A Dissertation Report On

"Under Ground Cable Fault Detector Using GSM "

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

by

BANGI AQDAS

SHAIKH SHAHIR

SHIAKH TARIK

KHOT KHIZAR

Under the Guidance of

Prof.Ankur Upadhya

Anjuman Islam's kalsekar Technical Campus of Engineering

Affiliated to

UNIVERSITY OF MUMBAI

Anjuman-I-Islam's
Kalsekar Technical Campus

CERTIFICATE

This is to certify that project entitled on "Under Ground Cable Fault Detector Using GSM " is bonafide work of

BANGI AQDAS

SHAIKH SHAHIR

SHIAKH TARIK

KHOT KHIZAR

of B.E. (Electronics Engineering) or diploma (Sem-VIII) class and is submitted to the Mumbai University/MSBTE , Mumbai in partial fulfillment of the requirement for the degree/diploma of Bachelor of Engineering in Electronics Engineering. The project report has been approved.

Prof **(Internal Guide)**	**Mr. (External Guide)**

DISSERTATION APPROVAL FOR BACHELOR OF ENGINEERING

This dissertation report entitled "Under Ground Cable Fault Detector Using GSM " approved

for the degree of Bachelor of Electrical Engineering

Management) for the academic year 2016-17.

Examiners:

1. ____________________

2. ____________________

Guide:

1. ____________________

Date:

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

ACKNOWLEDGMENT

I would like to express my gratitude to my Project Guide Prof, Ankur Upadhaya for guidance and support throughout this thesis work. Both have been a

constant source of inspiration to me throughout the period of this work. I consider myself

extremely fortunate for having the opportunity to learn and work under their supervision over

the entire period.

I would also take this opportunity to express my gratitude and sincere thanks to all the

staffs for valuable support.

ABSTARCT

Many electricity transmission companies across the world and Ghana in particular are continuously looking for ways to utilize modern technologies, in order to improve reliability of power supply to consumers. These transmission companies manly relies on circuit indicators (FCIs) to assist in locating specific spots within their transmission lines where power fault had occured.

In this paper, a smart GSM based fault detection and location system was used to adequately and accurately indicate and locate the exact spot where fault had occured. This will ensure a shorter response time for technical crew to rectify these faults and thus help save transformers from damage and disasters. The system uses a current transformer, a voltage transformer, 8051 Series Micro-controller, RS-232 connector, and a GSM modem. The system automatically detects faults, analyses and classifies these faults and then, calculates the fault distance from the control room using an impedance-based algorithm method. Finally the fault information is transmitted to the control room. In conclusion, the time required to locate a fault is drastically reduced, as the system automatically and accurately provides accurate fault location information.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

1.1 INTRODUCTION

Many electric power transmission companies such as Ghana Grid Company limited (GRIDco) in Ghana, have primarily relied on circuit indicators to detect faulty sections of their transmission lines. However there are still challenges in detecting the exact location of these faults. Although fault indicator technology has provided a reliable means to locate permanent faults, the technical crew and patrol teams still has to physically patrol and inspect the devices for longer hours to detect faulty sections of their transmission lines.

1.2 CONCEPT OF THE PROJECT

The proposed system is intended to automatically detect faults when they occur, analyse the fault to determine the type and then send information based on the fault type and fault location to the control room via GSM. The device location is determined by the SIM card in the modem, each SIM card having a unique identification an hence is used as the device's address.

The system senses, analyses and transmits. It does this with the micro-controller which analyses, interprets and sends digital signals to the I/O devices for the system to operate. By programming, the micro-controller is made to perform these functions.

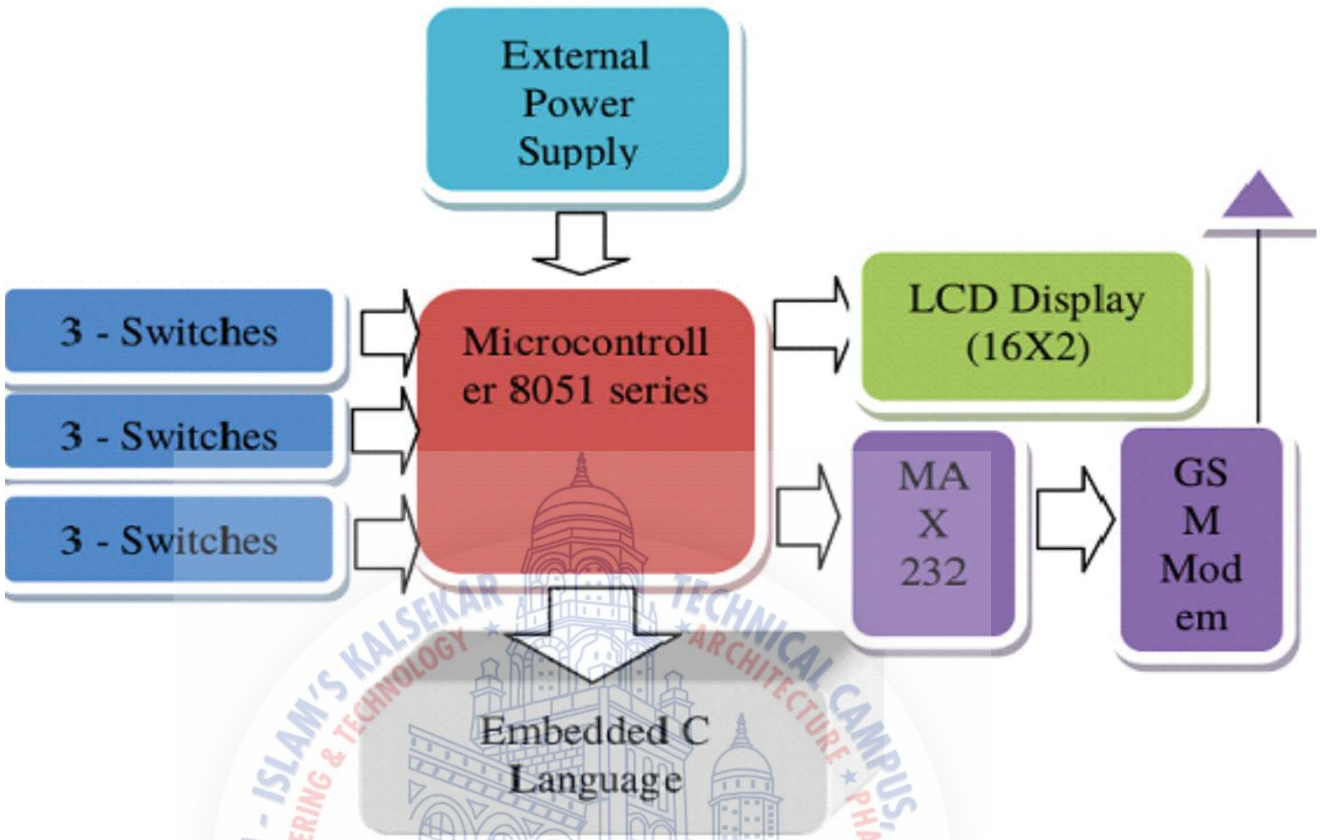
CHAPTER 2

2.1 LITARTURE SURVEY

The hardware model of Underground Cable Fault Locator is implemented and favorable results were brought forward. This hardware model can locate the exact fault location in an underground cable.

Further this project can be enhanced by using capacitor in an AC circuit to measure the impedance which can even locate the open circuited cable, unlike the short circuited

fault only using resistors in DC circuit as followed in the above proposed project.



BLOCK DIAGRAM OF UNDERGROUND CABLE FAULT DETECTOR USING GSM

FIG 2.1

2.2 HARDWARE

Step Down Transformer

Bridge Rectifier

Capacitors

Voltage Regulator IC

Micro-Controller

MAX 232

GSM Modem

LCD

2.3 SOFTWARE

Compiler:-Keil Compiler

Language:-Embedded C

2.4 MICRO CONTROLLER

MICROCONTROLLER AT89S52

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

Features:

• Compatible with MCS®-51 Products

• 8K Bytes of In-System Programmable (ISP) Flash Memory

– Endurance: 10,000 Write/Erase Cycles

• 4.0V to 5.5V Operating Range

• Fully Static Operation: 0 Hz to 33 MHz

• Three-level Program Memory Lock

• 256 x 8-bit Internal RAM

• 32 Programmable I/O Lines

• Three 16-bit Timer/Counters

• Eight Interrupt Sources

• Full Duplex UART Serial Channel

• Low-power Idle and Power-down Modes

• Interrupt Recovery from Power-down Mode

• Watchdog Timer

• Dual Data Pointer

• Power-off Flag

• Fast Programming Time

• Flexible ISP Programming (Byte and Page Mode)

• Green (Pb/Halide-free) Packaging Option

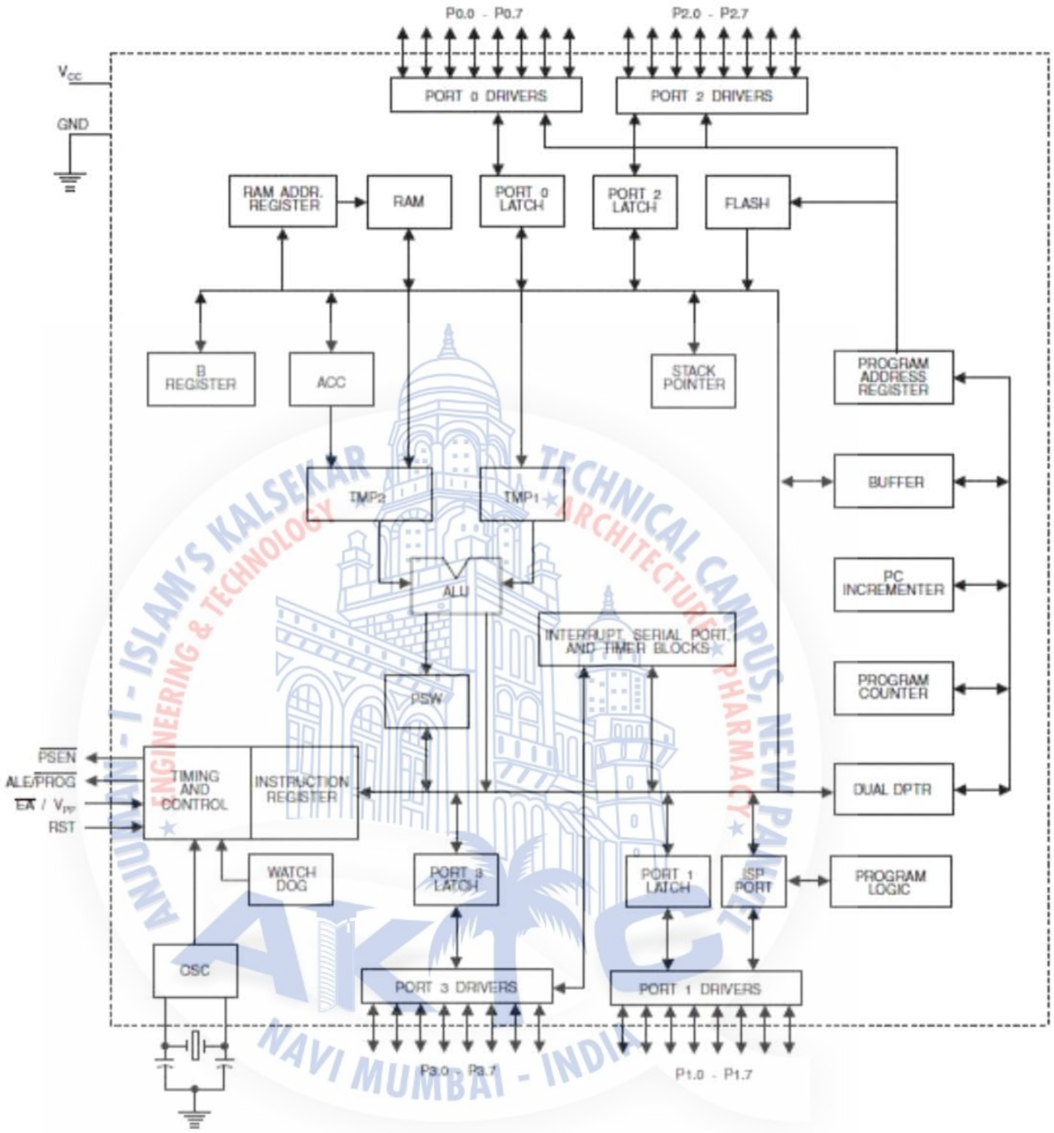


FIG 2.2 MICROCONTOLLER AT8952S

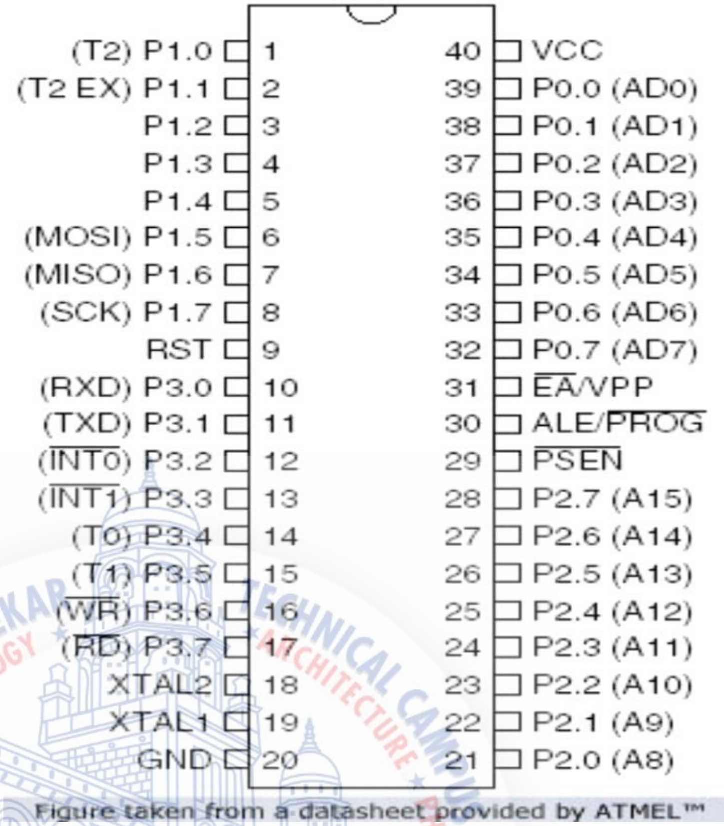


FIG 2.3 PIN CONFIGURATION OF MICRO-CONTROLLER

Pin Description:

VCC:

Supply voltage.

GND:

Ground.

Port 0:

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

Port 1:

Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX).

Port 2:

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

Port 3:

Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull-ups.

RST:

Reset input. 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 DISRTO 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.

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.

PSEN:

Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S52 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/VPP:

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

XTAL1:

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

XTAL2:

Output from the inverting oscillator amplifier.

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, as shown in Figure 1. Either a quartz crystal or 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 in Figure 6.2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

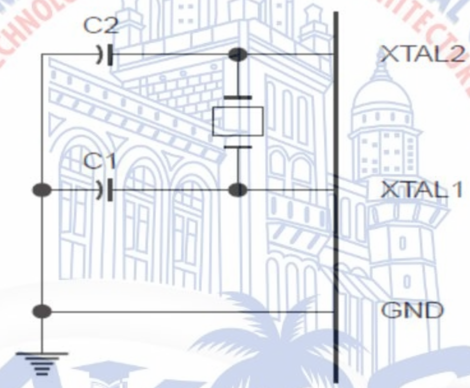


FIG 2.4 OSCILLATOR SIGNAL

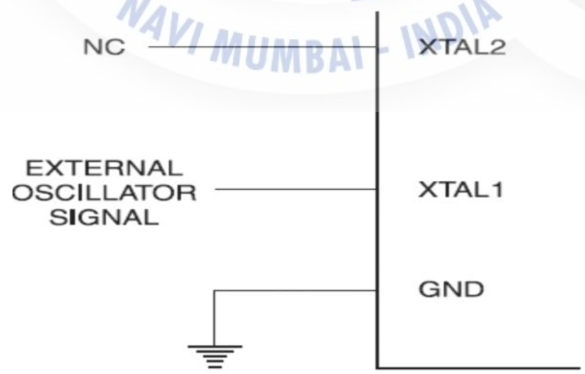


FIG 2.5 POWER MODE

Power down Mode

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

2.5 POWER SUPPLY:-

In this project circuits, sensors & motor are used which require +12V & +5V(DC)supply, to fulfill this requirement we have used following circuit of power supply which provides regulated +12V & +5V.(DC)

Power Supplies

Types of Power Supply

There are many types of power supply. Most are designed to convert high voltage AC mains electricity to a suitable low voltage supply for electronics circuits and other devices. A power supply can by broken down into a series of blocks, each of which performs a particular function.

For example a 5V regulated supply:

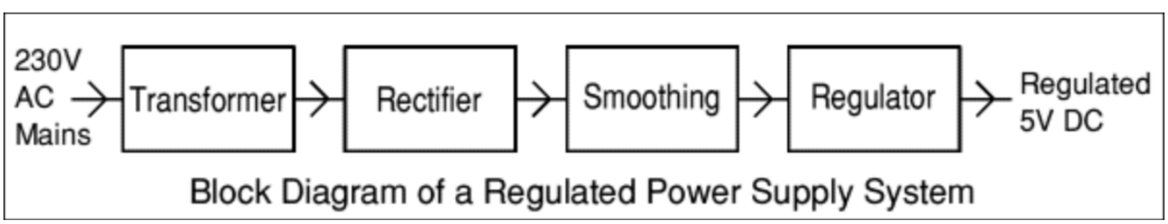


FIG 2.5

Each of the blocks is described in more detail below:

- Transformer - steps down high voltage AC mains to low voltage AC.
- Rectifier - converts AC to DC, but the DC output is varying.
- Smoothing - smooths the DC from varying greatly to a small ripple.
- Regulator - eliminates ripple by setting DC output to a fixed voltage.

Power supplies made from these blocks are described below with a circuit diagram and a graph of their output:

- Transformer only
- Transformer + Rectifier
- Transformer + Rectifier + Smoothing
- Transformer + Rectifier + Smoothing + Regulator

The Power Supply

You will definitely need a **regulated 5-volt power supply** to work with TTL chips. As mentioned previously, neither Radio Shack nor Jameco seem to offer a standard, inexpensive 5-volt regulated power supply. One option you have is to buy from Jameco something like part number 116089. This is a 5-volt power supply from an old Atari video game. If you look in the Jameco catalog, you will find that they have about 20 different **surplus power supplies** like this, producing all sorts of voltages and amperages. You need **5 volts** at **at least 0.3 amps** (300 milliamps) -- you need no more than 2 amps, so do not purchase more power supply than you need. What you can do is buy the power supply, then cut off the connector and get access to the 5-volt and ground wires. That will work fine, and is probably the easiest path. You can use your volt meter (see below) to make sure the power supply produces the voltage you need.

Your alternative is to build a 5-volt supply from a little **power-cube transformer**. What you need is a transformer that produces **7 to 12 DC volts** at **100 milliamps or more**. Note that:

- The transformer MUST produce DC voltage.
- It MUST produce 7 to 12 volts.
- It MUST produce 100 milliamps (0.1 amps) or more.

Dual Supplies

Some electronic circuits require a power supply with positive and negative outputs as well as zero volts (0V). This is called a 'dual supply' because it is like two ordinary supplies connected together as shown in the diagram.

Dual supplies have three outputs, for example a ±3.3V supply

2.6 KIEL COMPILER

SOFTWARE REQUIREMENT

INTRODUCTION TO KEIL MICRO VISION (IDE)

Keil an ARM Company makes C compilers, macro assemblers, real-time kernels, debuggers, simulators, integrated environments, evaluation boards, and emulators for ARM7/ARM9/Cortex-M3, XC16x/C16x/ST10, 251, and 8051 MCU families.

Keil development tools for the 8051 Microcontroller Architecture support every level of software developer from the professional applications engineer to the student just learning about embedded software development. When starting a new project, simply select the microcontroller you use from the Device Database and the µVision IDE sets all compiler, assembler, linker, and memory options for you.

Keil is a cross compiler. So first we have to understand the concept of compilers and cross compilers. After then we shall learn how to work with keil.

CONCEPT OF COMPILER

Compilers are programs used to convert a High Level Language to object code. Desktop compilers produce an output object code for the underlying microprocessor, but not for other microprocessors. I.E the programs written in one of the HLL like 'C' will compile the code to run on the system for a particular processor like x86 (underlying microprocessor in the computer). For example compilers for Dos platform is different from the Compilers for Unix platform So if one wants to define a compiler then compiler is a program that translates source code into object code.

The compiler derives its name from the way it works, looking at the entire piece of source code and collecting and reorganizing the instruction. See there is a bit little difference between compiler and an interpreter. Interpreter just interprets whole program at a time while compiler analyses and execute each line of source code in succession, without looking at the entire program.

The advantage of interpreters is that they can execute a program immediately. Secondly programs produced by compilers run much faster than the same programs executed by an interpreter. However compilers require some time before an executable program emerges. Now as compilers translate source code into object code, which is unique for each type of computer, many compilers are available for the same language.

CONCEPT OF CROSS COMPILER

A cross compiler is similar to the compilers but we write a program for the target processor (like 8051 and its derivatives) on the host processors (like computer of x86). It means being in one environment you are writing a code for another environment is called cross development. And the compiler used for cross development is called cross compiler. So the definition of cross compiler is a compiler that runs on one computer but produces object code for a different type of computer.

KEIL C CROSS COMPILER

Keil is a German based Software development company. It provides several development tools like

- IDE (Integrated Development environment)
- Project Manager
- Simulator
- Debugger
- C Cross Compiler, Cross Assembler, Locator/Linker

The Keil ARM tool kit includes three main tools, assembler, compiler and linker. An assembler is used to assemble the ARM assembly program. A compiler is used to compile the C source code into an object file. A linker is used to create an absolute object module suitable for our in-circuit emulator.

Building an Application in µVision2

To build (compile, assemble, and link) an application in µVision2, you must:

Select Project -(forexample,166\EXAMPLES\HELLO\HELLO.UV2).

Select Project - Rebuild all target files or Build target.µVision2 compiles, assembles, and links the files in your project.

Creating Your Own Application in µVision2

To create a new project in µVision2, you must:

Select Project - New Project.

Select a directory and enter the name of the project file.

Select Project - Select Device and select an 8051, 251, or C16x/ST10 device from the Device Database™.

Create source files to add to the project.

Select Project - Targets, Groups, Files. Add/Files, select Source Group1, and add the source files to the project.

Select Project - Options and set the tool options. Note when you select the target device from the Device Database™ all special options are set automatically. You typically only need to configure the memory map of your target hardware. Default memory model settings are optimal for most applications.

Select Project - Rebuild all target files or Build target.

Debugging an Application in µVision2

To debug an application created using µVision2, you must:

Select Debug - Start/Stop Debug Session.

Use the Step toolbar buttons to single-step through your program. You may enter G, main in the Output Window to execute to the main C function.

Open the Serial Window using the Serial #1 button on the toolbar.

Debug your program using standard options like Step, Go, Break, and so on.

Starting µVision2 and Creating a Project

µVision2 is a standard Windows application and started by clicking on the program icon. To create a new project file select from the µVision2 menu Project – New Project…. This opens a standard Windows dialog that asks you for the new project file name. We suggest that you use a separate folder for each project. You can simply use the icon Create New Folder in this dialog to get a new empty folder. Then select this folder and enter the file name for the new project, i.e. Project1. µVision2 creates a new project file with the name PROJECT1.UV2 which contains a default target and file group name. You can see these names in the Project.

Window – Files.

Now use from the menu Project – Select Device for Target and select a CPU for your project. The Select Device dialog box shows the µVision2 device data base. Just select the microcontroller you use. We are using for our examples the Philips 80C51RD+ CPU. This selection sets necessary tool Options for the 80C51RD+ device and simplifies in this way the tool Configuration.

Building Projects and Creating a HEX Files

Typical, the tool settings under Options – Target are all you need to start a new application. You may translate all source files and line the application with a click on the Build Target toolbar icon. When you build an application with syntax errors, µVision2 will display errors and warning messages in the Output Window – Build page. A double click on a message line opens the source file on the correct location in a µVision2 editor window. Once you have successfully generated your application you can start debugging.

After you have tested your application, it is required to create an Intel HEX file to download the software into an EPROM programmer or simulator. µVision2 creates HEX files with each build process when Create HEX files under Options for Target – Output is enabled. You may start your PROM programming utility after the make process when you specify the program under the option Run User Program #1.

CPU Simulation

µVision2 simulates up to 16 Mbytes of memory from which areas can be mapped for read, write, or code execution access. The µVision2 simulator traps
and reports illegal memory accesses. In addition to memory mapping, the simulator also provides support for the integrated peripherals of the various 8051 derivatives. The on-chip peripherals of the CPU you have selected are configured from the Device.

Database selection

You have made when you create your project target. Refer to page 58 for more Information about selecting a device. You may select and display the on-chip peripheral components using the Debug menu. You can also change the aspects of each peripheral using the controls in the dialog boxes.

Start Debugging

You start the debug mode of µVision2 with the Debug – Start/Stop Debug Session Command. Depending on the Options for Target – Debug Configuration, µVision2 will load the application program and run the startup code µVision2 saves the editor screen layout and restores the screen layout of the last debug session. If the program execution stops, µVision2 opens an editor window with the source text or shows CPU instructions in the disassembly window. The next executable statement is marked with a yellow arrow. During debugging, most editor features are still available.

For example, you can use the find command or correct program errors. Program source text of your application is shown in the same windows. The µVision2 debug mode differs from the edit mode in the following aspects:

_ The "Debug Menu and Debug Commands" described on page 28 are available. The additional debug windows are discussed in the following.

_ The project structure or tool parameters cannot be modified. All build commands are disabled.

Disassembly Window

The Disassembly window shows your target program as mixed source and assembly program or just assembly code. A trace history of previously executed instructions may be displayed with Debug – View Trace Records. To enable the trace history, set Debug – Enable/Disable Trace Recording.

If you select the Disassembly Window as the active window all program step commands work on CPU instruction level rather than program source lines. You can select a text line and set or modify code breakpoints using toolbar buttons or the context menu commands.

You may use the dialog Debug – Inline Assembly… to modify the CPU instructions. That allows you to correct mistakes or to make temporary changes to the target program you are debugging. Numerous example programs are included to help you get started with the most popular embedded 8051 devices.

The Keil µVision Debugger accurately simulates on-chip peripherals (I²C, CAN, UART, SPI, Interrupts, I/O Ports, A/D Converter, D/A Converter, and PWM Modules) of your 8051 device. Simulation helps you understand hardware configurations and avoids time wasted on setup problems. Additionally, with simulation, you can write and test applications before target hardware is available.

EMBEDDED C

Use of embedded processors in passenger cars, mobile phones, medical equipment, aerospace systems and defense systems is widespread, and even everyday domestic appliances such as dish washers, televisions, washing machines and video recorders now include at least one such device.

Because most embedded projects have severe cost constraints, they tend to use low-cost processors like the 8051 family of devices considered in this book. These popular chips have very limited resources available most such devices have around 256 bytes (not megabytes!) of RAM, and the available processor power is around 1000 times less than that of a desktop processor. As a result, developing embedded software presents significant new challenges, even for experienced desktop programmers. If you have some programming experience - in C, C++ or Java - then this book and its accompanying CD will help make your move to the embedded world as quick and painless as possible.

CHAPTER 3

ADOPTED METHODOLOGY

TRANSFORMERS AND RECTIFIERS

3.1 TRANSFORMERS

Transformer only

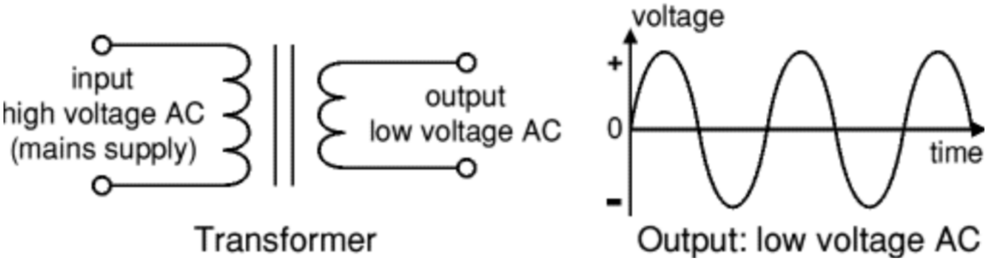


FIG 3.1

The **low voltage AC** output is suitable for lamps, heaters and special AC motors. It is **not** suitable for electronic circuits unless they include a rectifier and a smoothing capacitor.

Transformer + Rectifier

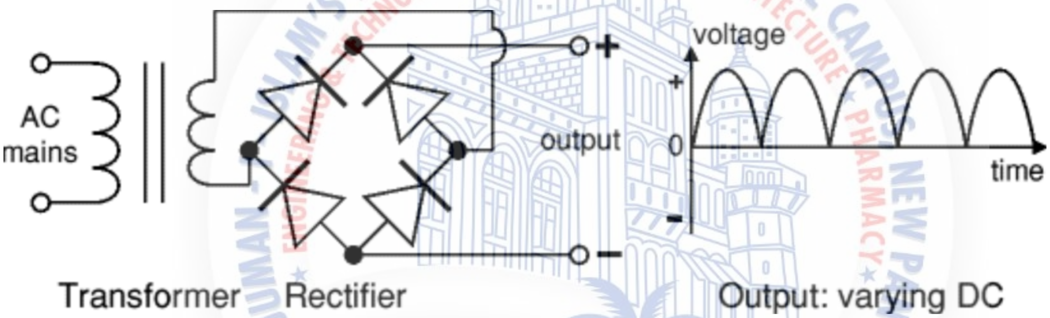


FIG 3.2

The **varying DC** output is suitable for lamps, heaters and standard motors. It is **not** suitable for electronic circuits unless they include a smoothing capacitor.

Transformer + Rectifier + Smoothing

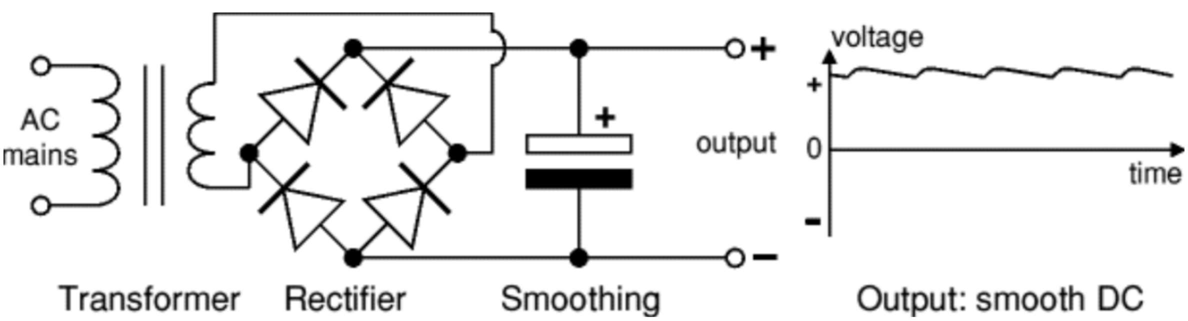


FIG 3.3

The **smooth DC** output has a small ripple. It is suitable for most electronic circuits.

Transformer + Rectifier + Smoothing + Regulator

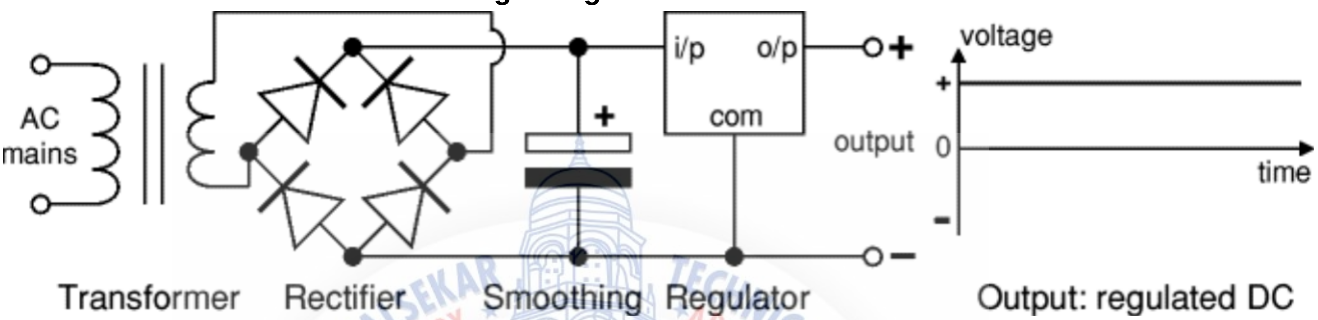


FIG 3.4

The **regulated DC** output is very smooth with no ripple. It is suitable for all electronic circuits.

Transformer

Transformer
circuit symbol

FIG 3.5
A TYPICAL TRANSFORMER

Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC.

Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in UK) to a safer low voltage.

The input coil is called the **primary** and the output coil is called the **secondary**. There is no electrical connection between the two coils, instead they are linked by an alternating magnetic field created in the soft-iron core of the transformer. The two lines in the middle of the circuit symbol represent the core.

Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up.

The ratio of the number of turns on each coil, called the **turns ratio**, determines the ratio of the voltages. A step-down transformer has a large number of turns on its primary (input) coil which is connected to the high voltage mains supply, and a small number of turns on its secondary (output) coil to give a low output voltage.

turns ratio = $\frac{Vp}{Vs} = \frac{Np}{Ns}$	and	power out = power in $Vs \times Is = Vp \times Ip$	

Vp = primary (input) voltage Np = number of turns on primary coil Ip = primary (input) current	Vs = secondary (output) voltage Ns = number of turns on secondary coil Is = secondary (output) current

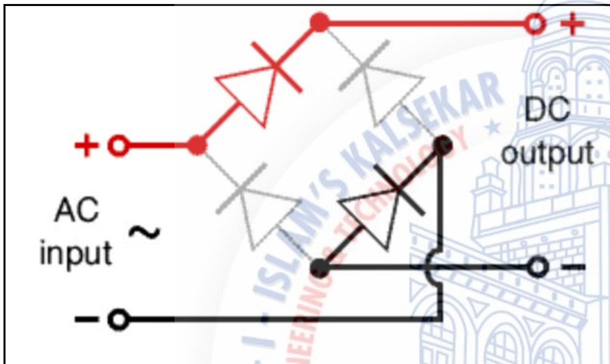
3.2 Rectifier

There are several ways of connecting diodes to make a rectifier to convert AC to DC. The bridge rectifier is the most important and it produces **full-wave** varying DC. A full-wave rectifier can also be made from just two diodes if a centre-tap transformer is used, but this method is rarely used now that diodes are cheaper. A single diode can be used as a rectifier but it only uses the positive (+) parts of the AC wave to produce **half-wave** varying DC.

Bridge rectifier

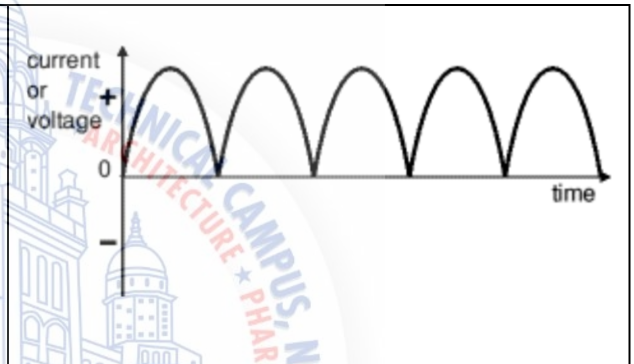
A bridge rectifier can be made using four individual diodes, but it is also available in special packages containing the four diodes required. It is called a full-wave rectifier because it uses all the AC wave (both positive and negative sections). 1.4V is used up in the bridge rectifier because each diode uses 0.7V when conducting and there are always two diodes conducting, as shown in the diagram below. The maximum current they can pass rates bridge rectifiers and the maximum reverse voltage they can withstand (this must be at least three times the supply RMS voltage so the rectifier can withstand the peak voltages). Please see the Diodes page for more details, including pictures of bridge rectifiers.

FIG 3.6 BRIDGE RECTIFIER



Bridge rectifier

Alternate pairs of diodes conduct, changing over the connections so the alternating directions of AC are converted to the one direction of DC.

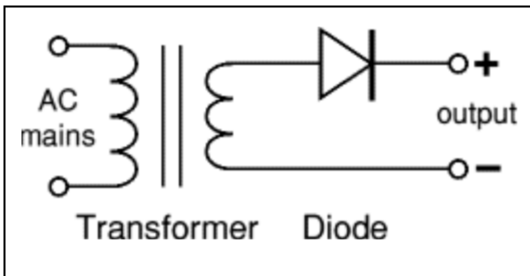


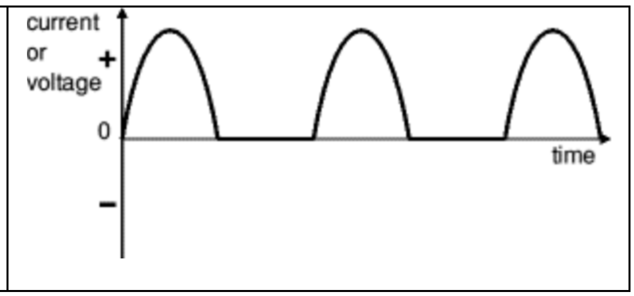
Output: full-wave varying DC

(using all the AC wave)

Single diode rectifier

A single diode can be used as a rectifier but this produces **half-wave** varying DC which has gaps when the AC is negative. It is hard to smooth this sufficiently well to supply electronic circuits unless they require a very small current so the smoothing capacitor does not significantly discharge during the gaps. Please see the Diodes page for some examples of rectifier diodes.





Single diode rectifier	**Output: half-wave varying DC** (using only half the AC wave)

FIG 3.7 SINGLE DIODE RECTIFIER

3.3 Smoothing

Smoothing is performed by a large value electrolytic capacitor connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The diagram shows the unsmoothed varying DC (dotted line) and the smoothed DC (solid line). The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output.

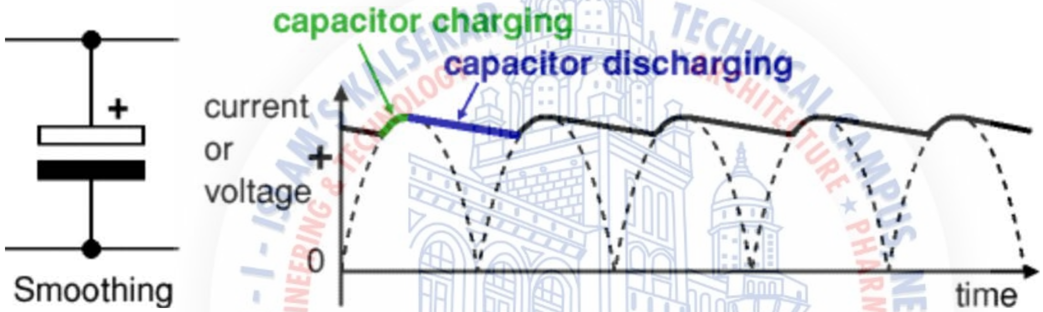


FIG 3.8 SMOOTHING

Note that smoothing significantly increases the average DC voltage to almost the peak value (1.4 × RMS value). For example 6V RMS AC is rectified to full wave DC of about 4.6V RMS (1.4V is lost in the bridge rectifier), with smoothing this increases to almost the peak value giving 1.4 × 4.6 = 6.4V smooth DC.

Smoothing is not perfect due to the capacitor voltage falling a little as it discharges, giving a small **ripple voltage**. For many circuits a ripple which is 10% of the supply voltage is satisfactory and the equation below gives the required value for the smoothing capacitor. A larger capacitor will give less ripple. The capacitor value must be doubled when smoothing half-wave DC.

Smoothing capacitor for 10% ripple, $C = \frac{5 \times Io}{Vs \times f}$

C = smoothing capacitance in farads (F)
Io = output current from the supply in amps (A)
Vs = supply voltage in volts (V), this is the peak value of the unsmoothed DC
f = frequency of the AC supply in hertz (Hz), 50Hz in the UK

3.4 Regulator

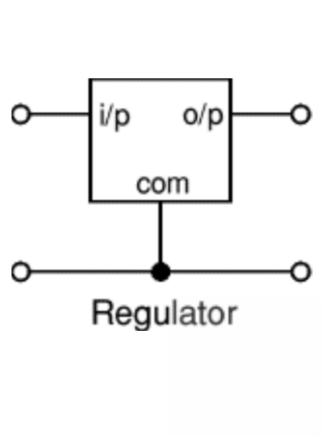
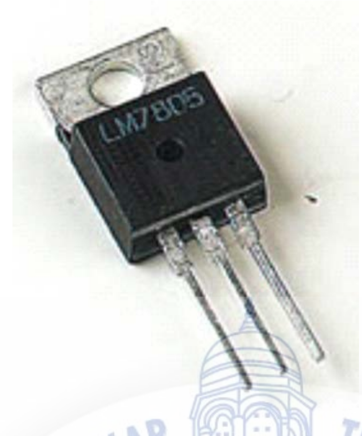


FIG 3.9



REGULATOR

Voltage regulator ICs are available with fixed (typically 5, 12 and 15V) or variable output voltages. They are also rated by the maximum current they can pass. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current ('overload protection') and overheating ('thermal protection').

Many of the fixed voltage regulator ICs have 3 leads and look like power transistors, such as the 7805 +5V 1A regulator shown on the right. They include a hole for attaching a heatsink if necessary.

CHAPTER 4

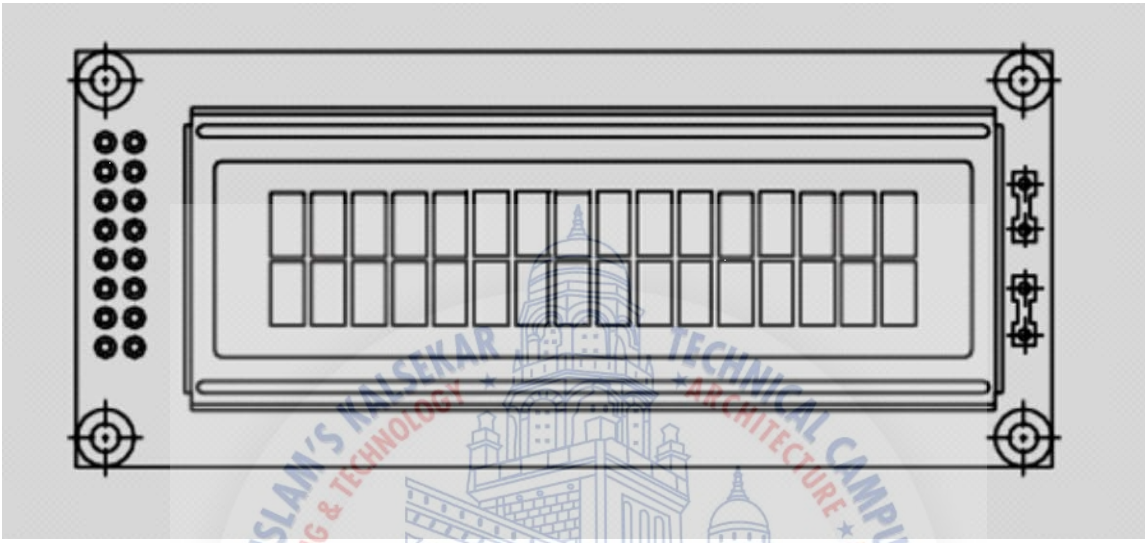
LCD ,PCB LAYOUT, PROGRAMING AND TESTINGS

4.1 LCD

LCD DISPLAY

Various display device such as seven segment display. LCD display, etc can be interfaced with micro-controller to read the output directly. In our project we use a two line LCD display with 16 characters each.

FIG 4.1 LCD Display



FEATURES

5 x 8 dots with cursor

Built-in controller (KS 0066 or Equivalent)

+ 5V power supply

1/16 duty cycle

B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

N.V. optional for + 3V power supply

RS232 compatible serial interface (2400 & 9600 baud select-able)

Externally select-able serial polarities (Inverted & Non-Inverted)

Serially controllable contrast and backlight levels

8 user programmable custom characters

16 Byte serial receive buffer

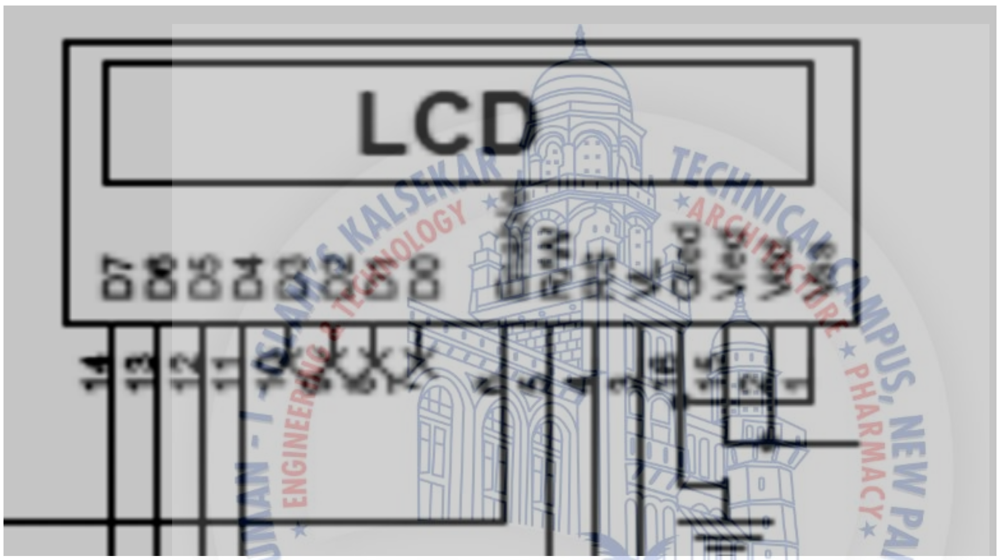


FIG 4.2 PIN DIAGRAM OF LCD

4.2 PCB MAKING

Implement of the PCB making

To make the Printed Circuit Board, following implement becomes necessary.

The way of making the Printed Circuit Board is not only the way of introducing this time.

Implement and material to use

Positive exposure printed board	The material which the ultraviolet rays change the nature to ha been painted the copper foil of the PCB. It piles the film which pictured the mask pattern on the positiv exposure printed board and it irradiates the ultraviolet rays. The part where the ultraviolet rays were irradiated changes in the quality. When putting the positive exposure printed board which finished irradiating the ultraviolet rays in the developer, the part where the ultraviolet rays were irradiated dissolves in the developer and the masked part doesn't dissolve. The part which wasn't dissolved in the developer is left as the mask pattern of the etching and doesn't dissolve in the etchant too.
Development medicine of a sensitizer	It dissolves the sensitizer in the part where the ultraviolet rays of the positive exposure printed board were irradiated.
Mask pattern	It may picture the mask pattern in the black ink at the transparent sheet. In the case of me, I picture the mask pattern drawing by the software, "Visio", and am using it for the overhead projector (OHP) sheet, printing.
Ultraviolet ray exposure equipment	It is made even if it uses the fluorescence light desk stand. Because about 20 minutes of the light were necessary, I made the ultraviolet ray exposure equipment with the timer (the fluorescence light with the timer).
Clamp	It is the implement to hold for board and the mask pattern to stick when irradiating the ultraviolet rays to the positive exposure printed board with the ultraviolet ray exposure equipment.
Resist pen	It uses for the repair of the mask pattern. It is a kind of the oily dry ink.
Etching apparatus	It is the equipment to dissolve the copper(Etching) which the printed board is unnecessary to. Even if there is not such equipment, the etchant can be dissolved by putting it in the palette to use for the developmen of the photograph, too.
Quartz pipe heater	It is the heater which raises the temperature of the etchant at 40-43 degC. In the liquid temperature, when high, the copper melts early. It says that it must not raise at equal to or more than 45 degC.
Etching liquid	It is the solution which dissolves the copper. It is the solution of the **Ferric Chloride**.
Resist solvent	It uses to remove the mask pattern of the printed board that th etching was ended.
Battery style mini drill	It makes a hole which lets through the lead line of the part. The hand drill is good but I am using the mini drill of the electric formula because the bits of the drill is thin and it is eas to break.
Flux	It paints to prevent the copper of the printed board from the oxidation.
Solder resist	It paints the wiring surface of the printed board and extra solde

	makes not be stuck.

4.3 PCB LAYOUT

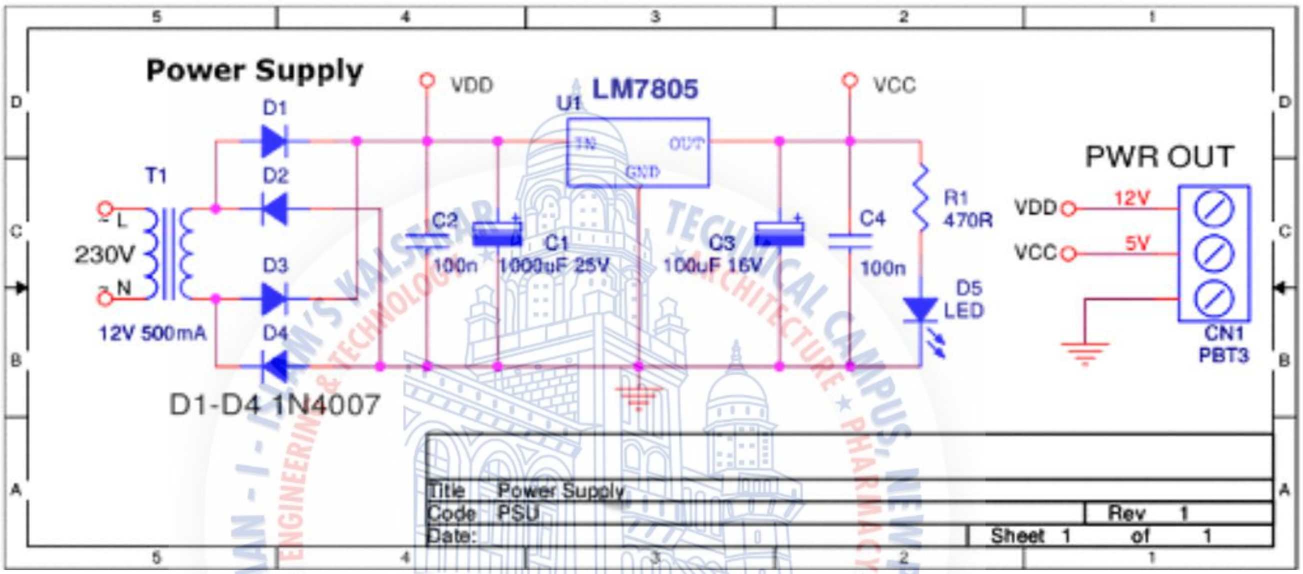


FIG 4.3 P .C.B. LAYOUT

P.C.B. is printed circuit board which is of insulating base with layer of thin copper-foil.

The circuit diagram is then drawn on the P. C. B. with permanent marker and then it is dipped in the solution of ferric chloride so that unwanted copper is removed from the P.C.B., thus leaving components interconnection on the board.

The specification of the base material is not important to know in most of the application, but it is important to know something about copper foil which is drawn through a thin slip.

The resistance of copper foil will have an affect on the circuit operation.

Base material is made of lamination layer of suitable insulating material such as treated paper, fabric; or glass fibers and binding them with resin. Most commonly used base materials are formed paper bonded with epoxy resin.

It is possible to obtain a range of thickness between 0.5 mm to 3 mm.

Thickness is the important factor in determining mechanical strength particularly when the commonly used base material is “Formed” from paper assembly.

Physical properties should be self supporting these are surface resistivity, heat dissipation, dielectric, constant, dielectric strength.

Another important factor is the ability to withstand high temperature.

DESIGNING THE LAYOUT :

While designing a layout, it must be noted that size of the board should be as small as possible.

Before starting, all components should be placed properly so that an accurate measurement of space can be made.

The component should not be mounted very close to each other or far away from one another and neither one should ignore the fact that some component reed ventilation, which considerably the dimension of the relay and transformer in view of arrangement, the bolting arrangement is also considered.

The layout is first drawn on paper then traced on copper plate which is finalized with the pen or permanent marker which is efficient and clean with etching.

The resistivity also depends on the purity of copper, which is highest for low purity of copper. The high resistance path are always undesired for soldered connections.

The most difficult part of making an original printed circuit is the conversion from, theoretical circuit diagram into wiring layout. without introducing cross over and undesirable effect.

Although it is difficult operation, it provides greatest amount of satisfaction because it is carried out with more care and skill.

The board used for project has copper foil thickness in the range of 25 40 75 microns.

The soldering quality requires 99.99% efficiency.

It is necessary to design copper path extra large. There are two main reasons for this,

The copper may be required to carry an extra large overall current:-

It acts like a kind of screen or ground plane to minimize the effect of interaction.

The first function is to connect the components together in their right sequence with minimum need for interlinking i.e. the jumpers with wire connections.

It must be noted, that when layout is done, on the next day it should be dipped in the solution and board is move continuously right and left after etching perfectly the board is cleaned with water and is drilled.

After that holes are drilled with 1 mm or 0.8 mm drill. Now the marker on the P. C. B. is removed.

The Printed Circuit Board is now ready for mounting the components on it.

SOLDERING :

For soldering of any joints first the terminal to be soldered are cleaned to remove oxide film or dirt on it. If required flux is applied on the points to be soldered.

Now the joint to be soldered is heated with the help of soldering iron. Heat applied should be such that when solder wire is touched to joint, it must melt quickly.

The joint and the soldering iron is held such that molten solder should flow smoothly over the joint.

When joint is completely covered with molten solder, the soldering iron is removed.

The joint is allowed to cool, without any movement.

The bright shining solder indicates good soldering.

In case of dry solder joint, a air gap remains in between the solder material and the joint. It means that soldering is improper. This is removed and again soldering is done.

Thus is this way all the components are soldered on P. C. B

4.4 PROGRAMING

```
#include <REGX51.H> // standard 8051 defines
#include <stdio.h> // printf
```

```
sbit BUZZER=P1^6;
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
// -=-=-=-=-    Main Program         -=-=-=-=-=-=-=
// -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=



void sendsms1()
{

        BUZZER = 1;
        printf("AT\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGF=1\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGS="9702856015");
        delayms(2000); // 2 sec delay
        delayms(1000); // 1 sec delay

        printf ("L1:5KM-FAULT HAS BEEN DETECTED KINDLY PAY
ATTENTION    2016 17 A.I.K.T.C.CLASS B.E.");

        BUZZER = 0;
```

```
}
 void sendsms2()
{

	BUZZER = 1;
	printf("AT\n");
	delayms(2000); // 2 sec delay

	delayms(1000); // 1 sec delay

	printf("AT+CMGF=1\n");
	delayms(2000); // 2 sec delay

	delayms(1000); // 1 sec delay

	printf("AT+CMGS="9702856015");
	delayms(2000); // 2 sec delay
	delayms(1000); // 1 sec delay

	printf ("L1:5KM-10KM FAULT HAS BEEN DETECTED KINDLY PAY
ATTENTION    2016 17 A.I.K.T.C.CLASS B.E.");

	BUZZER = 0;
}
void sendsms3()
{

	BUZZER = 1;
	printf("AT\n");
```

```
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGF=1\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGS="9702856015");
        delayms(2000); // 2 sec delay
        delayms(1000); // 1 sec delay

        printf ("L1:10KM-FAULT HAS BEEN DETECTED KINDLY PAY
ATTENTION    2016 17 A.I.K.T.C.CLASS B.E.");

        BUZZER = 0;
}
void sendsms4()
{

        BUZZER = 1;
        printf("AT\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGF=1\n");
        delayms(2000); // 2 sec delay
```

```
        delayms(1000); // 1 sec delay

        printf("AT+CMGS=\"9702856015");
        delayms(2000); // 2 sec delay
        delayms(1000); // 1 sec delay

        printf ("L2:5KM-FAULT HAS BEEN DETECTED KINDLY PAY
ATTENTION    2016 17 A.I.K.T.C.CLASS B.E.");

        BUZZER = 0;
}
void sendsms5()
{

        BUZZER = 1;
        printf("AT\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGF=1\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGS="9702856015");
        delayms(2000); // 2 sec delay
        delayms(1000); // 1 sec delay
```

```
        printf ("L2:5KM-10KM FAULT HAS BEEN DETECTED KINDLY PAY
ATTENTION    2016 17 A.I.K.T.C.CLASS B.E.");

        BUZZER = 0;
}
void sendsms6()
{

        BUZZER = 1;
        printf("AT\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGF=1\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGS="9702856015");
        delayms(2000); // 2 sec delay
        delayms(1000); // 1 sec delay

        printf ("L2:10KM-FAULT HAS BEEN DETECTED KINDLY PAY
ATTENTION    2016 17 A.I.K.T.C.CLASS B.E.");

        BUZZER = 0;
}
```

```
void sendsms7()
{

        BUZZER = 1;
        printf("AT\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGF=1\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGS="9702856015");
        delayms(2000); // 2 sec delay
        delayms(1000); // 1 sec delay

        printf ("L3:5KM-FAULT HAS BEEN DETECTED KINDLY PAY
ATTENTION    2016 17 A.I.K.T.C.CLASS B.E.");

        BUZZER = 0;
}
void sendsms8()
{

        BUZZER = 1;
        printf("AT\n");
        delayms(2000); // 2 sec delay
```

```
        delayms(1000); // 1 sec delay

        printf("AT+CMGF=1\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGS="9702856015");
        delayms(2000); // 2 sec delay
        delayms(1000); // 1 sec delay

        printf ("L3:5KM-10KM FAULT HAS BEEN DETECTED KINDLY PAY
ATTENTION    2016 17 A.I.K.T.C.CLASS B.E.");

        BUZZER = 0;
}
void sendsms9()
{

        BUZZER = 1;
        printf("AT\n");
        delayms(2000); // 2 sec delay

        delayms(1000); // 1 sec delay

        printf("AT+CMGF=1\n");
        delayms(2000); // 2 sec delay
```

```
        delayms(1000); // 1 sec delay

        printf("AT+CMGS="9702856015");
        delayms(2000); // 2 sec delay
        delayms(1000); // 1 sec delay

        printf ("L3:10KM-FAULT HAS BEEN DETECTED KINDLY PAY
ATTENTION    2016 17 A.I.K.T.C.CLASS B.E.");

        BUZZER = 0;
}
void main()
{
unsigned char adc1, adc2,adc3;
float t1, h2, l3 ;
// -=-=- Intialize variables -=-=-=

        BUZZER=0;
// -=-=- Intialise -=-=-=
    initADC();
    lcdInit();
    serialInit();

// -=-=- Welcome LCD Message -=-=-=
    lcdClear();
    lcdGotoXY(0,0);     // 1st Line of LCD
    //            "xxxxxxxxxxxxxxxx"
    lcdPrint("UNDERGROUD CABLE");
    lcdGotoXY(0,1);     // 2nd Line of LCD
```

```
    //              "xxxxxxxxxxxxxxxx"
    lcdPrint("FAULT DETECTOR");
    delayms(5000); // 5 sec
    lcdClear();
    lcdClear();
    lcdGotoXY(0,0);     // 1st Line of LCD
    //              "xxxxxxxxxxxxxxxx"
    lcdPrint("A.I.K.T.C");
    lcdGotoXY(0,1);     // 2nd Line of LCD
    //              "xxxxxxxxxxxxxxxx"
    lcdPrint("YEAR 2016-17.");
    delayms(5000); // 5 sec

// -=-=- Program Loop -=-=-=
    while(1)
    {
        delayms(250);
        adc1 = getADC(1);
        t1 = (float)(adc1);

        delayms(250);
        adc2 = getADC(2);
        h2 = (float)(adc2);
          delayms(250);

        adc3 = getADC(3);
        l3 = (float)(adc3);
        delayms(250);
```

```
        if(t1<69&& h2<69 && l3<69)
            {
            BUZZER = 0;
    lcdClear();
    lcdGotoXY(0,0);      // 1st Line of LCD
    //              "xxxxxxxxxxxxxxxx"
    lcdPrint("L1:N.F.      L2:N.F.");
    lcdGotoXY(0,1);      // 2nd Line of LCD
    //              "xxxxxxxxxxxxxxxx"
    lcdPrint("L3:N.F(NO FAULT)");
    delayms(1000); // 5 sec
        }
//
             if(t1>=70 && t1<=80)

            {
    BUZZER=1;
    lcdClear();
    lcdGotoXY(0,0);      // 1st Line of LCD
    //              "xxxxxxxxxxxxxxxx"
    lcdPrint("L1:5KM-FAULT");
    sendsms1();

        }
             if(t1>=81 && t1<=90)
            {
    BUZZER=1;
```

```
lcdClear();
lcdGotoXY(0,0);      // 1st Line of LCD
//              "xxxxxxxxxxxxxxxx"
lcdPrint("L1:5-10KM FAULT");
sendsms2();

    }

 if(t1>=91 && t1<=95)
{
BUZZER=1;
lcdClear();
lcdGotoXY(0,0);      // 1st Line of LCD
//              "xxxxxxxxxxxxxxxx"
lcdPrint("L1:10KM FAULT");
    sendsms3();


    }



 if(h2>=70 && h2<=80)


        {
BUZZER=1;
lcdClear();
lcdGotoXY(0,0);      // 1st Line of LCD
//              "xxxxxxxxxxxxxxxx"
```

```
lcdPrint("L2:5KM-FAULT");
sendsms4();

    }
        if(h2>=81 && h2<=90)
       {
BUZZER=1;
lcdClear();
lcdGotoXY(0,0);     // 1st Line of LCD
//          "xxxxxxxxxxxxxxxx"
lcdPrint("L2:5-10KM FAULT");
sendsms5();

    }

 if(h2>=91 && h2<=95)
{
BUZZER=1;
lcdClear();
lcdGotoXY(0,0);     // 1st Line of LCD
//          "xxxxxxxxxxxxxxxx"
lcdPrint("L2:10KM FAULT");
    sendsms6();


    }
        if(l3>=70 && l3<=80)

       {
```

```
BUZZER=1;
lcdClear();
lcdGotoXY(0,0);      // 1st Line of LCD
//              "xxxxxxxxxxxxxxxx"
lcdPrint("L3:5KM-FAULT");
sendsms7();

      }
              if(l3>=81 && l3<=90)
            {
BUZZER=1;
lcdClear();
lcdGotoXY(0,0);      // 1st Line of LCD
//              "xxxxxxxxxxxxxxxx"
lcdPrint("L3:5-10KM FAULT");
sendsms8();

      }

  if(l3>=91 && l3<=95)
{
BUZZER=1;
lcdClear();
lcdGotoXY(0,0);      // 1st Line of LCD
//              "xxxxxxxxxxxxxxxx"
lcdPrint("L3:10KM FAULT");
      sendsms9();
```

```
        }
    }
}
```

4.5 TESTING OF COMPONENTS

Easy way of testing electronic components (resistor, capacitor, diode, transistor) using Multi-meter.

How to test resistors?

Read the indicated color code value then select the OHM-scale within but not way below the indicated value. A resistor is good if its resistance is close to the indicated. Tolerance should be considered with the ohmmeter reading. While, no resistance reading at all on the ohmmeter scale settings, the resistor is open. A zero resistance reading on all ohmmeter scale settings, resistor is shorted.

How to test capacitors?

In most cases, a capacitor fails due to the deterioration of the dielectric material between its plate. Defective capacitors can have an internal shorted terminals, excessive leakage and degradation of capacitance meter. Momentarily, short the terminal of the electrolytic capacitor to discharge it.

To test a capacitor, set the multi-tester to Rx 10 or Rx1K scale. Connect the tester negative probe to the capacitor positive terminal, the positive probe to the negative terminal. A good indication for electrolytic capacitor shows the meter needle deflecting towards zero and moves back again to infinite resistance position. For ceramic, Mylar and other capacitor with a capacitance with less than 1.0 uF, the meter will not deflect at all. A defective indication for an electrolytic capacitor shows that the meter will rest on zero and remain stationary at a point which is an indication that the capacitor is shorted.

How to test diodes?

Set the multi-tester knob to any of the resistance position (x1, x10, x1K or 10K ohm). Connect the positive probe to the anode and the negative probe to the cathode. Then connect the positive probe to the cathode and the negative probe to the anode of the diode. A good indication in the first procedure will show the meter deflected very little or may not deflect at all. And in the second procedure, the meter will deflect towards zero. The actual resistance reading is the forward resistance of the diode. A defective indication shows that the meter won't deflect at all even when the probes are reversed. Or the meter deflects at the same time or almost the same resistance reading for both steps.

How to test transistors:

Bipolar transistors are usually checked out of a circuit by means of an ohmmeter. When it is desired to check for the resistance across the transistor emitter and collector, NPN or PNP, ohmmeter probes may be connected either way. A good transistor will show above a reading above 1000 ohm.

How to determine if it is NPN or PNP transistor?

To determine the correct terminal of the transistors, set the range selector to x 1 or 10 ohm. Connect the positive probe to the emitter and the negative probe to the base of the transistor. Note the reading interchange the connection of the probes to the leads of the transistor.

CHAPTER 5
TYPES OF FAULTS ,GSM MODEM AND FUTURE SCOPE

5.1 Types of transmission line faults

Power system's faults may be categorized as shunt faults or series faults. The most common type of shunt faults is Single Line-to-ground faults (SLG). This type of fault occurs when one conductor falls to the ground or gets into contacts with the neutral wire. It could also be the result of falling trees in a rainy storm. This type could be represented below.

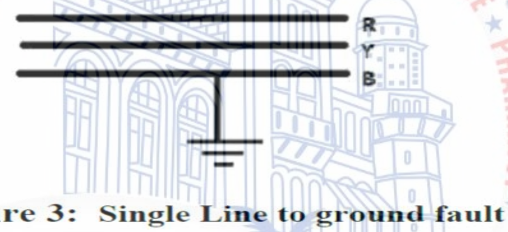


FIG 5.1

The second most occurring type of shunt faults is the Line-to-Line fault (LL). This is said to occur when two transmission lines are short-circuited. As in the case of a large bird standing on one transmission line and touching the other, or if a tree branch happens to fall on top of two power transmission lines. This type could be represented below.

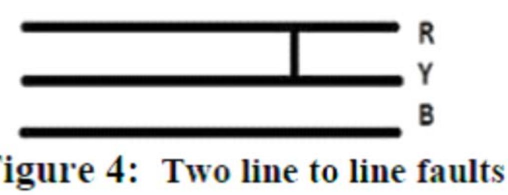


FIG 5.2

The third type of shunt fault is the Double Line-to-Ground fault (DLG) in. This can be a result of a tree falling on two of the power lines, or other causes.

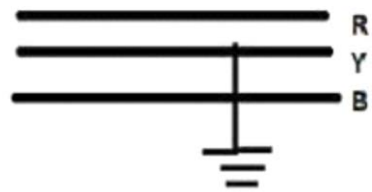


Figure 5: double line-to-ground fault

FIG 5.3

5.2 GSM MODEM

Architecture of the GSM Network

Global System for Mobile Communications (GSM) is a digital wireless network standard designed by standardization committees from major European telecommunications operators and manufacturers. The GSM standard provides a common set of compatible services and capabilities to all mobile users across Europe and several million customers worldwide.

The GSM Modem

A modem (modulator-demodulator) is a device that modulates an analog carrier signal to encode digital information, and also demodulates such a carrier signal to decode the transmitted information. The goal is to produce a signal that can be transmitted easily and decoded to reproduce the original digital data.

The GSM Modem comes with a serial interface through which the modem can be controlled using attention (AT) command interface. An antenna and a power adapter are provided.

The basic segregation or working of the modem is as follows:

- Voice calls
- Short Message Service (SMS)
- GSM Data calls

• General Packet Radio Services (GPRS)

FIG 5.4 GSM MODEM

5.3 FUTURE SCOPE

The proposed system is intended to automatically detect faults when they occur, analyse the fault to determine the type and then send information based on the fault type and fault location to the control room via GSM. The device location is determined by the SIM card in the modem, each SIM card having a unique identification an hence is used as the device's address.

The system senses, analyses and transmits. It does this with the microcontroller which analyses, interprets and sends digital signals to the I/O devices for the system to operate. By programming, the microcontroller is made to perform these functions.

The hardware model of Underground Cable Fault Locator is implemented and favorable results were brought forward. This hardware model can locate the exact fault location in an underground cable. Further this project can be enhanced by using capacitor in an AC circuit to measure the impedance which can even locate the open circuited cable, unlike the short circuited fault only using resistors in DC circuit as followed in the above proposed project

6 REFERENCES

[1] http://www.gridcogh.com/en/press-media/gridco-reliability-assessment-report.php

[2] Anon, (2009a) Electric Transmission Lines

[3] Anon, (2010) Generation Transmission Distribution

[4] Power Systems Energy Consulting, Ghana Power Reliability Report , 2010. <http://www.gridcogh.com/en/national-grid.php pp. 16>

[5] Power Systems Energy Consulting, Ghana Power Reliability Report , 2010. http://www.gridcogh.com/en/national-grid.php

****