (i) NPN & (ii) PNP

NPN Transistors:

When a positive voltage is applied to the base, the transistor begins to conduct by allowing current to flow through the collector to emitter circuit. The relatively small current flowing through the base circuit causes a much greater current to pass through the emitter / collector circuit. The phenomenon is called current gain and it is measure in beta.

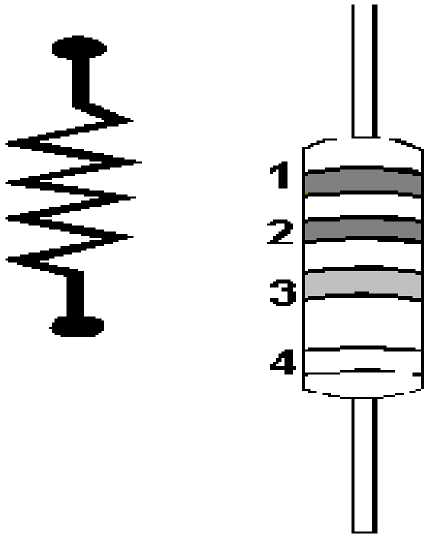
PNP Transistors:

It also does exactly same thing as above except that it has a negative voltage on its collector and a positive voltage on its emitter.



RESISTANCE

Resistance is the opposition of a material to the current. It is measured in Ohms (□). All conductors represent a certain amount of resistance, since no conductor is 100% efficient. To control the electron flow (current) in a predictable manner, we use resistors. Electronic circuits use calibrated lumped resistance to control the flow of current. Broadly speaking, resistor can be divided into two groups viz. fixed & adjustable (variable) resistors. In fixed resistors, the value is fixed & cannot be varied. In variable resistors, the resistance value can be varied by an adjuster knob. It can be divided into (a) Carbon composition (b) Wire wound (c) Special type. The most common type of resistors used in our projects is carbon type. The resistance value is normally indicated by colour bands. Each resistance has four colours, one of the band on either side will be gold or silver, this is called fourth band and indicates the tolerance, others three band will give the value of resistance (see table). For example if a resistor has the following marking on it say red, violet, gold. Comparing these coloured rings with the colour code, its value is 27000 ohms or 27 kilo ohms and its tolerance is ±5%. Resistor comes in various sizes (Power rating). The bigger, the size, the more power rating of 1/4 watts. The four colour rings on its body tells us the value of resistor value as given below.



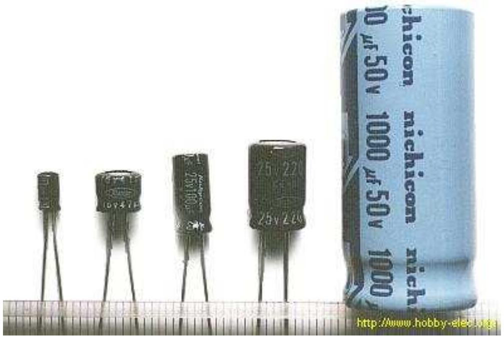
The first rings give the first digit. The second ring gives the second digit. The third ring indicates the number of zeroes to be placed after the digits. The fourth ring gives tolerance (gold ±5%, silver ± 10%, No colour ± 20%).

CAPACITOR

A **capacitor** (formerly known as **condenser**) is a passive two-terminal electrical component used to store energy in an electric field. The forms of practical capacitors vary widely, but all contain at least two electrical conductors separated by a dielectric (insulator); for example, one common construction consists of metal foils separated by a thin layer of insulating film. Capacitors are widely used as parts of electrical circuits in many common electrical devices.

When there is a potential difference (voltage) across the conductors, a static electric field develops across the dielectric, causing positive charge to collect on one plate and negative charge on the other plate. Energy is stored in the electrostatic field. An ideal capacitor is characterized by a single constant value, capacitance, measured in farads. This is the ratio of the electric charge on each conductor to the potential difference between them.

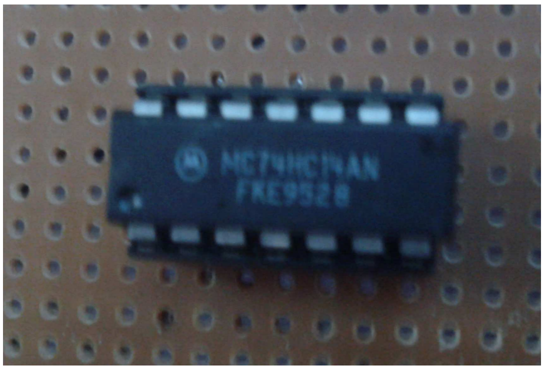
The capacitance is greatest when there is a narrow separation between large areas of conductor, hence capacitor conductors are often called "plates," referring to an early means of construction. In practice, the dielectric between the plates passes a small amount of leakage current and also has an electric field strength limit, resulting in a breakdown voltage, while the conductors and leads introduce an undesired inductance and resistance..



Ceramic Capacitors

Ceramic capacitors are constructed with materials such as titanium acid barium used as the dielectric. Internally, these capacitors are not constructed as a coil, so they can be used in high frequency applications. Typically, they are used in circuits which bypass high frequency signals to ground. These capacitors have the shape of a disk. Their capacitance is comparatively small. The capacitor on the left is a 100pF capacitor with a diameter of about 3 mm. The capacitor on the right side is printed with 103, so 10 x 10^3pF becomes 0.01 µF. The diameter of the disk is about 6 mm. Ceramic capacitors have no polarity. Ceramic capacitors should not be used for analog circuits, because they can distort the signal.

MC74HC14A: HEX SCHMITT TRIGGER INVERTER



The MC74HC14A is identical in pinout to the LS14, LS04 and the HC04. The device inputs are compatible with Standard CMOS outputs; with pullup resistors, they are compatible with LSTTL outputs. The HC14A is useful to “square up” slow input rise and fall times.

Due to hysteresis voltage of the Schmitt trigger, the HC14A finds applications in noisy environments.

• Output Drive Capability: 10 LSTTL Loads

• Outputs Directly Interface to CMOS, NMOS and TTL

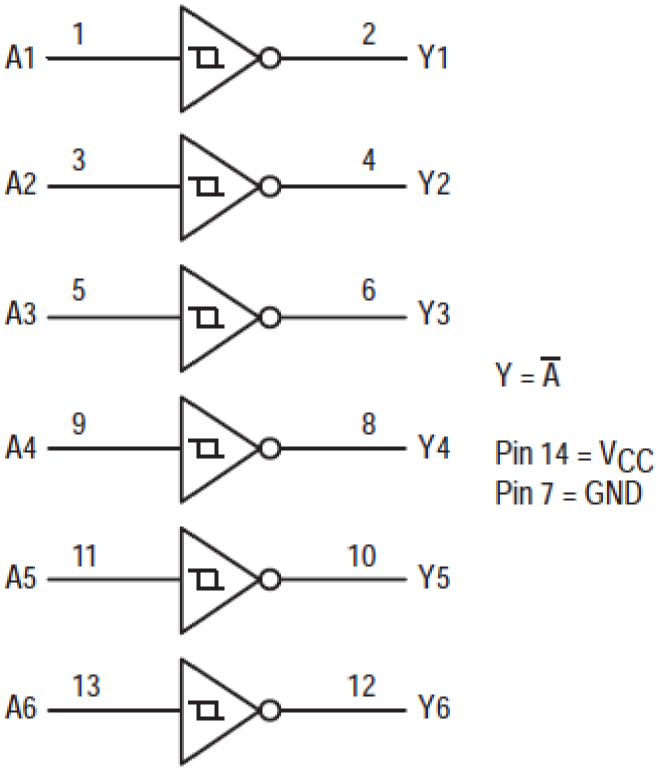
• Operating Voltage Range: 2 to 6V

• Low Input Current: 1mA

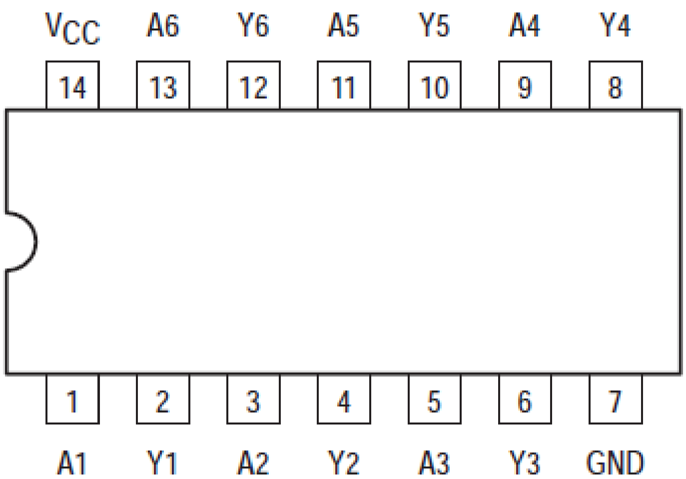
• High Noise Immunity Characteristic of CMOS Devices

• In Compliance With the JEDEC Standard No. 7A Requirements

• Chip Complexity: 60 FETs or 15 Equivalent Gates

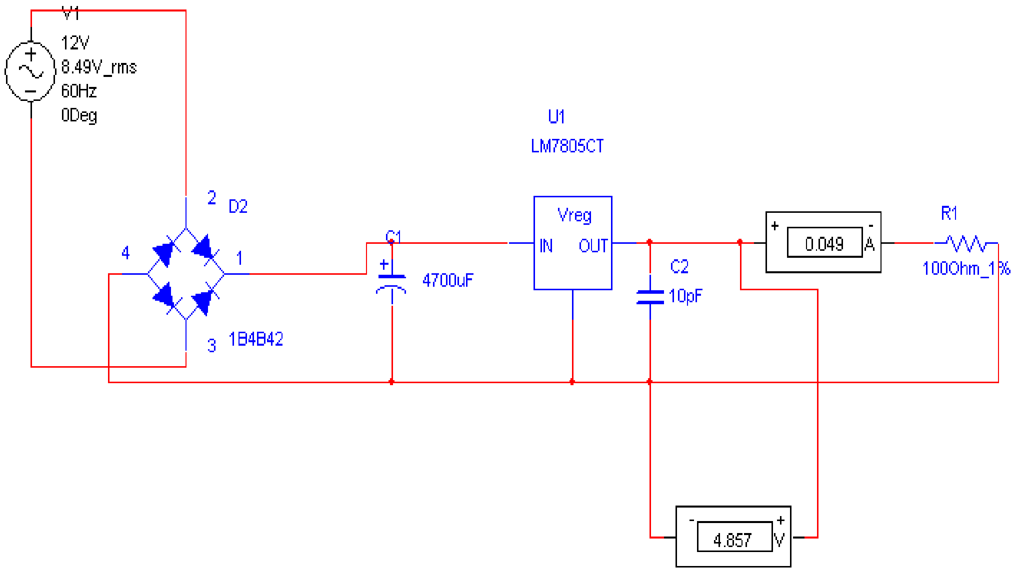


Pinout: 14–Lead Packages (Top View)



3-TERMINAL VOLTAGE REGULATOR

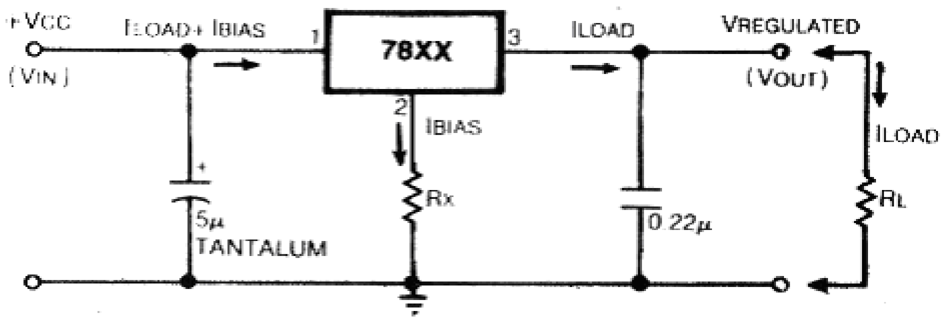
CIRCUIT DIAGRAM OF POWER SUPPLY



One can get a constant high-voltage power supply using inexpensive 3-terminal voltage regulators through some simple techniques described below. Depending upon the current requirement, a reasonable load regulation can be achieved. Line regulation in all cases is equal to that of the voltage regulator used.

Though high voltage can be obtained with suitable voltage boost circuitry using ICs like LM 723, some advantages of the circuits presented below are: simplicity, low cost, and practically reasonable regulation characteristics. For currents of the order of 1A or less, only one zener and some resistors and capacitors are needed. For higher currents, one pass transistor such as ECP055 is needed.

Before developing the final circuits, let us first understand the 3-terminal type constant voltage regulators. Let us see the schematic in Fig. where 78XX is a 3-terminal voltage regulator.



<u>Schematic for obtaining low-voltage regulated output using 3-terminal voltage regulators</u>.

Rectified and filtered unregulated voltage is applied at VIN and a constant voltage appears between pins 2 and 2 of the voltage regulator. *The distribution of two currents in the circuit (IBIAS and ILOAD) is as shown.

It is highly recommended to use the two capacitors as shown. Electrically regulator will be at a distance from the rectifier supply. Thus, a tantalum grade capacitor of 5mf and rated voltage is good. Electrolytic capacitor is not suitable for it is poor in response to load transients, which have high frequency components. At the output side a 0.22mf disc ceramic capacitor is useful to eliminate spurious oscillations, which the regulator might break into because of its internal high gain circuitry.

These voltage regulators have a typical bias current of 5 mA, which is reasonably constant. By inserting a small resistor Rx between pin 2 and ground, the output voltage in many cases. By this method voltage increment of 5 to 10 per cent is practically feasible. However, if a high-value resistance is used to obtain a higher output voltage, a slight variation in bias current will result in wide variation of the output voltage.

Now let us see that what can be done to get a higher but constant output voltage. If to the circuit of Fig. resistor RY and zener Vz are added as shown in Fig., the output voltage is now given by

$$V_{OUT}=V_R+V_Z + I_{BIAS} R_X$$

A constant current flows through RY** because VOUT is constant, and small variations in IBIAS do not change practically the operating point of Vz. This situation is like constant current biasing of zener, which results in a very accurate setting of the zener voltage.

** As long a sVIN>VOUT+2 volts, VOZ is constant from the reasoning of Fig, and thus current through RY is constant.

$$V_{OZ}=V_R + I_{BIAS} R_x$$

Here the pin 2 of the regulator is raised above ground by Vz + IBIAS Rx. Thus, any combination of zener with a proper selection of RY can be used.

For example, Let VR=+15 V for 7815

$$I_{BIAS}=5mA$$

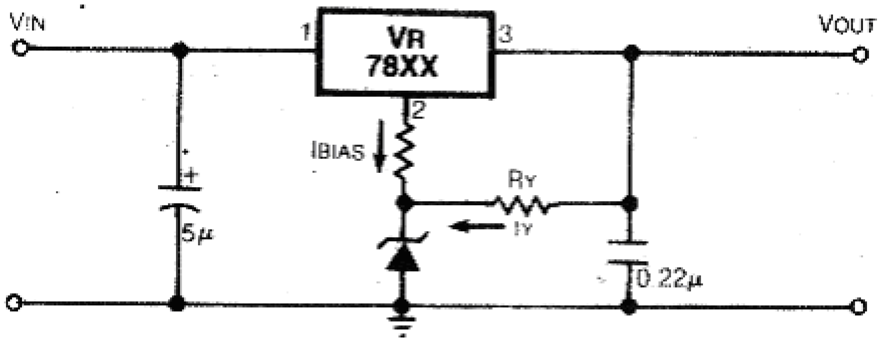
$$V_Z=39V \text{ (standard from ECIL)}$$

For a standard 400mW zener of ECIL make, IZ MAX=10 mA. Thus, if we let pass 5mA through RY to make a 55-volt supply

$$R_Y = \frac{55 - 39}{5 \times 10^{-3}} = 3.2k » 3.3k$$

$$R_X = \frac{55 - 39 - 15}{} = \frac{1}{} = 200 \text{ ohm}$$

IBIAS 5 x 10-3



Schematic for constant high-voltage power supplies

It should be noted here that the maximum input voltage allowed for 78XX regulators is 35V between pins 1 and 2. We see that the actual voltage betweens pin 1 and 2 of the regulator in this circuit is

VIN - VZ - IBIAS RX

It is therefore necessary that VIN be so chosen that voltage between pins 1 and 2 of the IC does not exceed the maximum rating. Also, a high input-output differential voltage VIN-VOUT means more power dissipation in the series-pass element, the regulator. Thus, with proper selection of the input transformer voltage and capacitor, this should be minimized.

For example, if 7805 is used, VR equals + 5V and VZ is 40V, so VOUT=45 volts. For 7805, the maximum input voltage is 35 V and the minimum 7V. Therefore,

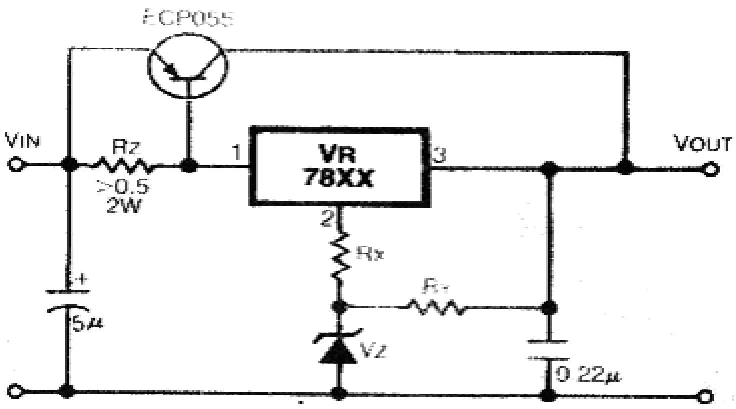
VIN MAX = 45 + 35 - 5 = 75 VOLTS

VIN MIN = 45 + 7 - 5 = 47 VOLTS

Thus, from no-load to full-load condition, the unregulated input voltage-including peak ripple-should be within these limits. This gives a margin of 75-47, i.e. 28 volt. Hence, the designer can work out the maximum transformer voltage from the no-load input voltage chosen on the upper side.

The capacitor's value can be determined from the full load unregulated voltage chosen. Roughly, per 100mA current, 100mf capacitor gives 1-volt peak-to-peak ripple. Hence, capacitor's value can be determined for the desired current.

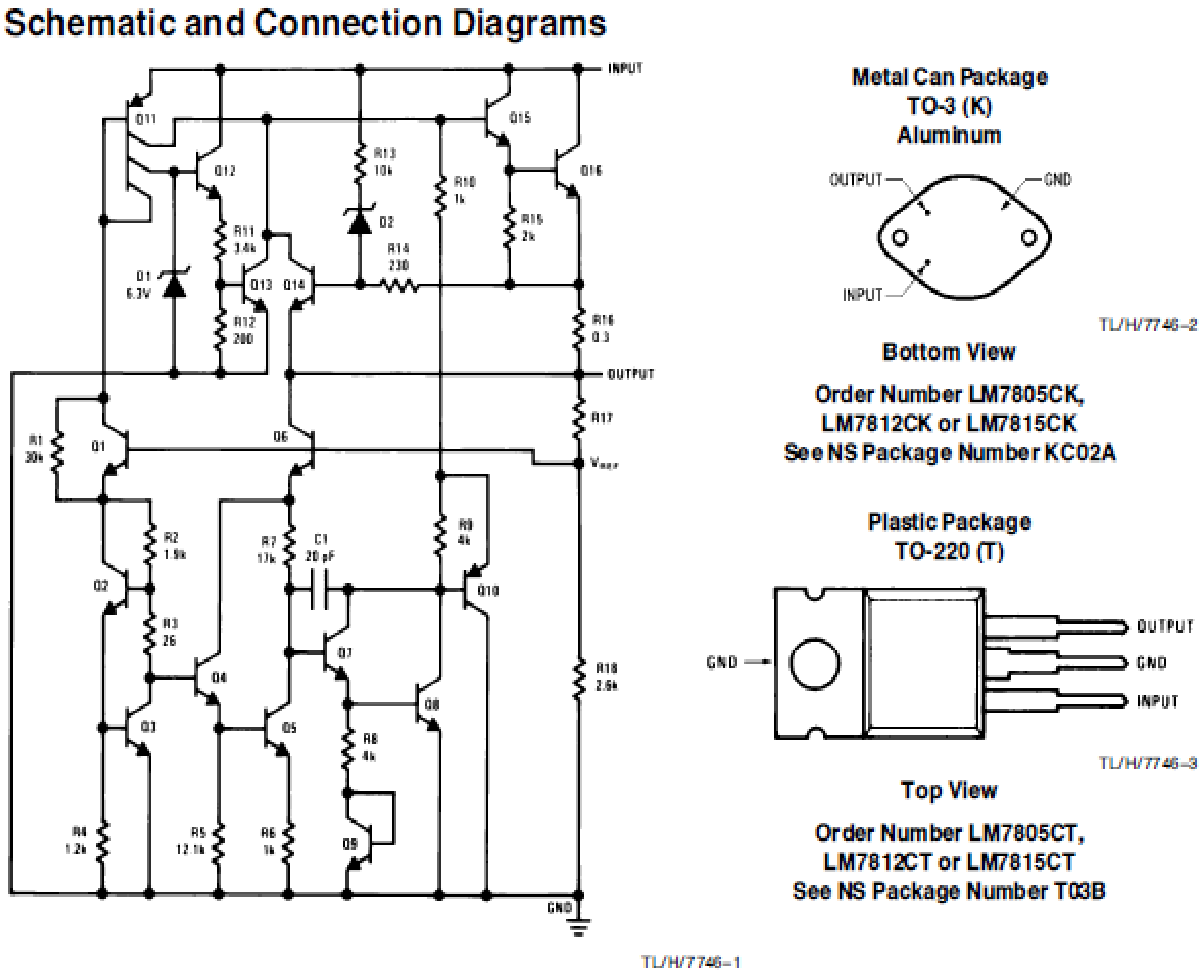
This circuit will have an excellent load and line regulation. For shot-circuit protection, it is recommended to use a fast-blow fuse of suitable value. Although the regulator has inherent short-circuit protection, the maximum current differs from device to device. Adequate heat sink should be used with the regulator.

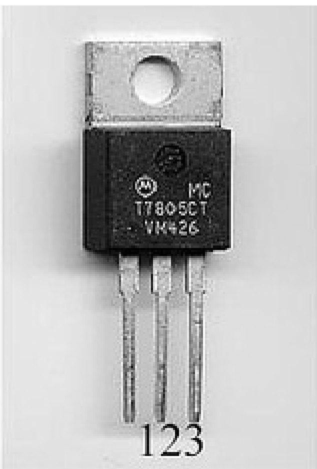


Schematic for constant high-voltage power supplies providing currents in excess of one ampere

Now if currents in excess of 1A are needed, the circuit shown in fig. is useful. This circuit is similar to that in Fig. except that a pass transistor ECP055 is added besides a 0.5-ohm or more resistors. This transistor bypasses the excessive current. By selecting proper Rz the ratio of two currents passing through the regulator and transistor can be altered.

This circuit will show load and live regulation within 1% and will function properly for VIN-VOUT as low as 4 volt. For short-circuit protection, a fast blow fuse is recommended as this circuit does not have inherent short-circuit protection. Adequate heat sink is to be used for the pass transistors. For negative voltages, use 79XX series regulators and ECN055 as the pass transistor. Some advantages of the circuits described above are: the lowest cost among comparable performance circuits, ability to work at low input-output differential, and flexibility in design for various applications.





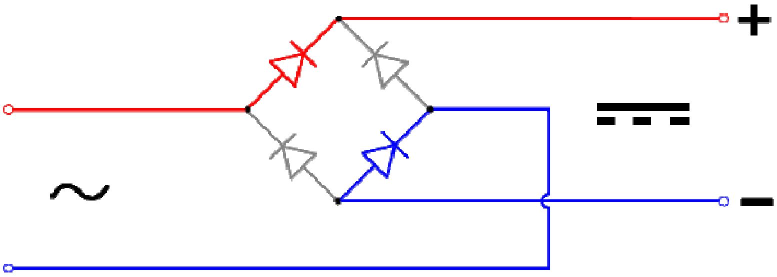
BRIDGE RECTIFIER

A bridge rectifier makes use of four diodes in a bridge arrangement to achieve full-wave rectification. This is a widely used configuration, both with individual diodes wired as shown and with single component bridges where the diode bridge is wired internally.

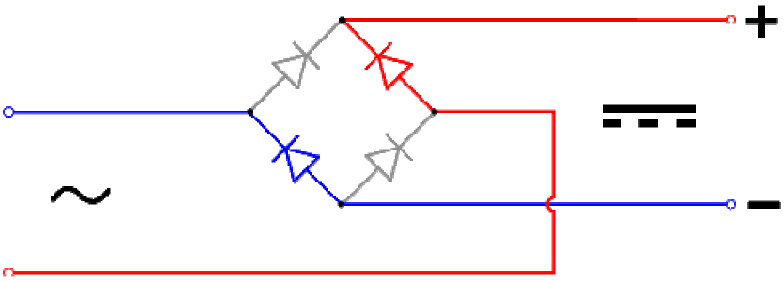
BASIC OPERATION

According to the conventional model of current flow originally established by Benjamin Franklin and still followed by most engineers today, current is *assumed* to flow through electrical conductors from the **positive** to the **negative** pole. In actuality, free electrons in a conductor nearly always flow from the **negative** to the **positive** pole. In the vast majority of applications, however, the *actual* direction of current flow is irrelevant. Therefore, in the discussion below the conventional model is retained.

In the diagrams below, when the input connected to the **left** corner of the diamond is **positive**, and the input connected to the **right** corner is **negative**, current flows from the **upper** supply terminal to the right along the **red** (positive) path to the output, and returns to the **lower** supply terminal via the **blue** (negative) path.



When the input connected to the **left** corner is **negative**, and the input connected to the **right** corner is **positive**, current flows from the **lower** supply terminal to the right along the **red** path to the output, and returns to the **upper** supply terminal via the **blue** path.

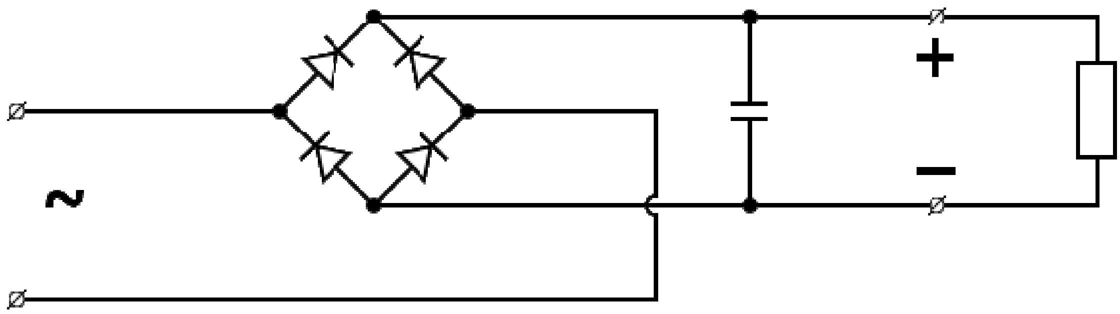


In each case, the upper right output remains positive and lower right output negative. Since this is true whether the input is AC or DC, this circuit not only produces a DC output from an AC input, it can also provide what is sometimes called "reverse polarity protection". That is, it permits normal functioning of DC-powered equipment when batteries have been installed backwards, or when the leads (wires) from a DC power source have been reversed, and protects the equipment from potential damage caused by reverse polarity.

Prior to availability of integrated electronics, such a bridge rectifier was always constructed from discrete components. Since about 1950, a single four-terminal component containing the four diodes connected in the bridge configuration became a standard commercial component and is now available with various voltage and current ratings.

OUTPUT SMOOTHING

For many applications, especially with single phase AC where the full-wave bridge serves to convert an AC input into a DC output, the addition of a capacitor may be desired because the bridge alone supplies an output of fixed polarity but continuously varying or "pulsating" magnitude (see diagram above).



The function of this capacitor, known as a reservoir capacitor (or smoothing capacitor) is to lessen the variation in (or 'smooth') the rectified AC output voltage waveform from the bridge. One explanation of 'smoothing' is that the capacitor provides a low impedance path to the AC component of the output, reducing the AC voltage across, and AC current through, the resistive load. In less technical terms, any drop in the output voltage and current of the bridge tends to be canceled by loss of charge in the capacitor. This charge flows out as additional current through the load. Thus the change of load current and voltage is reduced relative to what would occur without the capacitor. Increases of voltage correspondingly store excess charge in the capacitor, thus moderating the change in output voltage / current.

The simplified circuit shown has a well-deserved reputation for being dangerous, because, in some applications, the capacitor can retain a *lethal* charge after the AC power source is removed. If supplying a dangerous voltage, a practical circuit should include a reliable way to safely discharge the capacitor. If the normal load cannot be guaranteed to perform this function, perhaps because it can be disconnected, the circuit should include a bleeder resistor connected as close as practical across the capacitor. This resistor should consume a current large enough to discharge the capacitor in a reasonable time, but small enough to minimize unnecessary power waste.

Because a bleeder sets a minimum current drain, the regulation of the circuit, defined as percentage voltage change from minimum to maximum load, is improved. However in many cases the improvement is of insignificant magnitude.

The capacitor and the load resistance have a typical time constant $\tau = RC$ where C and R are the capacitance and load resistance respectively. As long as the load resistor is large enough so that this time constant is much longer than the time of one ripple cycle, the above configuration will produce a smoothed DC voltage across the load.

In some designs, a series resistor at the load side of the capacitor is added. The smoothing can then be improved by adding additional stages of capacitor–resistor pairs, often done only for sub-supplies to critical high-gain circuits that tend to be sensitive to supply voltage noise.

The idealized waveforms shown above are seen for both voltage and current when the load on the bridge is resistive. When the load includes a smoothing capacitor, both the voltage and the current waveforms will be greatly changed. While the voltage is smoothed, as described above, current will flow through the bridge only during the time when the input voltage is greater than the capacitor voltage. For example, if the load draws an average current of n Amps, and the diodes conduct for 10% of the time, the average diode current during conduction must be 10n Amps. This non-sinusoidal current leads to harmonic distortion and a poor power factor in the AC supply.

In a practical circuit, when a capacitor is directly connected to the output of a bridge, the bridge diodes must be sized to withstand the current surge that occurs when the power is turned on at the peak of the AC voltage and the capacitor is fully discharged. Sometimes a small series resistor is included before the capacitor to limit this current, though in most applications the power supply transformer's resistance is already sufficient.

Output can also be smoothed using a choke and second capacitor. The choke tends to keep the current (rather than the voltage) more constant. Due to the relatively high cost of an effective choke compared to a resistor and capacitor this is not employed in modern equipment.

Some early console radios created the speaker's constant field with the current from the high voltage ("B +") power supply, which was then routed to the consuming circuits, (permanent magnets were then too weak for good performance) to create the speaker's constant magnetic field. The speaker field coil thus performed 2 jobs in one: it acted as a choke, filtering the power supply, and it produced the magnetic field to operate the speaker.

DIODE

A diode is a semiconductor device which allows current to flow through it in only one direction. Although a transistor is also a semiconductor device, it does not operate the way a diode does. A diode is specifically made to allow current to flow through it in only one direction. Some ways in which the diode can be used are listed here.

- A diode can be used as a rectifier that converts AC (Alternating Current) to DC (Direct Current) for a power supply device.
- Diodes can be used to separate the signal from radio frequencies.
- Diodes can be used as an on/off switch that controls current.

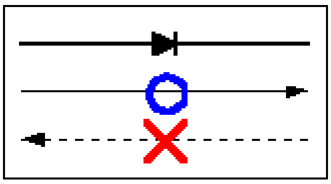
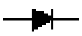
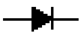


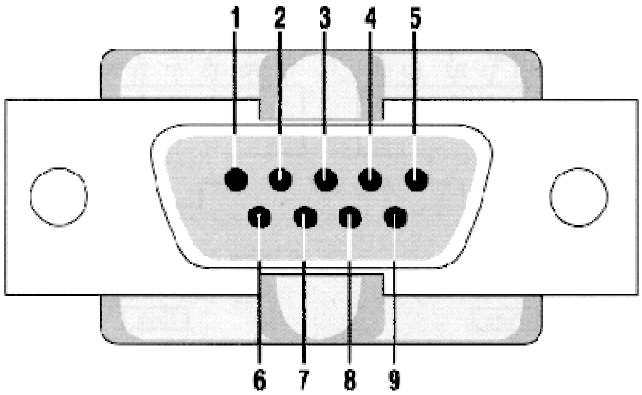
Fig. 2.26 Diode Symbol

This symbol  is used to indicate a diode in a circuit diagram. The meaning of the symbol is (Anode)  (Cathode).

Current flows from the anode side to the cathode side.

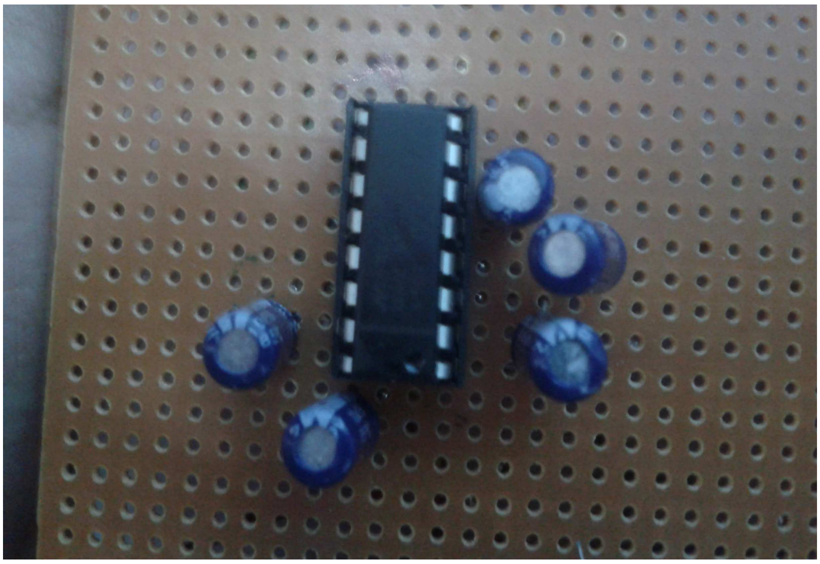
Although all diodes operate with the same general principle, there are different types suited to different applications. For example, the following devices are best used for the applications noted.

D-CONNECTOR



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

RS-232



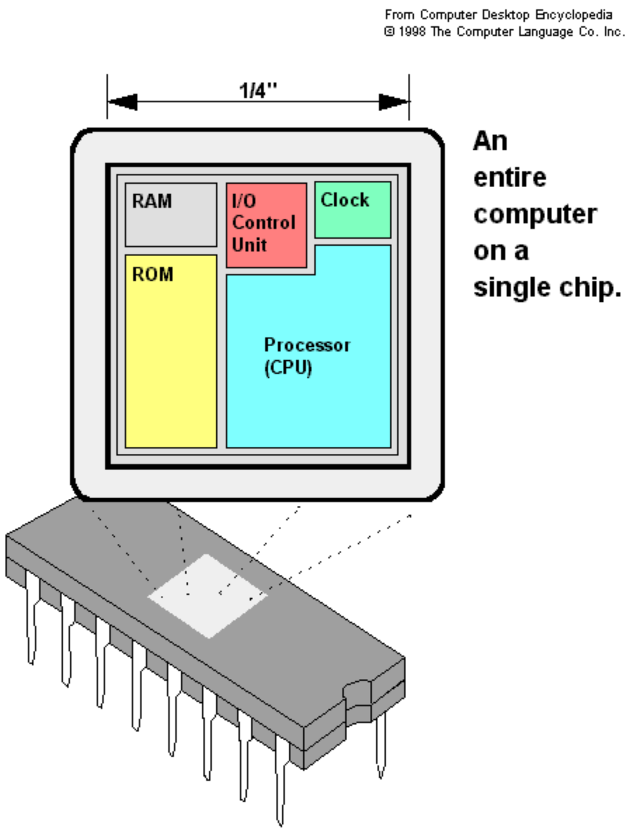
In telecommunications, RS-232 is a standard for serial binary data signals connecting between a *DTE* (Data terminal equipment) and a *DCE* (Data Circuit-terminating Equipment)[14]. It is commonly used in computer serial ports**.** In RS-232, data is sent as a time-series of bits. Both synchronous and asynchronous transmissions are supported by the standard. In addition to the data circuits, the standard defines a number of control circuits used to manage the connection between the DTE and DCE [14]. Each data or control circuit only operates in one direction that is, signaling from a DTE to the attached DCE or the reverse. Since transmit data and receive data are separate circuits, the interface can operate in a full duplex manner, supporting concurrent data flow in both directions [15]. The standard does not define character framing within the data stream, or character encoding.

Serial transfer using TI and RI flags

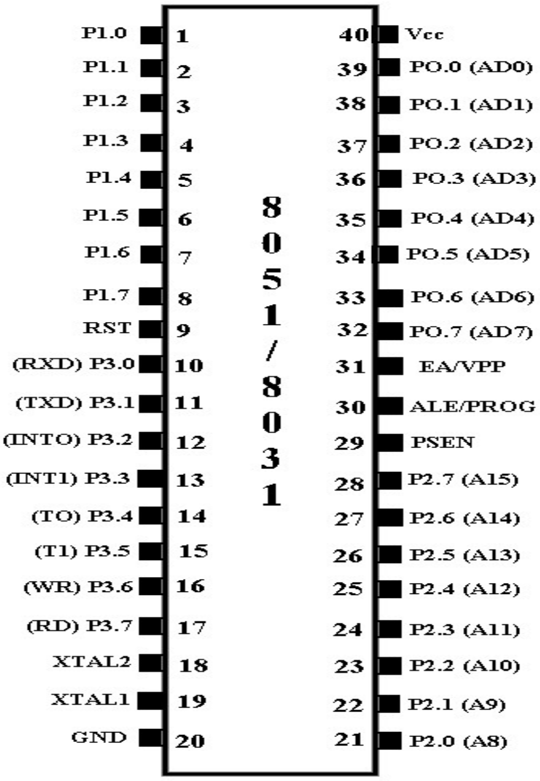
After setting the baud rates of the two devices both the devices are now ready to transmit and receive data in form of characters. Transmission is done when TI flag is set and similarly data is known to be received when the Rx flag is set. The microcontroller then sends an AT command to the modem in form of string of characters serially just when the TI flag is set. After reception of a character in the SBUF register of the microcontroller (response of MODEM with the read message in its default format or ERROR message or OK message), the RI flag is set and the received character is moved into the physical memory of the microcontroller.

8051 MICROCONTROLLER

A microcontroller is a single chip that contains the processor (the CPU), non-volatile memory for the program (ROM or flash), volatile memory for input and output (RAM), a clock and an I/O control unit. Also called a "computer on a chip," billions of microcontroller units (MCUs) are embedded each year in a myriad of products from toys to appliances to automobiles. For example, a single vehicle can use 70 or more microcontrollers. The following picture describes a general block diagram of microcontroller.



The pin diagram of the 8051 shows all of the input/output pins unique to microcontrollers:



The following are some of the capabilities of 8051 microcontroller.

- ✓ Internal ROM and RAM
- ✓ I/O ports with programmable pins
- ✓ Timers and counters
- ✓ Serial data communication

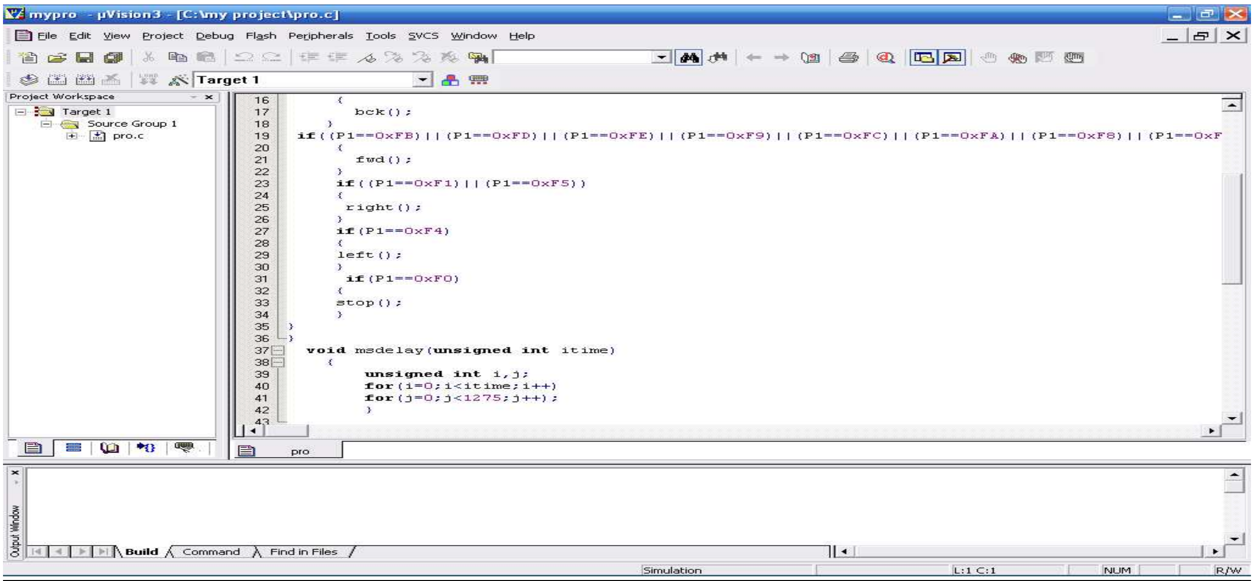
The 8051 architecture consists of these specific features:

- 16 bit PC &data pointer (DPTR)
- 8 bit program status word (PSW)
- 8 bit stack pointer (SP)
- Internal ROM 4k
- Internal RAM of 128 bytes.
- 4 register banks, each containing 8 registers
- 80 bits of general purpose data memory
- 32 input/output pins arranged as four 8 bit ports: P0-P3
- Two 16 bit timer/counters: T0-T1

- Two external and three internal interrupt sources Oscillator and clock circuits.

Simulator

KEIL Micro Vision is an integrated development environment used to create software to be run on embedded systems (like a microcontroller). It allows for such software to be written either in assembly or C programming languages and for that software to be simulated on a computer before being loaded onto the microcontroller. The software used is c programming



Keil µVision3 is an IDE (Integrated Development Environment) that helps write, compile, and debug embedded programs. It encapsulates the following components:

- A project manager.
- A make facility.
- Tool configuration.
- Editor.
- A powerful debugger.

CIRCUIT

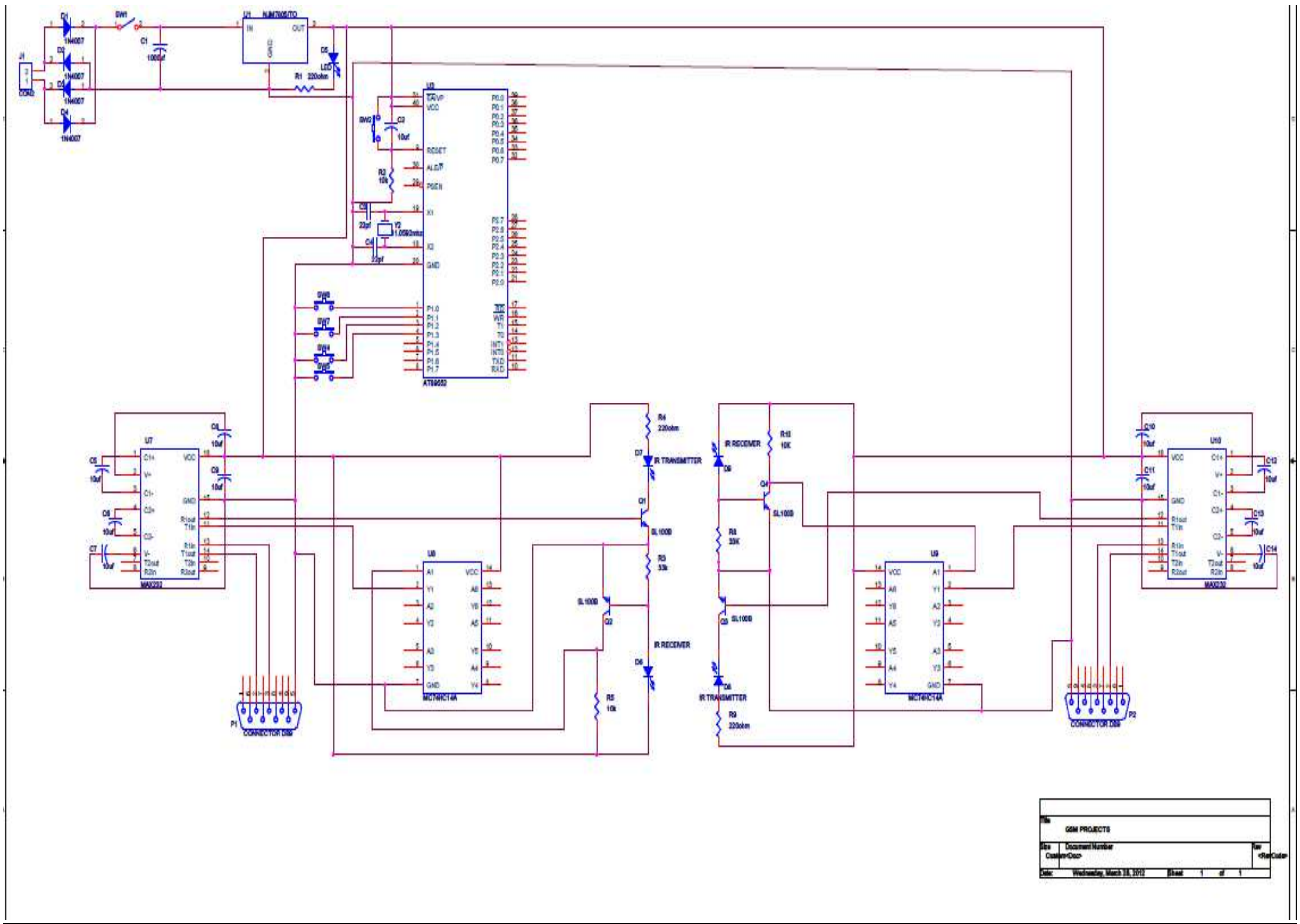
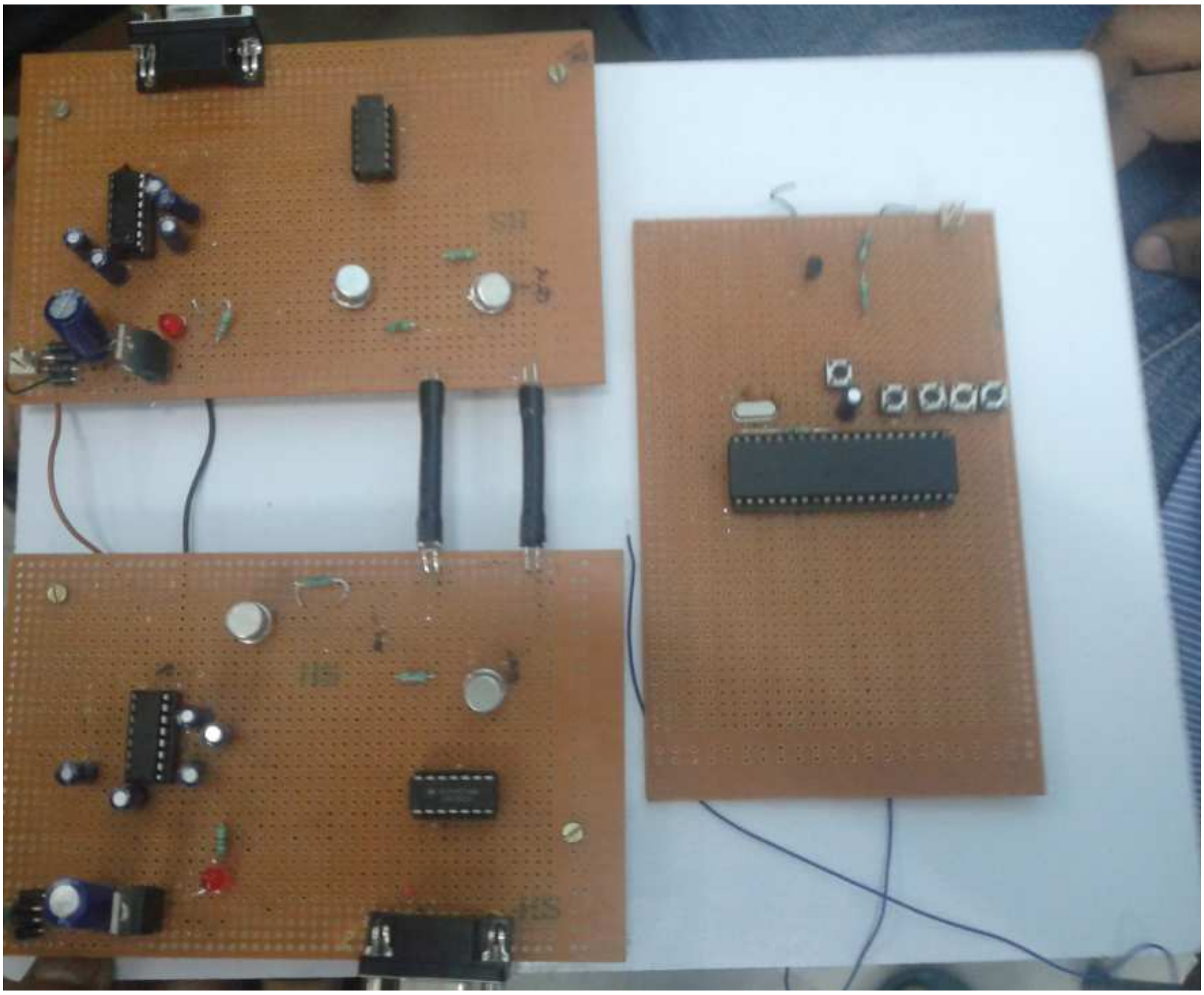


FIG: CIRCUIT OF OPTICAL FIBRE COMMUNICATIO N BASED MESSAGE RETRIEVAL WITH PASSWORD



TRANSMITTER

Data signals transmitted through pin 3 of 9-pin (or 2 of 25-pin) 'D' connector of RS232 COM port are sent to pin 8 of MAX232 and it converts these EIA RS232C compatible levels of $\pm$9V to 0/5V TTL levels, as given in Table-1. The output pin 9 of MAX232 IC drives the pnp transistor SK100 and power the IR LEDs. Output pin 9 also drives an LED indicator (LED2) during the positive output at its pin 9. At logic '0' output at pin 9, LED2 goes 'off', but drives the pnp transistor through a bias resistor of 1 kilo-ohm (R5), to switch 'on' IR LED1 and IR LED2 and also a visible LED3. Since very low drive current is used, use of high-efficiency visible LEDs, which light up at 1 mA, is needed. The electrical pulses sent by the COM port are now converted into corresponding modulated pulses of IR light.

TABLE I		
MAX 232 CONVERSION LEVELS		
TTL +5V	to	-9V RS 232
TTL 0V	to	+9V RS 232
RS 232 +9V	to	0V TTL
RS 232 –9V	to	5V TTL

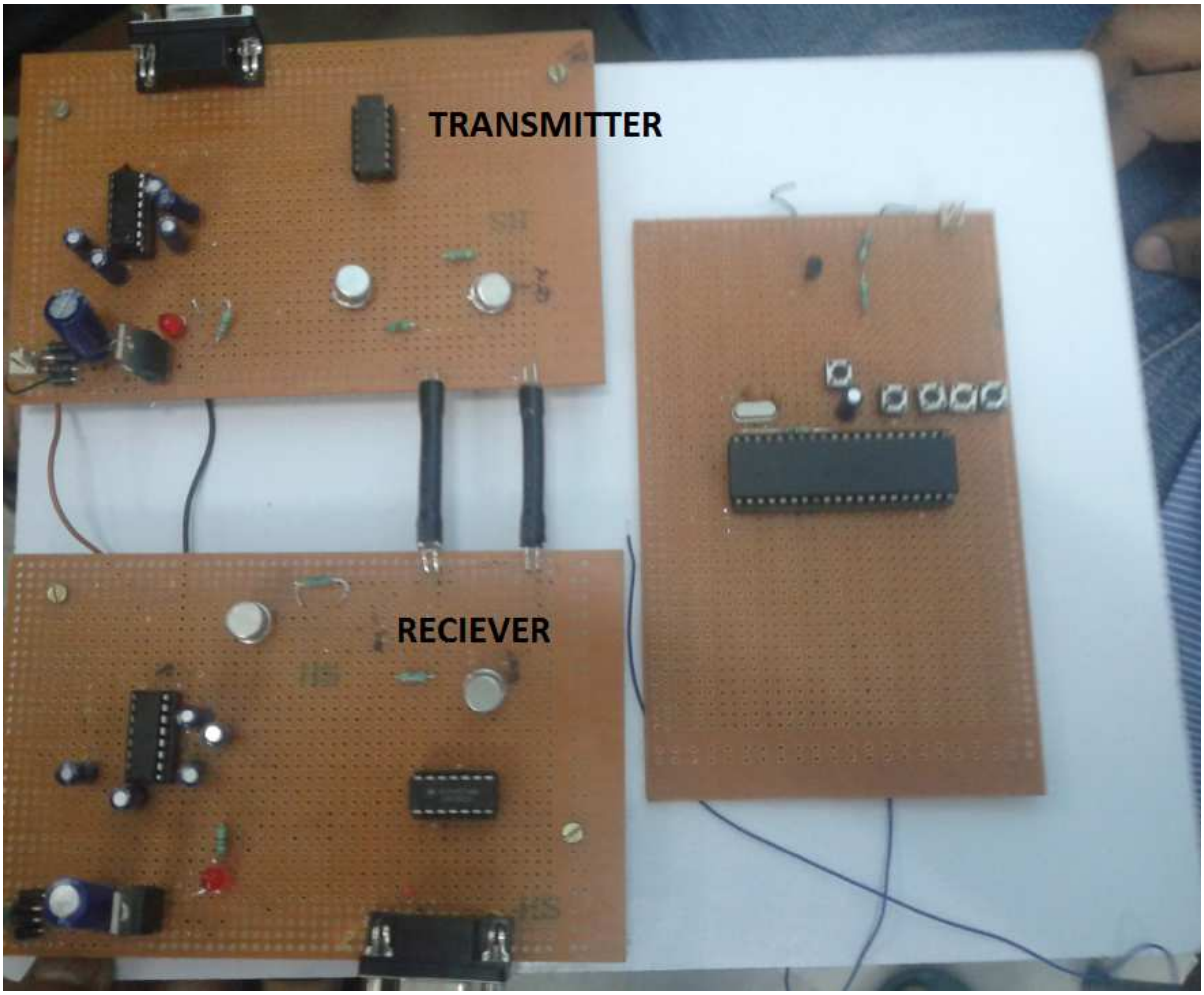
RECEIVER

The IR signals are detected by a photodiode (D1). (A photodiode is reversing biased and breaks down when IR light falls on its junction.) The detected TTL level (0/5V) signals are coupled to pin 10 of MAX232 IC. These TTL levels are converted to $\pm$9V levels internally and output at pin 7.

A visible LED1 at pin 7 of MAX232 IC indicates that the signals are being received. Pin 7 is also connected to pin 2 (receiver pin) of 9-pin (or pin 3 of 25-pin) 'D' connector used for the serial port in the PC, so that the data may be read. The optical signals received by the photodiodes are in fact converted to electrical pulses and both PCs 'think' that there is a null modem cable connected between them. Table-II shows the correspondence between the various pins of a 9-pin (or 25-pin) 'D' connector of serial port of PC. In some PCs, the serial port of is terminated into a 9-pin 'D' connector and in some others into a 25-pin 'D' connector.

TABLE II

DB 9 Pin	DB25 Pin	Signal Direction	Description
1	8	In	DCD (data carrier detect)
2	3	In	RX (receiver data)
3	2	Out	TX (transmit data)
4	20	Out	DTR (data terminal ready)
5	7	---	GND (signal ground)
6	6	In	DSR (data set ready)
7	4	Out	RTS (request to sent)
8	5	In	CTS (clear to send)
9	22	In	RI (ring indicator)


TRANSMITTER
RECIEVER

PROGRAMMING FOR COMMUNICATION

There are two methods for writing the program, namely, polling method and interrupt method.

Polling Method

In this method, the UART is repeatedly checked for any new data in its receiving buffer.

Interrupt Method

An interrupt handler is used to remove the data from the UART when an interrupt is generated.

The port addresses and IRQs for COM ports are as follows: -

Name	Address	IRQ	Int (Hex)
COM1	3F8	4	0C
COM2	2F8	3	0B
COM3	3E8	4	0C
COM4	2E8	3	0B

We use in port ( ) and output port ( ) instructions to read and write to the registers used for serial communication. Though there are only eight addresses, we are able to control twelve registers by using divisor latch-enable bit (DLAB), i.e. bit 7 of the line control register.

UART SETTINGS

As the computer works with data in parallel form, we need the serial transmission converted back to parallel format so that data could be used. This can be done using a universal asynchronous receiver transmitter (UART) chip.

In order to use the UART, we first need to make certain register settings. The sequential steps for polling method are given below:

1) Stop interrupt generation on the UART, so that the UART doesn’t interrupt the initialization.

2) Now the interrupt vector is set.

3) The next step is to set the speed at which the communication takes place. This is done by setting bit 7 (DLAB) of the LCR to logic 1, so as to have an access to divisor latch high and low bytes. Some common baud rates and the corresponding hex values we used to set the high and low bytes. (Baud rates are based on the assumption that the UART utilizes a 1.832MHz crystal, which is divided by 16 inside the UART to obtain the basic baud rate of 115,200 bits/sec.)

4) Then we set the divisor latch-low byte followed by divisor latch-high byte in the registers as per the address.

5) The next step is to reset/turn off the divisor latch access bit (DLAB) and set the word length and parity in the LCR. Here we’ve selected the word length as 8 bits, no parity, and one stop bit. This is done by moving 0x03 to the line control register. As this will also turn off the DLAB, we can access the interrupt enable register and receiver and transmitter buffers.

6) Now the FIFO buffers are turned ‘on’ and cleared.

7) DTR, RTS, and OUT 2 are made high in the modem control register.

PROGRAM IN C:

```
#include <stdio.h>

#include <dos.h>

#include <conio.h>

#include<graphics.h>

#include<stdlib.h>

#include<io.h>

#include<fcntl.h>

#define DEL 25  /* Preprocessor - Delay Variable */

#define COM 0X03f8  /*0x02f8 -com2,0x03f8-com1 */

char gra='Y';           /* Global Variables */

int flag=0;

union REGS inregs,outregs; /* Union declaration for registers */

FILE *fp;  /* File declaration */

int status;

char temp='\n',t2;

int t1=10;


void main(void)

{

        char ch,chr,chs;

        clrscr();

        if(flag==0)
```

```
    flag++;
textcolor(4);gotoxy(26,6);
cprintf("PC TO PC FILE TRANSFER PROTOCOL");
gotoxy(34,9);textcolor(10);
cprintf("R");textcolor(7);cprintf("Receive mode");
textcolor(14);gotoxy(35,12);
cprintf("S");textcolor(7);cprintf("end mode");
textcolor(6);gotoxy(37,15);
cprintf("E");textcolor(7);cprintf("Exit");
ch = getch();
switch(toupper(ch))
{
    case 'R':  R:
                    clrscr();
                    textcolor(4);gotoxy(26,6);
                    cprintf("SERIAL PORT COMMUNICATION");
                    textcolor(138);gotoxy(33,9);
                    cprintf("RECEIVE MODE");
                    textcolor(9);gotoxy(33,12);
                    cprintf("A");textcolor(7);cprintf("lign device");
                    textcolor(11);gotoxy(33,15);
                    cprintf("F");textcolor(7);cprintf("ile receive");
                    textcolor(6);gotoxy(36,18);
```

```
		cprintf("Q");textcolor(7);cprintf("uit");
		chr = getch();
		switch(toupper(chr))
		{
			case 'A': ralgn();break;
			case 'F': f_rcv();break;
			case 'Q': main();
			default : clrscr();
				printf("Wrong Key Pressed");
			goto R;
		}
	break;
case 'S': S:
		clrscr();
		textcolor(4);gotoxy(26,6);
		cprintf("INFRARED/LASER COMMUNICATION");
		textcolor(142);gotoxy(36,9);
		cprintf("SEND MODE");
		textcolor(9);gotoxy(34,12);
		cprintf("A");textcolor(7);cprintf("lign device");
		textcolor(11);gotoxy(34,15);
		cprintf("T");textcolor(7);cprintf("ransfer file");
		textcolor(6);gotoxy(38,18);
```

```
}                       cprintf("Q");textcolor(7);cprintf("uit");
                        chs = getch();
                        switch(toupper(chs))
                        {
                             case 'A': salgn();break;
                             case 'T': f_snd();break;
                             case 'Q': main();
                             default : clrscr();
                                   printf("Wrong Key Pressed");
                                   goto S;
                        }
                        break;
   case 'E': clrscr();
        textcolor(143);
        gotoxy(35,13);
        cprintf("GOOD BYE");
        exit(1);
   default : clrscr();
        printf("Wrong Key Pressed");
        main();
        return ;

}
```

```
}

   /* Function for receive (For Device Alignment) */

ralgn(void)
{
        char st = ' ';    /* Local variables */
        clrscr();
        gotoxy(30,2);
        textcolor(10);
        cprintf("RECEIVE MODE :");
        textcolor(9);cprintf(" ALIGN DEVICE");
        printf("\n");
        initial(); /* Call Initialisation routine */
        loop:
        if(!kbhit())
        {
            if(st==0x04) /* Check for end of Transmission */
            {
                clrscr();
                textcolor(140);
                gotoxy(30,12);
                cprintf("ALIGNED PROPERLY");
```

```
}

                gotoxy(48,24);

                printf(" Press any key to quit .");

                getch();

                main(); /* Got to main function after aligning properly */

        }

        status = inp(0X3fd); /*Checking status at com1 port */

        // printf("%d",status);

        if((status & 0x01)==0x00) /* Check for Data Ready */

                goto loop;

        else if(!kbhit())

        {

                st = inp(COM); /*Get  character from com1 port till */


                printf("%c",st); /* key hit or end of transmission */

                goto loop;

        }

        else

                main(); /*Call main function if key hit */

      }

      return;

}


  /*Function for File Receive */
```

```
f_rcv()

{

        int flag=0,bytecount=0,count;  /* Local Variables */

        float ot = 0.00,nt = 0.00;

        char ch,st[55000],fnm[30];

        clrscr();

        initial(); /*Calling Initialisation Routine */

        ot = clock() / 18.2; /*Calculate exec time in secs from start of

program */


        gotoxy(2,2);

        printf(" FILE NAME ?  : ");

        fp=fopen(gets(fnm),"wb");  /*Get file name in write mode */

        gotoxy(26,10);

        printf("(Ready for) RECEIVING DATA ....");

        gotoxy(50,24);

        textcolor(138);

        cprintf("Don't press any KEY It may loss data");


       loop:

        nt = clock()/18.2; /*Calculate exec time in secs from start of

                                        program */
```

```
        status = inp(0X3FD);/*Get character from com1 port */
if((status & 0x01)==0x00) /* Check for Data Ready */
        {

          /* Check for no data reception for five seconds after
            start of reception if  no data is received continue other process
*/
           if((bytecount>0) && (nt-ot)>5.0)
           {
                    clrscr();
                    for(count=0;count<flag;count++)
                    {
                              gotoxy(26,10);
                              textcolor(11);
                              cprintf(" Saving data in");
                              gotoxy(43,10);
                              textcolor(12);
                              cprintf(" %s",fnm);
                              /*Dump the data received in a File */
                              fprintf(fp,"%c",st[count]);
                    }
                    fclose(fp);
                    gotoxy(26,13);
```

```
            textcolor(11);
            cprintf(" File %s of %d bytes created ",fnm,count);
            gotoxy(50,24);
            textcolor(7);
            cprintf(" Press any key to quit .");
            getch();
            main();
    }
    goto loop;
}
 else
    if(!kbhit())
    {
            st[flag] = inp(COM);/*Get  character from Com1 port */
            flag++;
            bytecount++;
            ot = clock()/18.2; /*Calculate exec time during receiving */
            goto loop;
    }
    else
    {
            /* If transmission is cut terminate abnormally */
            clrscr();
```

```
        for(count=0;count<flag;count++)
        {
            gotoxy(26,3);
            textcolor(140);
            cprintf(" TERMINATED ABNORMALLY ");
            gotoxy(26,10);
            textcolor(11);
            cprintf(" Saving data in");
            textcolor(12);
            cprintf(" %s",fnm);
            fprintf(fp,"%c",st[count]);
        }
        fclose(fp);
        gotoxy(26,13);
        textcolor(11);
        cprintf(" File %s of %d bytes created ",fnm,count);
        sleep(5);
        main();/*Go to main after dumping in file */
    }
    return;
}


/* Function for send align ( for device alignment) */
```

```
salgn(void)
{
    int flag=0; /* Local Variables */
    char st[127];
    clrscr();
    initial();
    textcolor(14);
    cprintf("Type the sentence ( < 127 chars)");
    puts("\n");
    gets(st);    /* Get string to send */
    loop:
    status = inp(0X3FD); /* Get com1 port status */
    if((status & 0x20)==0x00) /* Check Transfer holding register empty */
      goto loop;
    else
    {
      do
      {
            if(!kbhit())     /* Check for key hit */
            {
                  outport(COM,0X0D); /* Send carriage return */
                  outport(COM,0X0A); /* Send line feed */
```

```
			if(flag==strlen(st)) /* Check for length of string*/
			{
				printf("\n");
				flag=0;
				outport(COM,0X0D);
				/* Send carriage return */
				delay(5);
outport(COM,0X0A); /* Send carriage return */
				delay(5);
			}
			else
			{
				outport(COM,st[flag]); /* Send character to com1 port*/
				printf("%c",st[flag]);
				flag++;
				delay(DEL);
			}

		}


	if(kbhit()) /* Check key hit */
	{
```

```
            delay(1);
            outport(COM,0x04);/*Send End of transmission */
            main();
        }
    }while(!kbhit());
  }//else
}

/*Function for file transfer*/
f_snd()
{
    int flag=0,count=0,fl;     /* Local Variables */
    char ch,st[55000],fnm[20];
    clrscr();
    initial(); /* Calling Initialisation Routine */
    gotoxy(2,2);
    printf("FILE NAME ? : ");
    fp = fopen(gets(fnm),"rb"); /* Get file name to be sent */
    if(fp==NULL)
    {
      clrscr();
      gotoxy(35,13);
      printf(" FILE NOT FOUND !");
```

```
    delay(1000);
    main();
}
else
{
    int handle;
    handle=open(fnm,O_RDONLY);
    fl = filelength(handle); /* Calculate file length */
    gotoxy(23,20);
    printf("File being transferred has %u bytes",fl);
    do
    {
            ch = fgetc(fp);
            st[count] = ch;
            count++;
    }
    while(count<=fl);
}
//  printf("File being transferred has %d characters",count--);
printf("\n");
fclose(fp);


loop:
```

```
status = inp(0x3FD); /*Check com1 port status */
if((status & 0x20)==0x00)/* Check Transfer holding register empty */
    goto loop;
else
do
{
    if(flag==fl) /*Check for file length */
    {

            gotoxy(50,24);
            printf(" Press any key to exit !");
            getch();
            main(); /*Call main function */
    }
    else
    {
            outport(COM,st[flag]);/*Send each character in the file*/
            printf("\t%c",st[flag]);
            flag++;
            delay(DEL);
    }
}
while(!kbhit()); /* Check for key hit */
```

```
}

 /*Initiialisation Function */
initial()
{
        inregs.h.ah = 0; /*Initialisation of port */
        inregs.h.al = 0X63; /* Baudrate , Parity , Databits , Stopbit(s) */
        inregs.x.dx = 0;  /*Select port COM1 */
        int86(0x14,&inregs,&outregs);/*Complete Communication service
Interrupt*/
}

  /*Function for Splash screen*/
/*splash(void)
{
int d=DETECT,m,j,i;
struct palettetype pal; /* Structure for palette colours */
//initgraph(&d,&m,""); /*Initialisation for splash screen */
//getpalette(&pal);    /*Get palette colours*/
/* for(i=0;i<=pal.size;i++)
 setrgbpalette(pal.colors[i],i*5,i*4,i*4);/*Combination of RGB
colours*/
/* setfillstyle(8,8);
```

```
 setcolor(15);

 settextstyle(1,0,4);

 setbkcolor(4);

 for(i=0;i<17;i++)    /* Writing text with RGB palette colors */

/* {

  setcolor(i);

  outtextxy(45+i,200+i,"PC to PC Laser/IR Communication");

 }

 sleep(1);

 cleardevice();

for(i=0;i<17;i++)    /* Writing text with RGB palette colors */

/* {

  setcolor(i);

  outtextxy(175+i,200+i,"Mostek Electronics");

 }

 sleep(1);

cleardevice();

for(i=0;i<17;i++)    /* Writing text with RGB palette colors */

/* {

  setcolor(i);

  outtextxy(160+i,175+i,"K.S.Sankar");

 }

sleep(1);
```

```
cleardevice();

closegraph();

}

/*-----end------*/
```

Coding for 8051 (password retrieval system)

```
#include<reg51.h>
sbit C1=P1^0;
sbit C2=P1^1;
sbit C3=P1^2;
sbit C4=P1^3;
sbit BUZZPORT=P1^4;
unsigned char keypad(void);
void buzzer(unsigned int time);
void DelayMs(unsigned int count);
void main()
{
  while(keypad())
  {
        buzzer(100);

  }



}
void buzzer(unsigned int time)
{

            BUZZPORT=1;
            DelayMs(time);
```

```
            BUZZPORT=0;
}
void DelayMs(unsigned int count)
{  // mSec Delay 11.0592 Mhz
   unsigned int i;                  // Keil v7.5a
   while(count) {
      i = 115;                   // 115  exact value
            while(i>0)
            i--;
      count--;
   }
}

unsigned char keypad(void)
{
unsigned char l=0;
unsigned char b[4];

while(l<3)
{
if(C1==0)
{
while(C1==0);
b[l]=0;


l++;
}
if(C2==0)
{
while(C2==0);
b[l]=1;

l++;
```

```
}
if(C3==0)
{
while(C3==0);
b[l]=2;
l++;
}
if(C4==0)
{
while(C4==0);
b[l]=3;
l++;
}
}

if((b[0]==1)&&(b[1]==1)&&(b[2]==1))
{
return 1;
}
else
{
return 0;
}
}
```

TESTING

Assemble two transceiver modules and connect each of them, using 3-core cables, to Com-1 ports of the two PCs. Place them 15 to 20 cms apart so that the IR LEDs of each module face the photodiode detector of the other.

Power 'on' both the circuits to operate at stabilized 5V DC. You may alternatively use a 7805 regulator IC with a 9V DC source to obtain regulated 5V supply.

Check if the MAX232 IC is working properly by testing pin-2 for 9 to 10V positive supply and pin 6 for –9V supply. MAX232 used 1µF, 25V capacitors C1-C5 as a charge pump to internally generate ±9V from 5V supply. Generally, defective MAX232 ICs will not show a voltage generation of +9V and –9V at pins 2 and 6, respectively. Replace ICs, if required. Although 1µF, 25V capacitors are recommended in the datasheet, the circuit works well even with 10µF, 25V capacitors, which are easily available.

With both the PCs and supply to the transceiver modules 'on', throw some light with the torch on the photodiode. LED1 should flicker at the burst frequency rate of the transmitter. This proves that the IR signals are being detected by photodiodes and converted into RS232-compatible levels by the MAX232 and output at pin 7 of MAX232 ICs is available for the PC to read the pulses.

To test the transmitter side, disconnect the module from COM-1 (or COM-2) port of the PC, and with the device powered 'on', use a short jumper wire from +5V and touch it at pin 8 of MAX232 IC to simulate a positive pulse. LED2 should turn 'off' and IR LEDs and LED 3 should turn 'on' if the wiring is correct. IR LEDs would also be glowing, although one cannot see them glowing. Remove the link wire from +5V to pin 8 of MAX232 IC and connect back the 'D' connector to PC's COM-1 (or COM-2) port.

Run simple communication software like PROCOM or TELIX. Set the baud rate, parity, bits per character, and stop bits to 110, n, 8, 1, respectively, and send a few characters from the keyboard through COM-1 port. You should be able to see LED3 flickering for a few seconds, indicating data transmission.

Connect both PCs to the circuits and set the software to chat mode. You should be able to transfer data between the PCs, as if a cable was connected.
Depending on the sensitivity setting and power/angle of IR LEDs, increase the distance to about 35 cms (12 inches) and try again for better distances.

For more power, use metal-can type IR LEDs and reduce the value of resistor R7 for more drive current. If you use a laser beam, remove the IR LEDs and the device will track up to 10 metres without any data loss.

Password retrieval mechanism

The above was the first part of our project. To first make a connection between transmitter and receiver, to send a data by transmitter circuit and receive the same data by receiver circuit. Now what if any unauthorized person is sitting at the receiver side? He/she would have access to all our highly confidential data. To prevent this from happening we came across a thought of securing our data so that only the intended person is able to receive the intended data and not just any person. We now for securing our transmission have designed a password circuit. The password circuit has 8051 along with 4 push button and a reset push button and a transistor based switch. The password based circuit is connected at the receiver side. We also designed the coding program for the 8051 using "C" and configured it with the hardware. Now we come on the actual testing process. We send the data from the transmitter, at the receiver side we now first enter the correct confidential password known only to the intended person and no one else. Once the correct password is entered, the 8051 activates and the transistor switch is "ON" and the data is correctly received at the receiving PC. But when the incorrect password is entered, we see there is no data received and nothing appears on the receiving PC.

HINTS

- Aligning the laser beam is a problem, but once it is aligned carefully and fixed, the data transmission and reception would be error-free. Transmitter and receiver alignment routines have been included in this software program to aid in the alignment process.

- Ordinary clear photodiodes should be used for detector. If you use dark-red plastic-encapsulated diodes, you may have problems, as these react only to very bright natural light or infrared light.

INSTRUCTIONS FOR SETUP

1) First of all, restart the computer and open the Bios Setup option by pressing ‘Del’ key before system boot. Now set the Serial Port settings (in Win98 – Go to Integrated Peripheral option and in WinXP – go to advanced option) as given below: -

   Onboard Serial Port 1 – 3F8/IRQ4
   Onboard Serial Port 2 – 2F8/IRQ3

   Save the above settings, and allow the system to boot.

2) Now set the baud rate.

   In WinXP: -

   Go to My computer → Properties → Hardware → Device Manager → Open the Port option (Com & LPT) → Select the Communication Port (Com1) → Properties → Port Settings → Set Bits Per Second – 110, Data bits – 8, Parity – None, Stop bits – 1, Flow control – Hardware → Press ‘OK’.

3) To test the communication between two computers convert a standard serial to serial communication cable as cross link (pin 2 of one connector is connected to pin 3 of the other and vice versa. Also pin 5 of both connectors is connector together as it provides ground connections).

4) Connect both the computers initially with wire and after that the desired RF or IR based transmitter receiver modules.

5) Now run the program in Turbo C and select the option – Receive Mode or Send mode and send few characters from send mode (system 1) and receive the same in system 2.

6) If you are receiving the characters comfortably at the other systems, this means now you can send a text file from one computer to another.

CONCLUSION

As we have come to the end of our project, we now have the full confidence and pride to say that we have crossed the finish line. In these rigorous months of making the project, we have expanded our knowledge and ideas to new verticals. We are very thankful to ourselves for our commitment and sincerity to our project. With the completion, we would like to share our credits with our beloved mentor Prof. V.K. Sharma for his unwavering support and guidance. We would also want to extend our warm gratitude to Mr. Ashish Majoka for providing us the requirements of our project.

As we have made our project come to life, which was once just a distant imagination, we have come to know the insights of the optical communication. We also had the opportunity to know each minute detail of the components. We also had the bite of soldering lessons, in which we were just poorly skilled. Now we have come to know the working of PCB (printed-circuit board) which was once looked like a green cardboard, that PCB which is the fundamental of every working circuit. The transfer of data which needed to be synchronized was bit a tiresome task. We have devoured many hours in understanding each and every component of the circuit and its working.

The project which was supposed to be work on Optical fiber was performed on the basis of PC to PC communication, which works on the basic principle of Optical fiber Communication. We have hinged on this platform, since we were not properly equipped to work on the optical fiber and also due to its complexity. Nevertheless we are not disheartened and we are really proud of the concept of the project. The project PC to PC communication was programmed on the platform of C language. Where the ports of computer were programmed with C. the transmitter which has a LED is matched with the receiver using the cable which is then converted from the mainstream TTL logic to RS 232 for the serial port at the receiving end and the computer. The password which is collected from the user activates the microcontroller only if the correct password is retrieved. Then the receiver circuit is ultimately activated. With this concept, we can send many as files as we want without worrying about its security and protection. Although the completion of the project was on time, we faced quite hindrances in maintaining the synchronization of the data transfer.

As with this project as the ultimatum of all, our institute has taught us to work as a team member to appreciate each other work inspite of having petty differences in times. It has inspired us to work as an individual in a team which would really help us in our upcoming professional jobs.

APPENDICES

PROBLEMS FACED AND TROUBLESHOOTING

While testing, we did face problems with red plastic-encapsulated diodes as well as clear Darlington detectors probably because of various light sources in the room causing corruption of the data. Finally, we succeeded, after modification of the circuit as shown in Fig.3. We were able to flawlessly transfer files, from about 5-meter distance, between two 386-based PCs. We included a 38 KHz modulator in the transmitter section and used IR receiver module, which includes a bandpass filter and demodulator for 38KHz carrier.

The base addresses for the serial communication ports in a PC are shown in Table III. The offset address of the registers used in serial communication is given in Table IV.

For serial port initialization, the program makes use of BIOS interrupt 14H service 00H. It initializes the serial port pointed to by the contents of dx register. The contents of ‘al’ register initialize the specific communication port for baud rate, parity, stop-bit code, and character-size code as per Table V (and expanded in Tables VI through IX respectively).

The transmitter holding register and receiver data register both at address Base+0 (the former being write (only) and latter being read (only)) act as buffers during transmission and reception, respectively, of a character. The other most important register, which is referred to in the software program frequently, is the line status register at Base+5. Meaning of each of the bits of line status register is given in Table X. Its bit 0 is set when a byte is logged in the receiver buffer register and cleared when the byte is read by the CPU. Its bit 6 is set when both the transmitter holding register and the transmitter shift register are empty.

Presently, the software program is meant for COM-1 port initialized for 110 bauds. It can be changed for 1200, or 2400, or 4800, etc by changing the contents of ‘al’ register in the initialization function to 83H, or A3H, or C3H, etc in place of 63H. Similarly, for using COM-2 port, change all register addresses starting with OX3f… to OX2f... etc. in the program.

TABLE III

Base Address for the Communication Ports

Communication	Base Address Port
COM1	03F8H
COM2	02F8H
COM3	03F8H
COM4	02F8H

TABLE IV

8250 Registers: Offset from Base Address

Offset	LCR Bit 7	Meaning	Read/write
0	0	Transmitter holding register (THR) (When written to port)	Write
0	0	Receiver data register (RDR) (When read from port)	Read
0	1	Baud rate divisor—low byte (BRDL)	Read/write
1	0	Interrupt enable register (IER)	Read/write
1	1	Baud rate divisor—high byte (BRDL)	Read/write
2	x	Interrupt identification register (IIR)	Read only
3	x	Line control register (LCR)	Read/write
4	x	Modem control register (MCR)	Read/write
5	x	Line status register (LSR)	Read only
6	x	Modem status register (MSR)	Read only

TABLE V

AL Register Bits

Bit 7	6	5	4	3	2	1	0	Use
X	X	X	●	●	●	●	●	Baud-rate code
●	●	●	X	X	●	●	●	Parity code
●	●	●	●	●	X	●	●	Stop-bit code
●	●	●	●	●	●	X	X	Character-size code

TABLE VI

Baud Rate

Bit 7	6	5	Value	Bits Per Second
0	0	0	0	110
0	0	1	1	150
0	1	0	2	300
0	1	1	3	600
1	0	0	4	1200
1	0	1	5	2400
1	1	0	6	4800
1	1	1	7	9600

TABLE VII

Parity

Bit 4	Bit 3	Value	Meaning
0	0	0	None
0	1	1	Odd Parity
1	0	2	None
1	1	3	Even Parity

TABLE VIII

Stop Bits

Bit 2	Value	Meaning
0	0	One
0	1	Two

TABLE IX

Character Size

Bit 1	Bit 0	Value	Meaning
0	0	0	Not used
0	1	1	Not used
1	0	2	7-Bit*
1	1	3	8-Bit

REFERENCES

Related websites:

www.google.com

www.electronicsforyou.com

www.epanorama.net

www.fundoo.net

www.datasheetcatalogue.com

Related books:

D.V.Hall

“8051 and embedded system” by Mazidi and Mazidi

“LET US C” by Yashwant Kanitkar