**AUTOMATIC FAULT DETECTION & LOCATION OF THE TRANSMISSION LINES USING IOT**

**Dr.RAJAN.V.R**

**Assistant Professor**

**ABSTRACT**

A smart GSM based fault detection and location system was used to adequately and accurately indicate and locate the fault had occurred. 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, Arduino, current detector, voltage sensing circuit (for over load cases), and a GSM modem.

Finally, the fault information is transmitted to the control room. In conclusion, the time required to locate a fault is drastically reduced, the system automatically and accurately provides accurate fault location information. By using this project, we can detect the faults of three phase transmission lines one can monitor the Temperature, Voltage, Current by means of GSM modem by sending message.

Detecting and Locating faults in power is very necessary for health operation of power system. In the proposed concept power line is divided by wireless sensor network nodes that could sense the faulty condition in power line, display to operator as well as send SMS through GSM modern to service engineer.

1. **INTRODUCTION**

Transmission network is considered to be one of the vital parts of power system. The loss in transmission and distribution network is considered to be very high, compared to other parts of power system. The electric power infrastructure is highly vulnerable against many forms of natural and malicious physical events. Many electric power transmission companies have primarily relied on circuit indicators to detect faulty sections of their transmission lines.

Wireless sensor-based monitoring of transmission lines provides a solution for several of these concerns like real time structural awareness, faster fault localization, accurate fault diagnosis by identification and differentiation of electrical faults from the mechanical faults, cost reduction due to condition based maintenance rather than periodic maintenance, etc. These applications specify stringent requirements such as fast delivery of enormous amount of highly reliable data.

The success of these applications depends on the design of cost effective and reliable network architecture with a fast response time. The network must be able to transport sensitive data such as current state of the transmission line and control information to and from the transmission grid. This project provides a cost optimized framework to design a real time data transmission network. To monitor the status of the power system in real time, sensors are put in various components in the power network.

1. **EMBEDDED SYSTEM IMPLEMENTATION**

**2.1 Introduction:**

An embedded system is one kind of a computer system mainly designed to perform several tasks like to access, process, and store and also control the data in various electronics-based systems. Embedded systems are a combination of hardware and software where software is usually known as firmware that is embedded into the hardware. One of its most important characteristics of these systems is, it gives the o/p within the time limits.

Embedded systems support to make the work more perfect and convenient. So, we frequently use embedded systems in simple and complex devices too. The applications of embedded systems mainly involve in our real life for several devices like microwave, calculators, TV remote control, home security and neighborhood traffic control systems, etc.devices like microwave, calculators, TV remote control, home security and neighborhood traffic control systems, etc.

User interface

Embedded system

Hardware

Software

Inputs

Output

Link to other systems

**Fig. 2.1:- Overview of embedded system**

* 1. **Embedded system:**

Embedded system includes mainly two sections, they are

1. Hardware

2. Softwar

Power supply and oscillator circuits

Timers

Processor

Interrupt controller

Input devices interfacing

and driver circuits

Memory

Application specific circuits

Serial communication ports

Parallel ports

Output devices interfacing

**Fig. 2.2:- Block diagram of embedded system**

* 1. **Embedded System Hardware:**

As with any electronic system, an embedded system requires a hardware platform on which it performs the operation. Embedded system hardware is built with a microprocessor or microcontroller. The embedded system hardware has elements like input output (I/O) interfaces, user interface, memory and the display. Usually, an embedded system consists of:

* Power Supply
* Processor
* Memory
* Timers
* Serial communication ports
* Output/Output circuits
* System application specific circuits

Embedded systems use different processors for its desired operation. some of the processors used are

1. Microprocessor

2. Microcontroller

3. Digital signal processor

* 1. **Microprocessor vs. Microcontroller :-**

**Microprocessor**

* **CPU** on a chip.
* We can attach required amount of ROM, RAM and I/O ports.
* Expensive due to external peripherals.
* Large in size
* general-purpose

**Microcontroller**

* **Computer** on a chip
* fixed amount of on-chip ROM, RAM, I/O ports
* Low cost.
* Compact in size.
* Specific –purpose

1. **LITERATURE SURVEY**

S.Lefebvre [1] et.al displayed the benefits of versatile tuning of current controllers in a HVDC converter framework relying on the framework prerequisites. It is demonstrated that the converter SCR or the net recompense obstruction offered by the converter is the significant parameter which impacts tuning .An essential HVDC framework is linearised around a working point and the controller takes care that Eigen esteems and zeros of the framework are kept up at predefined areas for each arrangement of variety of framework parameters. Estimation of this parameter variety is finished by exposing the framework to a little commotion signal .Estimation and controls are done at various transfer speeds to improve heartiness of the controller.

John Reeve [2]et.al in his paper attempted to fuse gain booking versatile control in to the quick controller circles in the control of dc transmission frameworks to improve the exhibition of the framework under expansive intrusions ,low Effective Short Circuit Ratio’s and shortcomings that bring down the working SCR of the framework further. DC current blunder signal, dc voltage mistake signal, air conditioning voltage zero intersections, and terminating point at the rectifier are the different booking factors picked relying on the reaction of the framework under vast air conditioning aggravations. These factors are assumed for each case to improve the strength of the framework under vast intrusions dependent on the outcome given by the possibility pointers.

John Reeve [3] et.al amalgamated the hypothetical part of auto tuning with addition planning .Two points are essentially adderesed.1) regardless of whether auto tuning gives adequate focal points over fixed additions i.e ordinary controller gain booking or the blend of two controls can be connected to expand the heartiness of the two controllers for substantial unsettling influences. It was appeared for specific applications including nonstop or unexpected low short out proportion, auto tuning alone may not be solid in light of specific unsettling influences .It must be joined with increase planning.

P.K.Dash et al [4] et.al presented a viable control technique for a HVDC framework dependent on the guideline of input linearization. A neural estimation calculation has been utilized to follow the linearised control parameters which are elements of rectifier side dc voltage, inverter side dc voltage, dc connect reactance and proportionate opposition . The dc interface is liable to different transient conditions to demonstrate the exhibition of the controller. A superior blunder following law can in any case ad lib the working of the controller for dynamic solidness.

A.Routary et al [5] et.al supplanted the rectifier side current controller with a fluffy self tuning controller .The controller gain which deals with Kp and KI, relative in addition to essential controller constants is balanced through fluffy interface. The linearised estimations of current blunder and its subordinate are the two data sources used to produce the arranged estimation of terminating plot for the rectifier end .On a comparable lines requested estimation of annihilation edge at the inverter end is created and the exhibition of the framework is seen under transient conditions and demonstrated the predominance of self tuning the PI controller parameters utilizing Fuzzy rational.

Chi-Hshing Lin [6] et.al has discussed the peculiarity between two much of the time happening deficiencies in a HVDC interface. He made an examination between the failure to fire deficiency in the rectifier valve and inverter valve. A dynamic reenactment results discovered the distinction between the two resultant wonders. A failure to fire flaw in the rectifier valve creates a huge torsional torque in a turbine generator connecting the inverter station at whatever point the characteristic torsional modes are hindered by the power variety, which thusly influences the rectifier side framework recurrence. Despite what might be expected, a fizzle deficiency in an inverter valve attempts to cause compensation disappointment in converters, bringing about HVDC disappointment. This, promptly influences the rectifier and inverter sides of the generator.

1. **EXISTING SYSTEM**

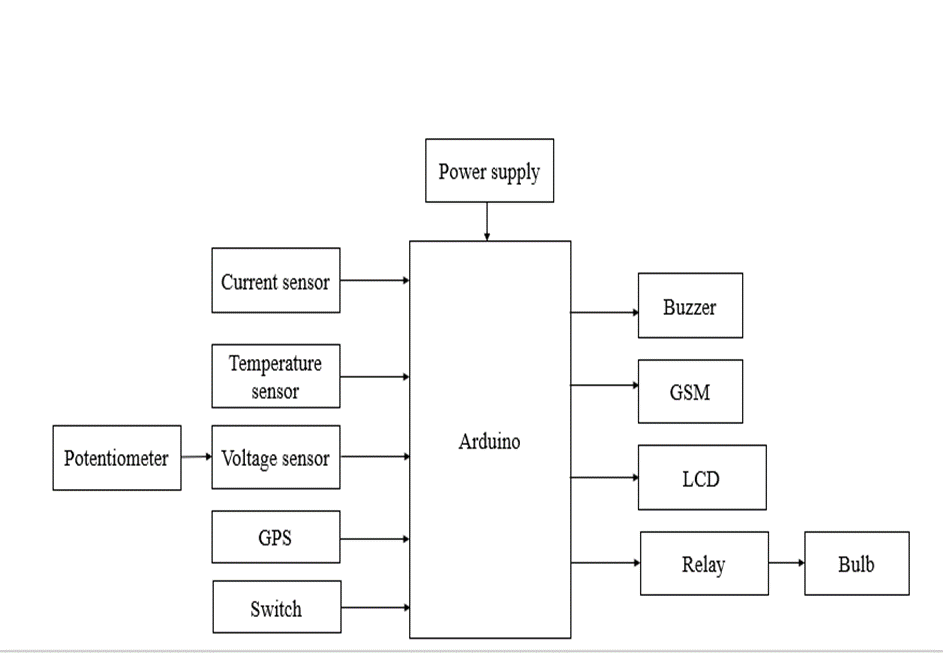
In existing system using nodes they are detecting the faults for that we do not know what fault it is, whether it is short circuit or wire cut or over load. To overcome that problem, we proposed a new system which is less cost and less complexity with good accurate result**.**

**Drawbacks:**

* No accuracy.
* We couldn’t know what type of fault occurred.
* High cost.

1. **PROPOSED SYSTEM**

Here we proposed a system uses an Arduino, current detector, voltage sensing circuit (for over load cases), and a GSM modem. The system automatically detects faults, analyses and classifies these faults and then. 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. And here LCD is used to see which type of fault it is.



**Fig. 5.3:- Block diagram**

1. **HARDWARE REQUIREMENTS**

**6.1. ARDUINO:**

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. For advanced Arduino users, prowl the web; there are lots of resources.

This is what the Arduino board looks like.



**Fig. 6.4:- Arduino**

The Arduino programming language is a simplified version of C/C++. If you know C, programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions.

*6.1.1 Atmega328p features:*

* High Performance, Low Power AVR® 8-Bit Microcontroller
* Advanced RISC Architecture

– 131 Powerful Instructions

– Most Single Clock Cycle Execution

– 32 x 8 General Purpose Working Registers

– Fully Static Operation

– Up to 20 MIPS Throughput at 20 MHz

– On-chip 2-cycle Multiplier

* High Endurance Non-volatile Memory Segments

– 4/8/16/32K Bytes of In-System Self-Programmable Flash progam memory (ATmega48PA/88PA/168PA/328P)

– 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)

– 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)

– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM

– Data retention: 20 years at 85°C/100 years at 25°C(1)

– Optional Boot Code Section with Independent Lock Bits In-System Programming by On-chip Boot Program True Read-While-Write Operation

– Programming Lock for Software Security

* Peripheral Features

– Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode

– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

– Real Time Counter with Separate Oscillator

– Six PWM Channels – 8-channel 10-bit ADC in TQFP and QFN/MLF package Temperature Measurement – 6-channel 10-bit ADC in PDIP Package Temperature Measurement

– Programmable Serial USART

– Master/Slave SPI Serial Interface

– Byte-oriented 2-wire Serial Interface (Philips I2 C compatible)

– Programmable Watchdog Timer with Separate On-chip Oscillator

– On-chip Analog Comparator

– Interrupt and Wake-up on Pin Change

* Special Microcontroller Features

– Power-on Reset and Programmable Brown-out Detection

– Internal Calibrated Oscillator

– External and Internal Interrupt Sources

– Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

* I/O and Packages

– 23 Programmable I/O Lines

– 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

* Operating Voltage:

– 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P

* Temperature Range:

– -40°C to 85°C

* Speed Grade:

– 0 - 20 MHz @ 1.8 - 5.5V

* Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:

– Active Mode: 0.2 mA

– Power-down Mode: 0.1 µA

– Power-save Mode: 0.75 µA (Including 32 kHz RTC)

*6.1.2 power:-*

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

**• VIN**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

**• 5V.**This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

**• 3V3**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

**• GND.** Ground pins.

*6.1.3 Input and Output:-*

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

**• Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

**• External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

**• PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.

**• SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication using the SPI library.

**• LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

**• TWI: A4 or SDA pin and A5 or SCL pin.** Support TWI communication using the Wire library.

There are a couple of other pins on the board:

**• AREF.** Reference voltage for the analog inputs. Used with analogReference().

**• Reset.** Bring this line LOW to reset the microcontroller**.** Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

* 1. **LCD:-**

LCD (Liquid Crystal Display) is the innovation utilized in scratch pad shows and other littler PCs. Like innovation for light-producing diode (LED) and gas-plasma, LCDs permit presentations to be a lot more slender than innovation for cathode beam tube (CRT). LCDs expend considerably less power than LED shows and gas shows since they work as opposed to emanating it on the guideline of blocking light.

A LCD is either made with a uninvolved lattice or a showcase network for dynamic framework show. Likewise alluded to as a meager film transistor (TFT) show is the dynamic framework LCD. The uninvolved LCD lattice has a matrix of conductors at every crossing point of the network with pixels. Two conductors on the lattice send a current to control the light for any pixel. A functioning framework has a transistor situated at every pixel crossing point, requiring less current to control the luminance of a pixel.

*6.2.1 Data/Signals/Execution of LCD:-*

Now that was all about the signals and the hardware. Let us come to data, signals and execution.

Two types of signals are accepted by LCD, one is data and one is control. The LCD module recognizes these signals from the RS pin status. By pulling the R / W pin high, data can now also be read from the LCD display. Once the E pin has been pulsed, the LCD display reads and executes data at the falling edge of the pulse, the same for the transmission case.

It takes 39-43μS for the LCD display to place a character or execute a command. It takes 1.53ms to 1.64ms except for clearing display and searching for cursor to the home position.

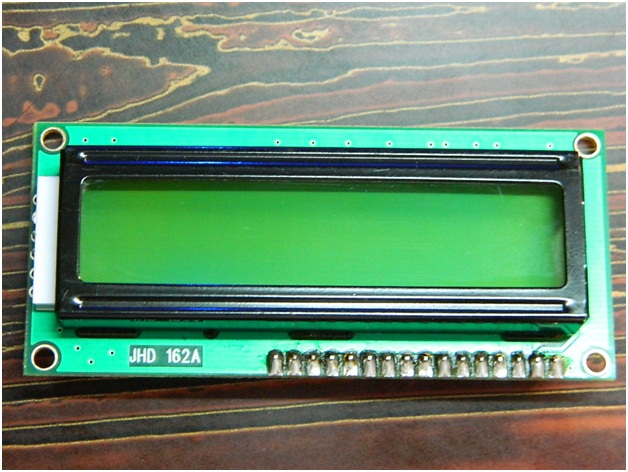
Any attempt to send data before this interval may result in failure in some devices to read data or execute the current data. Some devices compensate for the speed by storing some temporary registers with incoming data.

There are two RAMs for LCD displays, namely DDRAM and CGRAM. DDRAM registers the position in which the character would be displayed in the ASCII chart. Each DDRAM byte represents every single position on the display of the LCD.

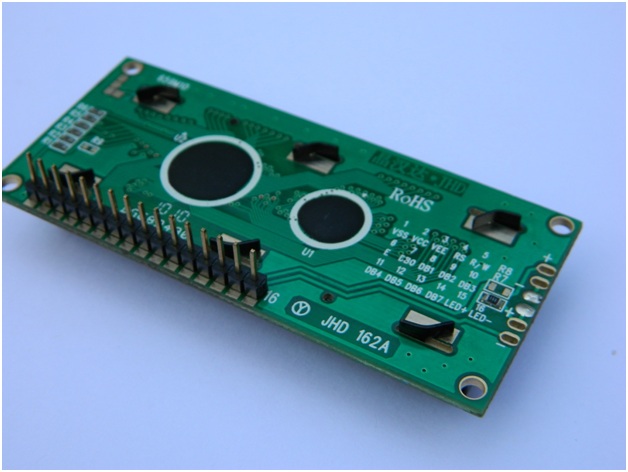
The DDRAM information is read by the LCD controller and displayed on the LCD screen. CGRAM enables users to define their personalized characters. Address space is reserved for users for the first 16 ASCII characters.

Users can easily display their custom characters on the LCD screen after CGRAM has been set up to display characters.

*6.2.2. Images of LCD Display:-*

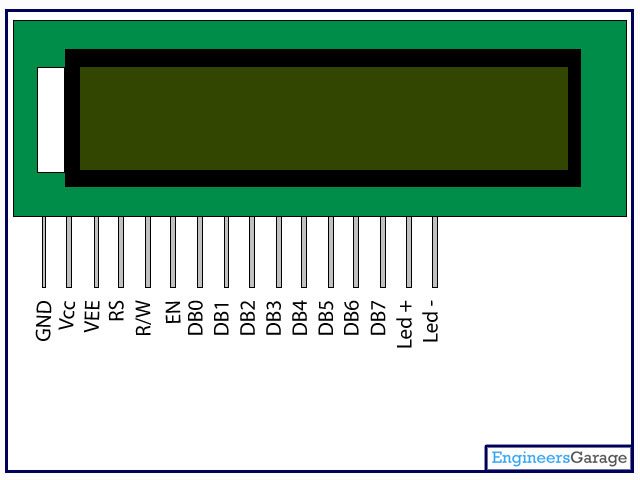
[](http://www.circuitstoday.com/wp-content/uploads/2012/02/LCD-Display-Front-Side.jpg)

**Fig.6.5: LCD – Front View**

[](http://www.circuitstoday.com/wp-content/uploads/2012/02/lcd-display-back-side.jpg)

**Fig.6.6: LCD – Back View**

* + 1. *Pin Diagram:-*



**Fig. 6.7:LCD Pin diagram**

* + 1. *Pin Description:*

**Table. 6.1: LCD Pin Description**

|  |  |  |
| --- | --- | --- |
| Pin No | Function | Name |
| 1 | Ground (0V) | Ground |
| 2 | Supply voltage; 5V (4.7V – 5.3V) | Vcc |
| 3 | Contrast adjustment; through a variable resistor | VEE |
| 4 | Selects command register when low; and data register when high | Register Select |
| 5 | Low to write to the register; High to read from the register | Read/write |
| 6 | Sends data to data pins when a high to low pulse is given | Enable |
| 7 | 8-bit data pins | DB0 |
| 8 | DB1 |
| 9 | DB2 |
| 10 | DB3 |
| 11 | DB4 |
| 12 | DB5 |
| 13 | DB6 |
| 14 | DB7 |
| 15 | Backlight VCC (5V) | Led+ |
| 16 | Backlight Ground (0V) | Led- |

*6.2.5 LCD Commands:*

There are some preset commands in the LCD that we need to send to the LCD via some microcontroller. The following are some important command instructions:

**Table. 6.2: LCD Commands**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Hex Code** | **Command to LCD instruction Register** |
| 1 | 01 | Clear display screen |
| 2 | 02 | Return home |
| 3 | 04 | Decrement cursor (shift cursor to left) |
| 4 | 06 | Increment cursor (shift cursor to right) |
| 5 | 05 | Shift display right |
| 6 | 07 | Shift display left |
| 7 | 08 | Display off, cursor off |
| 8 | 0A | Display off, cursor on |
| 9 | 0C | Display on, cursor off |
| 10 | 0E | Display on, cursor blinking |
| 11 | 0F | Display on, cursor blinking |
| 12 | 10 | Shift cursor position to left |
| 13 | 14 | Shift cursor position to right |
| 14 | 18 | Shift the entire display to the left |
| 15 | 1C | Shift the entire display to the right |
| 16 | 80 | Force cursor to beginning ( 1st line) |
| 17 | C0 | Force cursor to beginning ( 2nd line) |
| 18 | 38 | 1. lines and 5×7 matrix |

* 1. **GSM:**

GSM is a mobile communication modem; it is stands for global system for mobile communication (GSM). The idea of GSM was developed at Bell Laboratories in 1970.  It is widely used mobile communication system in the world. GSM is an open and digital cellular technology used for transmitting mobile voice and data services operates at the 850MHz, 900MHz, 1800MHz and 1900MHz frequency bands.

GSM system was developed as a digital system using time division multiple access (TDMA) technique for communication purpose. A GSM digitizes and reduces the data, then sends it down through a channel with two different streams of client data, each in its own particular time slot. The digital system has an ability to carry 64 kbps to 120 Mbps of data rates.

There are various cell sizes in a GSM system such as macro, micro, pico and umbrella cells. Each cell varies as per the implementation domain. There are five different cell sizes in a GSM network macro, micro, pico and umbrella cells. The coverage area of each cell varies according to the implementation environment.

### *6.3.1 GSM Architecture:-*

A GSM network consists of the following components:

* **A Mobile Station:**  It is the mobile phone which consists of the transceiver, the display and the processor and is controlled by a SIM card operating over the network.
* **Base Station Subsystem:** It acts as an interface between the mobile station and the network subsystem. It consists of the Base Transceiver Station which contains the radio transceivers and handles the protocols for communication with mobiles. It also consists of the Base Station Controller which controls the Base Transceiver station and acts as a interface between the mobile station and mobile switching centre.
* **Network Subsystem:** It provides the basic network connection to the mobile stations. The basic part of the Network Subsystem is the Mobile Service Switching Centre which provides access to different networks like ISDN, PSTN etc. It also consists of the Home Location Register and the Visitor Location Register which provides the call routing and roaming capabilities of GSM. It also contains the Equipment Identity Register which maintains an account of all the mobile equipments wherein each mobile is identified by its own IMEI number. IMEI stands for International Mobile Equipment Identity.

### *6.3.2 Features of GSM Module:-*

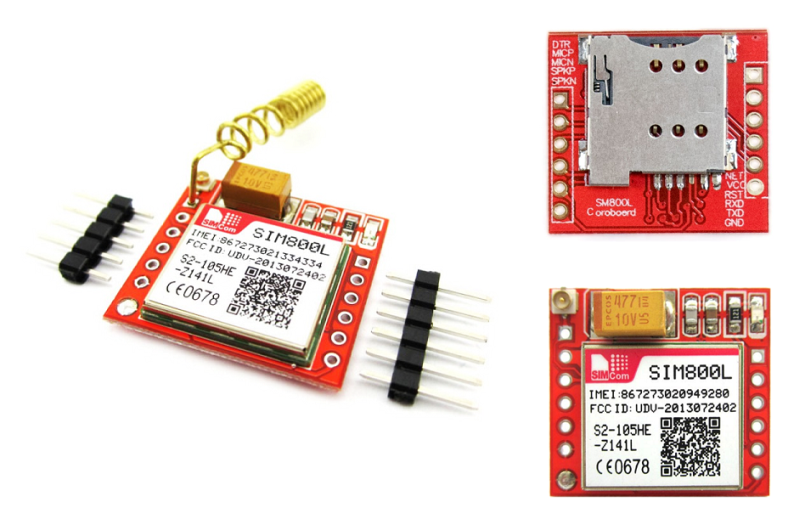
* Improved spectrum efficiency
* International roaming
* Compatibility with integrated services digital network (ISDN)
* Support for new services.
* SIM phonebook management
* Fixed dialing number (FDN)
* Real time clock with alarm management
* High-quality speech
* Uses encryption to make phone calls more secure
* Short message service (SMS)

The security strategies standardized for the GSM system make it the most secure telecommunications standard currently accessible. Although the confidentiality of a call and secrecy of the GSM subscriber is just ensured on the radio channel, this is a major step in achieving end-to- end security.

### *6.3.3 GSM Modem:-*

A GSM modem is a device which can be either a mobile phone or a modem device which can be used to make a computer or any other processor communicate over a network. A GSM modem requires a SIM card to be operated and operates over a network range subscribed by the network operator.  It can be connected to a computer through serial, USB or Bluetooth connection.

A GSM modem can also be a standard GSM mobile phone with the appropriate cable and software driver to connect to a serial port or USB port on your computer. GSM modem is usually preferable to a GSM mobile phone. The GSM modem has wide range of applications in transaction terminals, supply chain management, security applications, weather stations and GPRS mode remote data logging.

****

**Fig. 6.8: GSM**

It requires a **SIM (Subscriber Identity Module)** card just like mobile phones to activate communication with the network. Also they have **IMEI** (International Mobile Equipment Identity) number similar to mobile phones for their identification. A GSM/GPRS MODEM can perform the following operations:

1.      Receive, send or delete SMS messages in a SIM.

2.      Read, add, search phonebook entries of the SIM.

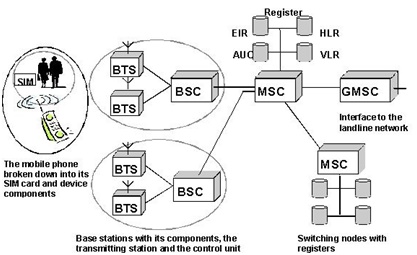
3.      Make, Receive, or reject a voice call.

The MODEM needs **AT commands**, for interacting with processor or controller, which are communicated through serial communication. These commands are sent by the controller/processor. The MODEM sends back a result after it receives a command. Different AT commands supported by the MODEM can be sent by the processor/controller/computer to interact with the **GSM and GPRS cellular network**.

### ***6.3.4 GSM Architecture:-***

The GSM architecture is divided into Radio Subsystem, Network and Switching Subsystem and the Operation Subsystem. The radio sub system consists of the Mobile Station and Base Station Subsystem.

The mobile station is generally the mobile phone which consists of a transceiver, display and a processor. Each handheld or portable mobile station consists of a unique identity stored in a module known as SIM (Subscriber Identity Chip). It is a small microchip which is inserted in the mobile phone and contains the database regarding the mobile station.



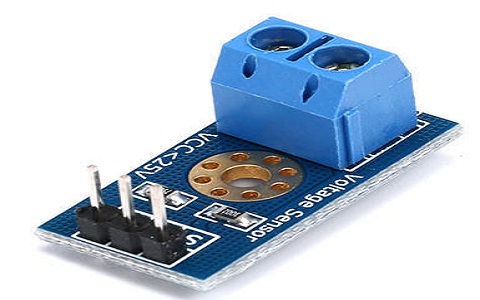
**Fig. 6.9: GSM Architecture**

## **Voltage Sensor:-**

This sensor is used to monitor, calculate and determine the voltage supply. This sensor can determine the AC or DC voltage level. The input of this sensor can be the voltage whereas the output is the switches, analog voltage signal, a current signal, an audible signal, etc. Some sensors provide sine waveforms or pulse waveforms like output & others can generate outputs like [AM (Amplitude Modulation)](https://www.elprocus.com/what-is-amplitude-modulation-derivations-typesand-applications/), [PWM (Pulse Width Modulation)](https://www.elprocus.com/pulse-width-modulation-pwm/) or [FM (Frequency Modulation)](https://www.elprocus.com/frequency-modulation-and-its-applications/). The measurement of these sensors can depend on the voltage divider.

**Table.6.3: Voltage Sensor Measurements**

|  |  |
| --- | --- |
| **Input Voltage (V)** | 0 to 25 |
| **Voltage Detection Range (V)** | 0.02445 to 25 |
| **Analog Voltage Resolution (V)** | 0.00489 |
| **Length (mm)** | 28 |
| **Width (mm)** | 14 |
| **Height (mm)** | 13 |
| **Weight (gm)** | 4 |



**Fig. 6.10 Voltage Sensor**

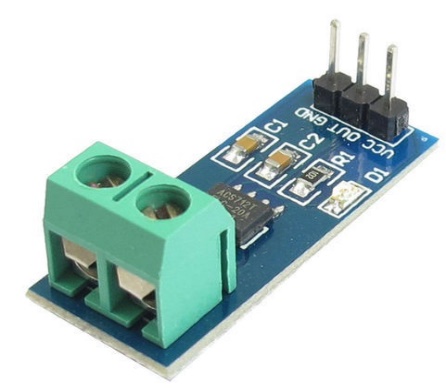
This sensor includes input and output. The input side mainly includes two pins namely positive and negative pins. The two pins of the device can be connected to the positive & negative pins of the sensor. The device positive & negative pins can be connected to the positive & negative pins of the sensor. The output of this sensor mainly includes supply voltage (Vcc), ground (GND), analog o/p data.

## **ACS712 Current Sensor:-**

Current flowing through a conductor causes a voltage drop. The relation between current and voltage is given by Ohm’s law. In electronic devices, an increase in the amount of current above its requirement leads to overload and can damage the device.

Measurement of current is necessary for the proper working of devices. Measurement of voltage is Passive task and it can be done without affecting the system. Whereas measurement of current is an Intrusive task which cannot be detected directly as voltage.

6.5.1 Specifications:

**Table. 6.4: Current Sensor Specifications**

|  |  |
| --- | --- |
| **Current sensor chip** | ACS712ELC-20A |
| **Operating Voltage (V)** | 4.5V ~ 5.5V DC |
| **Measure Current Range** | -20 ~ +20A |
| **Sensitivity** | 100mV/A |
| **Length (mm)** | 32 |
| **Width (mm)** | 13 |
| **Height (mm)** | 13.5 |
| **Weight (gm)** | 5 |

**Fig.6.11 : Current Sensor**

For measuring current in a circuit, a sensor is required. ACS712 Current Sensor is the sensor that can be used to measure and calculate the amount of current applied to the conductor without affecting the performance of the system.

ACS712 Current Sensor is a fully integrated, Hall-effect based linear sensor IC. This IC has a 2.1kV RMS voltage isolation along with a low resistance current conductor.

### *6.5.2 Working Principle:-*

Current Sensor detects the current in a wire or conductor and generates a signal proportional to the detected current either in the form of analog voltage or digital output.[Current Sensing](https://www.elprocus.com/current-sensor/) is done in two ways – Direct sensing and Indirect Sensing. In Direct sensing, to detect current, Ohm’s law is used to measure the voltage drop occurred in a wire when current flows through it.

A current-carrying conductor also gives rise to a magnetic field in its surrounding. In Indirect Sensing, the current is measured by calculating this magnetic field by applying either [Faraday’s law](https://www.elprocus.com/electromagnetic-induction-and-laws/) or Ampere law. Here either a [Transformer](https://www.elprocus.com/various-types-of-transformers-applications/) or [Hall effect sensor](https://www.elprocus.com/hall-effect-sensor-working-principle-and-applications/) or fiberoptic current sensor are used to sense the magnetic field.

ACS712 Current Sensor uses Indirect Sensing method to calculate the current. To sense current a liner, low-offset Hall sensor circuit is used in this IC. This sensor is located at the surface of the IC on a copper conduction path. When current flows through this copper conduction path it generates a magnetic field which is sensed by the Hall effect sensor. A voltage proportional to the sensed magnetic field is generated by the Hall sensor, which is used to measure current.

The proximity of the magnetic signal to the Hall sensor decides the accuracy of the device. Nearer the magnetic signal higher the accuracy. ACS712 Current Sensor is available as a small, surface mount SOIC8 package. In this IC current flows from Pin-1 and Pin-2 to Pin-3 and Pin-4. This forms the conduction path where the current is sensed. Implementation of this IC is very easy.

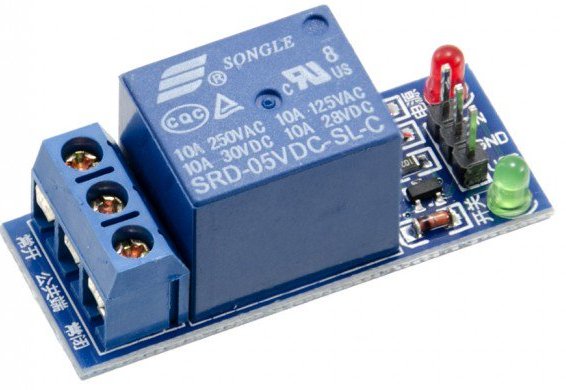
ACS712 can be used in applications requiring electrical isolation as the terminals of the conduction path are electrically isolated from the IC leads. Thus, this IC doesn’t require any other isolation techniques. This IC requires a supply voltage of 5V. Its output voltage is proportional to AC or DC current. ACS712 has a nearly zero [magnetic hysteresis](https://en.wikipedia.org/wiki/Magnetic_hysteresis).

Where Pin-1 to Pin-4 forms the conduction path, Pin-5 is the signal ground pin. Pin-6 is the FILTER pin that is used by an external capacitor to set the bandwidth. Pin-7 is the analog output pin. Pin-8 is the power supply pin.

**6.6 Relay:-**

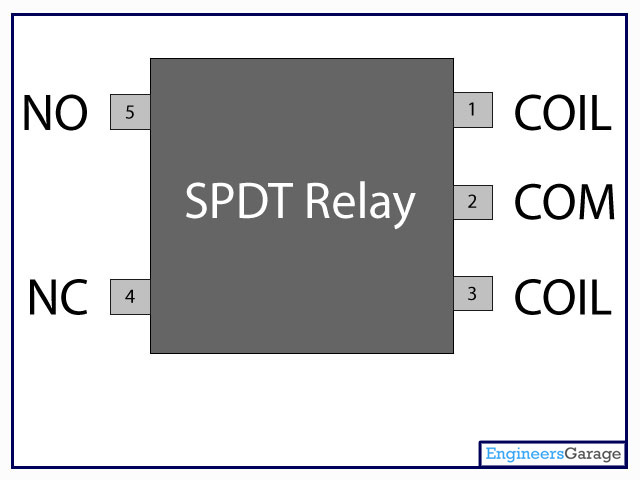
A relay is an electromagnetic switch that is used to turn on and turn off a circuit by a low power signal, or where several circuits must be controlled by one signal.

Most of the high end industrial application devices have relays for their effective working. Relays are simple switches which are operated both electrically and mechanically. Relays consist of an electromagnet and also a set of contacts. The switching mechanism is carried out with the help of the electromagnet. There are also other operating principles for its working. But they differ according to their applications. Most of the devices have the application of relays.



**Fig. 6.12: Relay**

### *6.6.1 Pin Diagram:-*



**Fig. 6.13: Relay Pin Diagram**

*6.6.2 Why is a relay used?*

The main operation of a relay comes in places where only a low-power signal can be used to control a circuit. It is also used in places where only one signal can be used to control a lot of circuits. The application of relays started during the invention of telephones. They played an important role in switching calls in telephone exchanges. They were also used in long distance telegraphy.

They were used to switch the signal coming from one source to another destination. After the invention of computers they were also used to perform Boolean and other logical operations. The high end applications of relays require high power to be driven by electric motors and so on. Such relays are called contactors.

* 1. **GPS:-**

Global Positioning System (GPS) is a satellite-based system that uses satellites and ground stations to measure and compute its position on Earth.

GPS is also known as Navigation System with Time and Ranging (NAVSTAR) GPS.

GPS receiver needs to receive data from at least 4 satellites for accuracy purpose. GPS receiver does not transmit any information to the satellites.

This GPS receiver is used in many applications like smartphones, Cabs, Fleet management etc.



**Fig. 6.13: GPS**

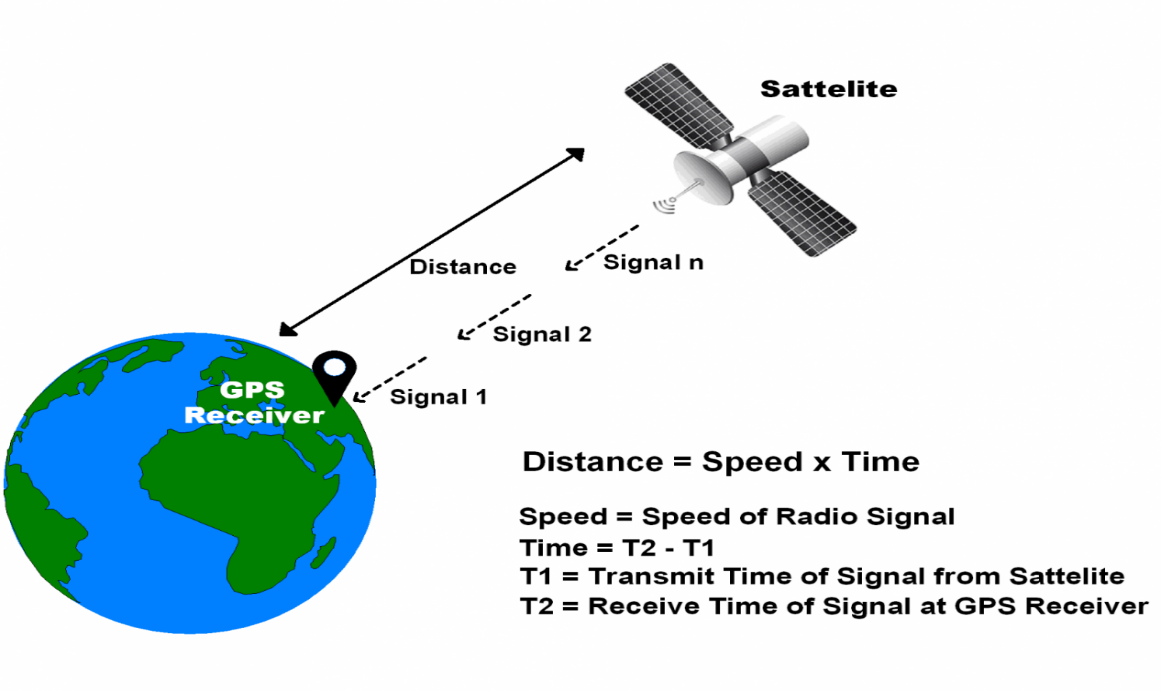
# ***6.7.1 How GPS Works:***

GPS receiver uses a constellation of satellites and ground stations to calculate accurate location wherever it is located.

These GPS satellites transmit information signal over radio frequency (1.1 to 1.5 GHz) to the receiver. With the help of this received information, a ground station or GPS module can compute its position and time.

***6.7.2 How GPS Receiver Calculates its Position and Time:-***

GPS receiver receives information signals from GPS satellites and calculates its distance from satellites. This is done by measuring the time required for the signal to travel from satellite to the receiver.



**Fig. 6.14: GPS Distance Calculation**

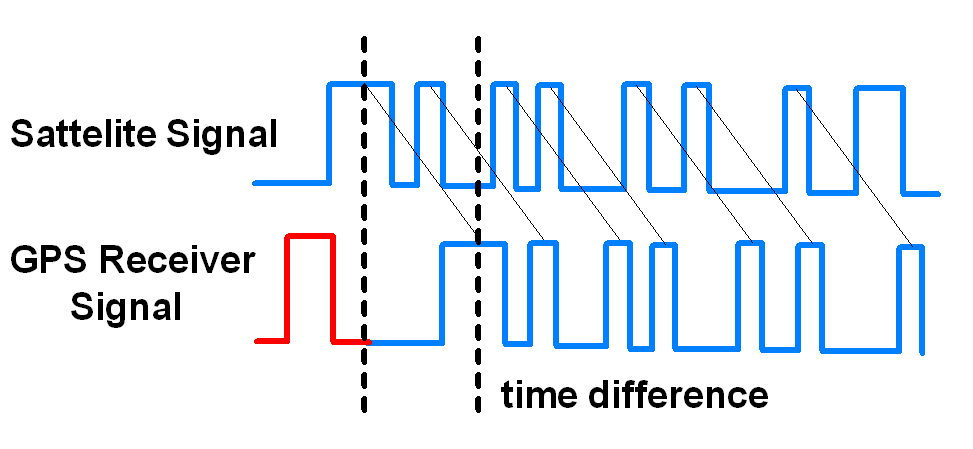
\textbf{Distance = Speed x Time}

Where,

Speed = Speed of Radio signal which is approximately equal to the speed of light i.e.3*10^{8}

Time = Time required for a signal to travel from the satellite to the receiver.

By subtracting the sent time from the received time, we can determine the travel time.



**Fig. 6.15: GPS Signal Time Difference**

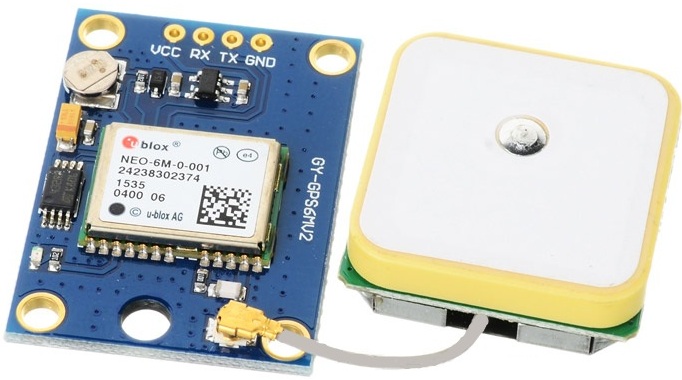
To determine distance, both the satellite and GPS receiver generate the same pseudocode signal at the same time.

The satellite transmits the pseudocode; which is received by the GPS receiver.

These two signals are compared and the difference between the signals is the travel time.

Now, if the receiver knows the distance from 3 or more satellites and their location (which is sent by the satellites), then it can calculate its location by using [Trilateration](http://electronics.howstuffworks.com/gadgets/travel/gps1.htm) method.

# **GPS Module:-**



**Fig. 6.16: GPS Receiver**

GPS receiver module gives output in standard (National Marine Electronics Association) NMEA string format. It provides output serially on Tx pin with default 9600 Baud rate.

This NMEA string output from GPS receiver contains different parameters separated by commas like longitude, latitude, altitude, time etc. Each string starts with ‘$’ and ends with carriage return/line feed sequence.

**E.g.**

$GPGGA,184237.000,1829.9639,N,07347.6174,E,1,05,2.1,607.1,M,-64.7,M,,0000\*7D

$GPGSA,A,3,15,25,18,26,12,,,,,,,,5.3,2.1,4.8\*36

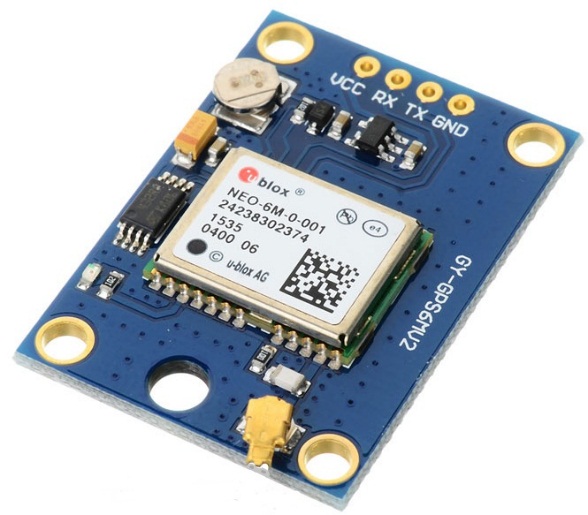
$GPGSV,3,1,11,15,47,133,46,25,44,226,45,18,37,238,45,26,34,087,40\*72

$GPGSV,3,2,11,12,27,184,45,24,02,164,26,29,58,349,,05,26,034,\*7F

$GPGSV,3,3,11,21,25,303,,02,11,071,,22,01,228,\*40

$GPRMC,184237.000,A,1829.9639,N,07347.6174,E,0.05,180.19,230514,,,A\*64

***6.8.1 Pin Description:-***



**Fig. 6.17: GPS Receiver Module**

**VCC:** Power Supply 3.3 – 6 V

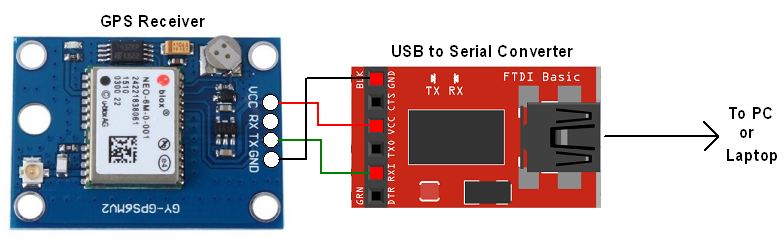
**GND:**Ground

**TX:**Transmit data serially which gives information about location, time etc.

**RX:**Receive Data serially. It is required when we want to configure GPS module.

# ***6.8.2 Check GPS module:-***

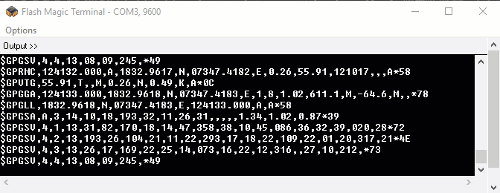
Before Interfacing GPS module with PIC18F4550 microcontroller, we can check the output of GPS module. From that string, we can extract information like longitude, latitude, time which is helpful to find location and timing information. To do this, connect this GPS module to the PC via USB to Serial converter or DB9 connector. Also, it is necessary to keep antenna of GPS module on proper location.



**Fig.6.18: GPS Serial Interface**

1. Now open any serial terminal e.g. Realterm, Hyper terminal, Putty etc. on PC/laptop.
2. Open the PORT with 9600 baud rate.
3. The terminal will show data coming from GPS receiver module.

The output data from GPS receiver module displaying on a serial terminal as follows.



In the above string, the NMEA string starting with “$GPGGA” is most popularly used. It provides us Time, Longitude, Latitude and Altitude along with directions. This information is helpful to find Time and Location.

1. **SOFTWARE REQUIREMENTS**

**7.1 Arduino IDE:-**

**Arduino IDE**where IDE stands for Integrated Development Environment – An official software introduced by Arduino.cc, that is mainly used for writing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go.

*7.1.1 Introduction to Arduino IDE:*

* Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.
* It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
* It is easily available for operating systems like MAC, Windows, and Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.
* A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, [Arduino Micro](https://www.theengineeringprojects.com/2018/09/introduction-to-arduino-micro.html) and many more.
* Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.
* The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.
* The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.
* This environment supports both C and C++ languages.

*7.1.2 How to install Arduino IDE:*

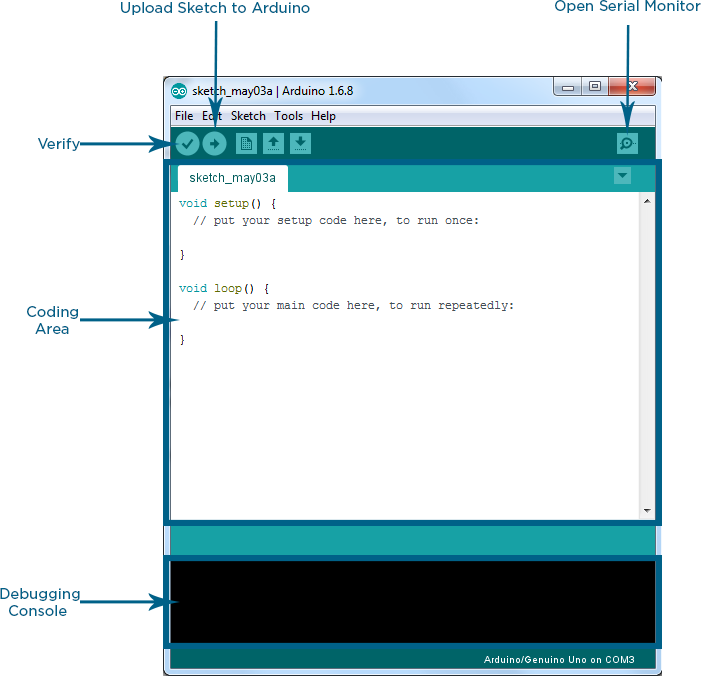
You can download the Software from [Arduino](https://www.arduino.cc/en/Main/Software) main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MAX, so make sure you are downloading the correct software version that is easily compatible with your operating system.

* If you aim to download Windows app version, make sure you have Windows 8.1 or Windows 10, as app version is not compatible with Windows 7 or older version of this operating system.

The IDE environment is mainly distributed into three sections

* **1. Menu Bar**
* **2. Text Editor**
* **3. Output Pane**

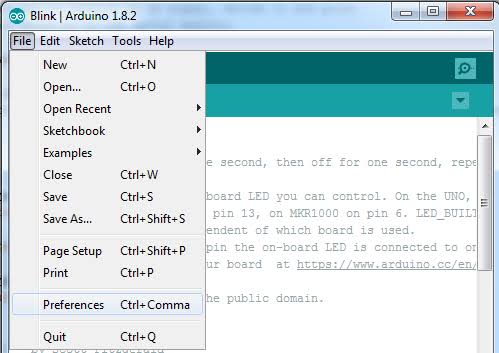
As you download and open the IDE software, it will appear like an image below.



**Fig.7.19: Arduino IDE**

The bar appearing on the top is called **Menu Bar** that comes with five different options as follow

* **File** – You can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into.

****

As you go to the preference section and check the compilation section, the Output Pane will show the code compilation as you click the upload button.

1. **SOURCE CODE**

int buttonState;

#include <OneWire.h>

#include <DallasTemperature.h>

// Data wire is plugged into port 2 on the Arduino

#define ONE\_WIRE\_BUS 2

// Setup a oneWire instance to communicate with any OneWire devices (not just Maxim/Dallas temperature ICs)

OneWire oneWire(ONE\_WIRE\_BUS);

// Pass our oneWire reference to Dallas Temperature.

DallasTemperature sensors(&oneWire);

/\*

\* The setup function. We only start the sensors here

\*/

#include <LiquidCrystal.h>

#define rs 8

#define en 9

#define d4 10

#define d5 11

#define d6 12

#define d7 13

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

#define vol A0

//#define cur A1

#define sw 5

#define buz 6

#define bulb 7

#include <SoftwareSerial.h>

#include <TinyGPS.h>

/\* This sample code demonstrates the normal use of a TinyGPS object.

It requires the use of SoftwareSerial, and assumes that you have a

4800-baud serial GPS device hooked up on pins 4(rx) and 3(tx).

\*/

float flat, flon;

TinyGPS gps;

SoftwareSerial ss(4, 3);

#include <SoftwareSerial.h>

//Create software serial object to communicate with SIM800L

//SoftwareSerial mySerial(5, 6); //SIM800L Tx & Rx is connected to Arduino #3 & #2

//String k;

String p,m,c,msg;

#include "ACS712.h"

ACS712 ACS(A1, 5.0, 1023, 100);

#include<EEPROM.h>

void setup()

{

Serial.begin(9600);

ss.begin(9600);

lcd.begin(16,2);

sensors.begin();

pinMode(vol,INPUT);

// pinMode(cur,INPUT)

pinMode(sw,INPUT);

pinMode(buz,OUTPUT);

digitalWrite(buz,LOW);

digitalWrite(bulb,OUTPUT);

digitalWrite(bulb,HIGH);

lcd.setCursor(0,0);

lcd.print("automatic");

lcd.setCursor(0,1);

lcd.print("fault detection");

}

void loop()

{

int value = analogRead(vol);

double voltage = map(value,0,1024,0,2500);

voltage/=100;

Serial.print("Voltage : ");

Serial.println(value);

delay(1000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("voltage :");

lcd.print(value);

delay(1000);

int mA = ACS.mA\_AC();

float A=mA\*0.001 ;

Serial.println("mA: ");

Serial.println(A);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("current :");

lcd.print(A);

delay(1000);

buttonState = digitalRead(sw);

Serial.println("sw : ");

Serial.println(buttonState);

delay(1000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("switch : ");

lcd.print(buttonState);

delay(1000);

sensors.requestTemperatures(); // Send the command to get temperatures

float tempC = sensors.getTempCByIndex(0);

delay(1000);

if (tempC != DEVICE\_DISCONNECTED\_C)

{

Serial.println("temperature:");

Serial.println(tempC);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Temperature : ");

lcd.print(tempC);

}

delay(1000);

if(value<100)

{

digitalWrite(bulb,LOW);

}

if(value>200)

{

digitalWrite(bulb,LOW);

}

if(buttonState==0)

{

digitalWrite(buz,HIGH);

Serial.println("buz high");

delay(3000);

digitalWrite(buz,LOW);

Serial.println("fault detected");

lcd.clear();

lcd.setCursor(0,0);

lcd.print("message sent");

msg="fault detected";

SEND();

}

else

{

//digitalWrite(bulb,LOW);

digitalWrite(buz,HIGH);

}

if(tempC>=32)

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("tempararture high");

digitalWrite(buz,HIGH);

Serial.println("buz high");

delay(3000);

digitalWrite(buz,LOW);

Serial.println("temprature high");

msg="temparature high";

SEND();

lcd.clear();

lcd.setCursor(0,0);

lcd.print("message sent");

}

else

{

digitalWrite(buz,LOW);

}

if (ss.available() > 0)

{

//Serial.write(mySerial.read());//Forward what Software Serial received to Serial Port

p = ss.readString();

Serial.println(p);

}

bool newData = false;

unsigned long chars;

unsigned short sentences, failed;

// For one second we parse GPS data and report some key values

for (unsigned long start = millis(); millis() - start < 1000;)

{

while (Serial.available())

{

char c = Serial.read();

// Serial.write(c); // uncomment this line if you want to see the GPS data flowing

if (gps.encode(c)) // Did a new valid sentence come in?

newData = true;

}

}

if (newData)

{

unsigned long age;

gps.f\_get\_position(&flat, &flon, &age);

Serial.print("LAT=");

Serial.print(flat == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flat, 6);

Serial.print(" LON=");

Serial.println(flon == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flon, 6);

}

}

void SEND()

{

ss.println("Setting the GSM in text mode");

ss.println("AT+CMGF=1\r");

delay(2000);

ss.println("Sending SMS to the desired phone number!");

ss.println("AT+CMGS=\"+917013475421\"\r");

// Replace x with mobile number

delay(2000);

ss.println("lat : "); // SMS Text

ss.println(flat);

ss.println("long : ");

ss.println(flon);

ss.println(msg);

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" LAT : ");

lcd.print(flat);

lcd.setCursor(0,1);

lcd.print(" LNG : ");

lcd.print(flon);

delay(2000);

ss.write(26);

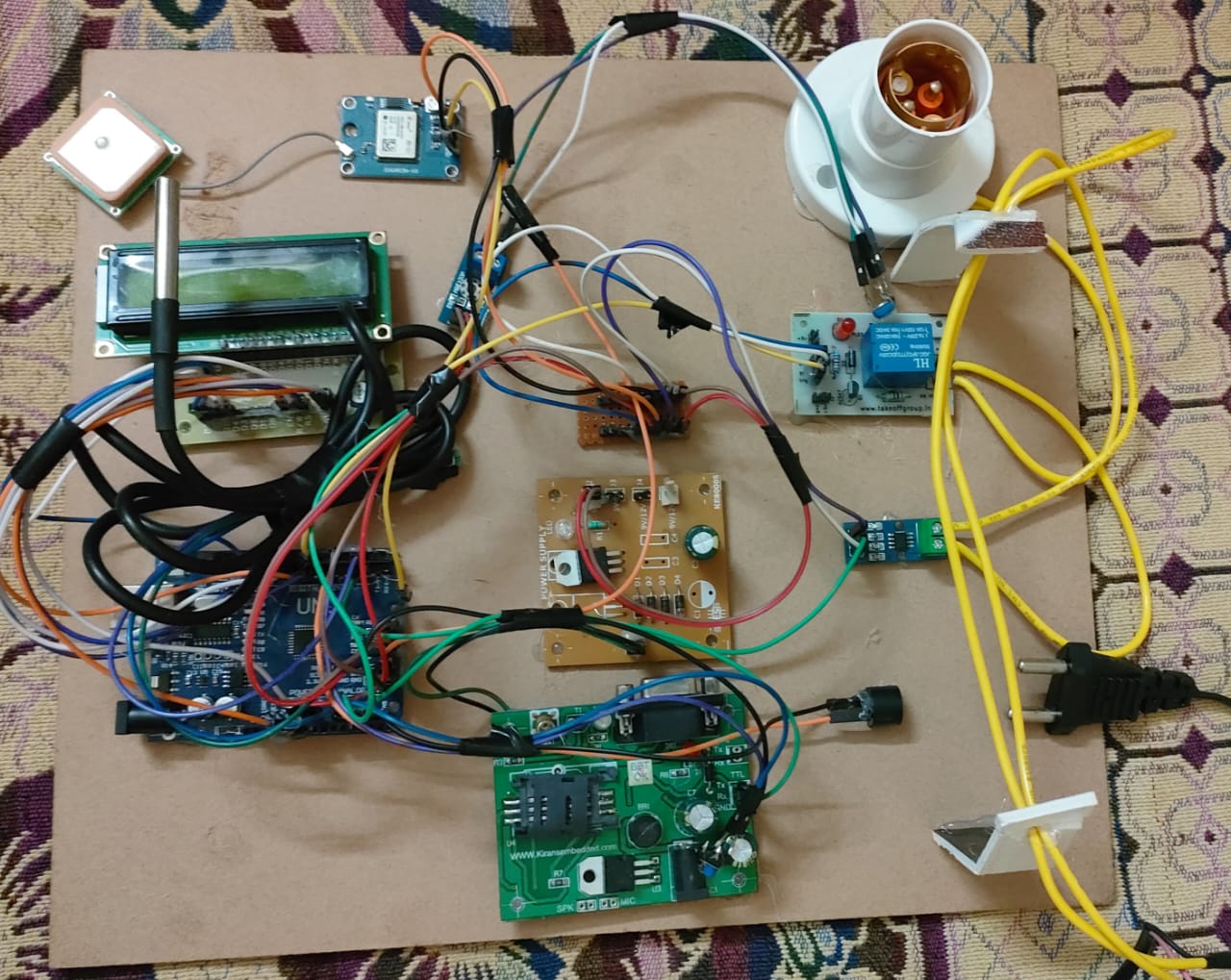
Serial.println("DONE..!!");

delay(3000);

}

1. **WORKING PRINCIPLE**

To attain our concept, need to use Arduino, voltage sensor, current sensor, speed sensor, buzzer, temperature sensor, relay, LCD. The project is assembled with a set of resistors representing cable length in KMs and fault creation is made by a set of switches at every known KM to cross check the accuracy of the same. The voltage drop across the feeder resistor is given to an ADC which develops a precise digital data which the programmed microcontroller would display the same in Kilo meters. The fault occurring at what distance and which phase is displayed on a 16X2 LCD interfaced with the microcontroller. If the temperature higher than the threshold value at that time buzzer and LCD will give intimation. Calculated values are sends to the internet with help of IOT.RTC is used here to time and date reference, that when the event occurs.



**Fig.9.20: Circuit Diagram**

****

**Fig.9.21: Circuit Diagram-2**

Whenever the fault takes place in the distribution line on the Pin insulator or the breaking of the conductor in the distribution line the Current Transformer will sense that there is break in the line or interruption of the signal, Immediately it will proceed the signal to Analog to Digital Converter from there the signal is given to the Raspberry-Pi Microprocessor. Raspberry-Pi Processor immediately sends the signal through cloud to the Operator operating in the control room and Linemen of that respective area on his respective mobile phone giving them the exact location of the fault. After getting the exact fault location the fault is corrected immediately. Thus saving much of the time of identification of fault location and the consumers will get the uninterrupted power supply.

1. **ADVANTAGEs**

* Devices are enabled by wireless communication.
* Coverage area is large compared to the existing system.
* Less number of components and manual observation. So, it is economically reliable and low cost.

1. **APPLICATIONS**

* Used in transmission line.
* Used in textile mills.
* Used in food industry.

1. **CONCLUSION**

The model design in such a way to solve the problems faced by consumer. By using such method, we can easily detect the fault and resolve it. It is highly reliable and locate the fault in three phase transmission line and also supposed to data storage. It works on real time so we maintain all data sheet and avoid the future problem in transmission line.

1. **REFERENCES**

[1] PitukBunnoon, “Fault Detaction Aprroaches to Power System : State-of-the Art Article Review for Searching a New Approach in the Future,” International Journal of Electrical and Computer Engineering, vol. 3, No. 4, pp 553- 560, August 2017.

[2] P. Chandra shekar., “Transmiision Line Fault Detection & Indication through GSM,” Internation Journal of Recent Advancs in Engineering & Technology , vol. 2, issue 5, pp 28- 30, 2019.

[3] S.Chavhan, V.Barsagade, A.Dutta, S.Thakre., “ Fault Detection in Power Line using Wireless Sensor Networks,” IPASJ Internaional Journal of Electrical Engineering , vol. 3, issue 3, pp 8-13, March 2018.

[4] Ing. KomiAgbesi, Felix AttuquayeOkai., “Automatic Fault Detection and Location in Power Transmission Lines using GSM Technology,” Internation Journal of Advance Reasearch in Science and Engineering , vol. 5, issue 1, pp 193-207, January 2016.

[5] Manohar Singh, Dr.B.K.Panigrahi, Dr.R.P.Maheswari, „‟ Transmission Line Fault Detection and Classification”, Proceedings of ICETECT, 2020.

[6] NwekeChisom B., IroegbuChibuisi, OgeChikanmaIhekweaba, Henkwe Clement E., “ Using GSM to Detect Fault in Microcontroller Based Power Transformer,” Internaional Journal for Research in Applied Science and Engineering Technology, vol. 2, issue 8, pp 271-274, August 2019.

[7] “FAULT DETECTION IN TRANSMISSION LINE USING ARDUINO UNO” Saurabh Rathod\*1, Abhishek Ghar\*2, Pavan Palve\*3, Mayur Patil\*4, Prof. N. V. Khadse\*5 \*1,2,3,4Electrical Engineering Department,Government College Of Engineering Nagpur, Nagpur, Maharashtra, India. \*5Assistant Professor Electrical Engineering Department, Government College Of Engineering Nagpur, Nagpur, Maharashtra, India., 2022.

[8] “Transmission Line Falling Detection using IOT” A. Jothimani1 , Sourabh Sahil2 , Durgesh Sahu3 , Mayank Mishra4 Department of Electronics & Communication Engineering SRM Institute of Science and Technologies, Ramapuram Campus, Chennai, India, 2020.

[9] “IOT BASED TRANSMISSION LINE FAULT MONITORING SYSTEM” R Navaneetha Krishna1 , Niranjan L2 , N Shyamsundar3 , Dr. C Venkatesan4 1, 4 Department of EEE, RRIT Bangalore 560090, 2020.

[10] “Fault Detection in Overhead Lines using IoT” Dr. A Radhika[1] , K Sathya Kumar[2] , M Rajeshvara Boopathi[3] , K V Priyadharshan[4] 1Associate Professor, Electrical and Electronics Engineering, Velammal College of Engineering and Technology, Madurai, Tamil Nadu, July/2020.

[11] “Transmission Line Fault Detection using IoT” Minal Karalkar1 , Sushrut Adlok2 , Ankit Ramteke3 , Sumit Parchake4 , Rajat Tawade5 , Sujata Manwar6 , Shubham Wankhade7 1Assistant Professor, Department of Electrical Engineering. Priyadarshini J. L. College of Engg., Nagpur, India, 2019.

[12] “Automatic Fault Detection and Location of Transmission Lines using IoT” 1Sajal Menon 2Don Tommey 3Rejoice Thomas Paul 4Krishnapriya Vinod 5 Rajalakshmi Menon 1,2B.Tech Student 3,4,5PG Scholar 1,2,3,4,5ASIET,Kalady, 2020.