**IoT-Based Automatic Detection and Localization of Transmission Line Faults**

**Dr.RAJAN.V.R**

**Assistant Professor**

**ABSTRACT**

A sophisticated fault detection and localization system, utilizing GSM technology, was implemented to precisely identify and pinpoint the occurrence of faults. This expedites the response time of technical teams, facilitating swift resolution of issues and safeguarding transformers from potential damage and emergencies. The setup incorporates key components such as a current transformer, voltage transformer, Arduino microcontroller, current detector, voltage sensing circuit (for overload scenarios), and a GSM modem.

Upon detection, the fault details are communicated to the central control room. In summary, this system significantly reduces fault localization time, delivering automated and precise fault location information. Through this project, it becomes possible to detect faults in three-phase transmission lines, while simultaneously monitoring factors like Temperature, Voltage, and Current via a GSM modem through message communication.

The importance of detecting and locating power faults cannot be overstated for the smooth operation of power systems. The proposed approach divides the power line into segments using a wireless sensor network. These nodes are capable of detecting faulty conditions, displaying them to operators, and triggering SMS alerts through a GSM modem, ensuring prompt attention from service engineers.

1. **INTRODUCTION**

The transmission network stands as a pivotal component within the power system. Notably, the transmission and distribution network registers considerable losses in comparison to other segments of the power system. The electric power infrastructure remains exceptionally susceptible to a range of natural and deliberate physical incidents. Traditional approaches in many electric power transmission companies have primarily relied upon circuit indicators to identify faulty segments along their transmission lines.

The adoption of wireless sensor-based monitoring for transmission lines presents a remedy to address numerous challenges. These include real-time structural awareness, expedited fault localization, accurate fault diagnosis achieved through distinguishing electrical faults from mechanical ones, reduced costs attributed to condition-based maintenance rather than routine servicing, among others. Such applications impose stringent demands, encompassing swift transmission of vast volumes of highly dependable data.

The triumph of these applications hinges upon the formulation of an economical and dependable network architecture, characterized by rapid responsiveness. This network must seamlessly convey delicate information, encompassing the real-time state of the transmission line and pivotal control instructions, to and from the transmission grid. This project introduces a cost-optimized framework tailored for the creation of a real-time data transmission network. To achieve real-time monitoring of the power system's status, sensors are strategically positioned across various components within the power network.

1. **EMBEDDED SYSTEM IMPLEMENTATION**

**2.1 Introduction:**

An embedded system is a specialized computer system meticulously crafted to execute diverse tasks such as data access, processing, storage, and data control within various electronics-driven environments. These systems amalgamate hardware and software components, with the software commonly referred to as firmware, which becomes an integral part of the hardware structure. Notably, one of their pivotal attributes is their capacity to deliver outputs within specified time constraints.

Embedded systems play a pivotal role in enhancing precision and convenience within various functionalities. Consequently, these systems find extensive application in both straightforward and intricate devices. Their utility spans across an array of real-life scenarios, encompassing commonplace objects like microwaves, calculators, TV remote controls, as well as more intricate systems like home security setups and neighborhood traffic control systems..

User interface

Embedded system

Hardware

Software

Inputs

Output

Link to other systems

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

* 1. **Embedded system:**

The embedded system comprises primarily of two key sections:

1. Hardware

2. Software

Power Source and Oscillation Circuitry

Timers

Processor

Integration of Input Devices and Driver Circuits

Memory

Application specific circuits

Serial communication ports

Interrupt Handling Unit

Parallel ports

Output devices interfacing

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

* 1. **Embedded System Hardware:**

Like any electronic system, an embedded system requires a foundational hardware framework to execute its operations. This hardware includes a microprocessor or microcontroller as its core. Critical elements within embedded system hardware consist of input-output (I/O) interfaces, user interfaces, memory, and display functionalities. Broadly, an embedded system encompasses various components, including a power supply, processor, memory, timers, serial communication ports, input/output circuits, and circuits tailored to specific system applications.

Embedded systems rely on specialized processors suited for their specific tasks, including microprocessors, microcontrollers, and digital signal processors (DSPs). Microprocessors function as central processing units contained on a single chip, capable of integrating substantial memory and I/O ports, albeit at a higher cost due to the need for external peripherals; they are generally larger and intended for diverse general-purpose applications. In contrast, microcontrollers encompass full computer systems within a chip, offering fixed on-chip memory and I/O ports, often at a more economical price point and in a compact size, making them suitable for specific-purpose functions.

1. **LITERATURE SURVEY**

S. Lefebvre [1] and colleagues illustrated the advantages of adaptable adjustment for current controllers within the context of an HVDC converter system, tailored to meet specific system requirements. The study demonstrates that the converter's Silicon-Controlled Rectifier (SCR) or the compensation impedance provided by the converter stands as the critical parameter influencing the tuning process. The core HVDC system is linearized around an operational point, and the controller ensures that the system's eigenvalues and zeros are maintained at predetermined positions for each set of system parameter variations. Determination of these parameter variations is accomplished by subjecting the system to a minor noise signal. The estimation and control procedures are conducted at different transmission rates to enhance the controller's robustness.

John Reeve [2] and his team endeavored to integrate adaptive gain scheduling control into the rapid control loops of DC transmission systems. This integration aimed to enhance the system's performance in the face of significant disturbances, low Effective Short Circuit Ratios, and deficiencies that further reduce the system's operational Silicon-Controlled Rectifier (SCR). Various scheduling factors such as DC current error signal, DC voltage error signal, AC voltage zero crossings, and rectifier firing point were selected based on the system's response to substantial AC disturbances. These factors were chosen for each scenario to bolster the system's resilience against major disruptions, guided by the outcomes derived from feasibility indicators.

John Reeve [3] and colleagues combined the theoretical concept of auto-tuning with gain scheduling, focusing on two key points. Firstly, they explored whether auto-tuning offers significant advantages over fixed gains, such as standard controller gain scheduling. Secondly, they investigated whether a combination of both approaches could enhance the robustness of the controllers when dealing with significant disturbances. The study demonstrated that in certain scenarios, like continuous or sudden low short-circuit ratios, relying solely on auto-tuning might not provide enough reliability due to specific disturbances. Therefore, a combination of auto-tuning and gain scheduling was found to be more effective.

P.K. Dash et al [4] introduced a practical control approach for an HVDC system based on input linearization principles. They employed a neural estimation algorithm to track the linearized control parameters. These parameters depend on factors like rectifier-side DC voltage, inverter-side DC voltage, DC link reactance, and equivalent resistance. Various transient conditions were applied to the DC link to demonstrate the controller's performance. An improved error-tracking strategy enhances the controller's dynamic stability.

In their work, A. Routary et al. [5] introduced a fuzzy self-tuning controller to replace the rectifier-side current controller. This new controller uses a fuzzy interface to adjust the gain, which involves the proportional (Kp) and integral (KI) controller constants. By utilizing linearized measurements of current error and its derivative, the controller calculates the firing angle at the rectifier end and determines the extinction angle at the inverter end. Through evaluations conducted under transient conditions, the study shows that this approach effectively tunes the PI controller parameters using fuzzy logic, enhancing the system's performance.

In the study by Chi-Hshing Lin [6], the focus was on contrasting two common faults within an HVDC interface. Specifically, the research delved into ignition failures occurring in rectifier valves and inverter valves. Through dynamic simulation, distinct patterns emerged for these scenarios. When a rectifier valve experiences ignition failure, it induces substantial torsional torque in a turbine generator, especially if power fluctuations disrupt natural torsional modes. This, in turn, affects the frequency of the rectifier-side system. Conversely, an ignition failure in an inverter valve results in compensation failure within converters, consequently leading to HVDC system breakdown. Notably, this issue directly impacts both the rectifier and inverter aspects of the generator.

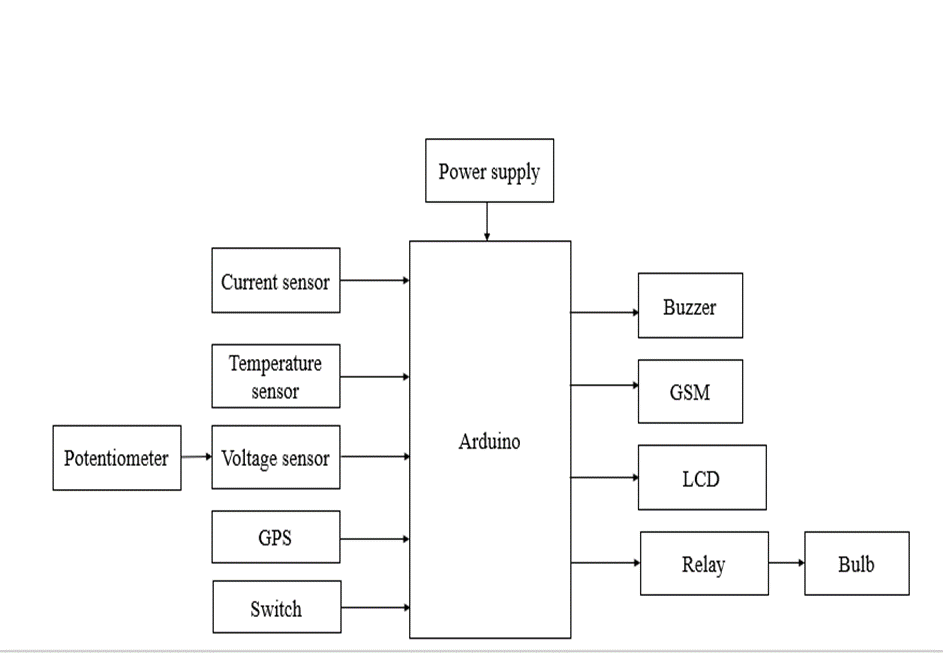
1. **EXISTING SYSTEM**

The current system relies on nodes for fault detection, but it falls short in providing clear information about the nature of the fault that has occurred, be it a short circuit, wire cut, or overload. To rectify this limitation, we propose an innovative system that not only enhances accuracy but also reduces the overall cost and complexity of implementation.

The drawbacks of the existing system stem from its inability to accurately pinpoint the exact fault type and its associated challenges. Additionally, the high costs associated with the current approach further underscore the need for a more efficient and cost-effective solution.

1. **PROPOSED SYSTEM**

In our proposal, we introduce a system that incorporates an Arduino, a current detector, a voltage sensing circuit (primarily for overload scenarios), and a GSM modem. This system operates by autonomously identifying faults, analyzing and categorizing them accordingly. The system then proceeds to transmit the fault data to a central control room. Ultimately, this integration significantly minimizes the time needed to pinpoint a fault, as the system not only detects faults but also precisely determines their location. Furthermore, an LCD screen is incorporated to visually display the specific type of fault detected.

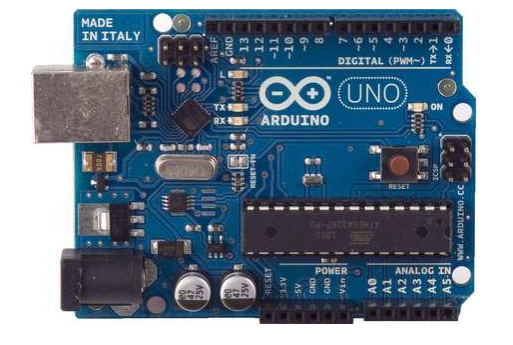


**Fig. 5.3:- Block diagram**

1. **HARDWARE REQUIREMENTS**

**6.1. ARDUINO:**

The Arduino microcontroller is a simple yet powerful computer on a single board. It's become quite popular in both hobbyist and professional circles. What's great is that the Arduino is open-source, so the hardware is affordable, and you can access the software for free. This guide is specifically aimed at students in ME 2011 or any newcomers to the Arduino world. For those who are more experienced with Arduino, you can explore various online resources. Here's a glimpse of what the Arduino board looks like:



**Fig. 6.4:- Arduino**

The Arduino programming language presents a simplified variant of C/C++, making it comfortable for those familiar with C and approachable for newcomers to C due to its requirement for only a few fundamental commands. Turning attention to the Atmega328p, it emerges as a high-performance, low-power AVR 8-bit microcontroller, employing an advanced RISC architecture comprising 131 efficient instructions that execute within a single clock cycle. It boasts 32 x 8 general-purpose working registers and operates fully in a static mode, achieving up to 20 MIPS throughput at 20 MHz. The microcontroller integrates a 2-cycle on-chip multiplier and showcases high endurance non-volatile memory segments. This device provides in-system self-programmable flash program memory (ranging from 4 to 32K bytes), EEPROM storage (ranging from 256 to 1K bytes), and internal SRAM (ranging from 512 to 2K bytes). With 10,000 flash/100,000 EEPROM write/erase cycles, it retains data for 20 years at 85°C or 100 years at 25°C. It supports on-chip boot programming and enables true read-while-write operation. The Atmega328p accommodates interrupts and wake-up on pin change and supports various sleep modes such as idle, ADC noise reduction, power-save, power-down, standby, and extended standby. It functions reliably within a voltage range of 1.8 to 5.5V and operates effectively across a temperature span of -40°C to 85°C. This microcontroller offers a speed range of 0 to 20 MHz at 1.8 to 5.5V and demonstrates low power consumption at 1 MHz, 1.8V, and 25°C.

Regarding power supply:

The Arduino Uno is capable of being powered through either a USB connection or an external source, with the appropriate power source selected automatically. When using a non-USB power option, such as an AC-to-DC adapter or a battery, the AC adapter should be connected to the power jack, while a battery can be linked to the Gnd and Vin pins of the POWER connector. The board's operational range with an external supply is 6 to 20 volts. It's important to note that if the supply voltage drops below 7V, the 5V pin's output might fall below five volts, leading to instability. Conversely, if the voltage surpasses 12V, the voltage regulator could potentially overheat, causing damage to the board. The recommended voltage range for safe operation is between 7 and 12 volts. The board features several key pins, including VIN for external power input, 5V for regulated 5V output from the onboard regulator, 3V3 for supplying 3.3 volts with a maximum current draw of 50 mA, and GND pins for grounding purposes.

For both input and output capabilities, the Arduino Uno boasts 14 digital pins, all of which can be configured as either inputs or outputs through the use of pinMode(), digitalWrite(), and digitalRead() functions. These pins operate at a voltage of 5 volts and can handle a maximum current of 40 mA. Additionally, each pin comes equipped with an internal pull-up resistor, ranging between 20 to 50 kOhms. Several pins hold specific functionalities: pins 0 (RX) and 1 (TX) manage TTL serial data, pins 2 and 3 are designated for external interrupts, enabling configuration for low triggers, rising or falling edges, and value changes. PWM output is achievable through pins 3, 5, 6, 9, 10, and 11, offering 8-bit PWM using analogWrite(). SPI communication is supported by pins 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK). Lastly, the onboard LED is associated with digital pin 13.

Within its configuration, the Uno incorporates 6 analog inputs (A0 to A5) with a 10-bit resolution, designed to measure values from ground to 5 volts as their default range. However, their upper range can be adjusted by utilizing the AREF pin in conjunction with the analogReference() function. Some pins serve specialized purposes: A4 or SDA and A5 or SCL are dedicated to TWI communication. Additionally, there are two more pivotal pins on the board: AREF, acting as the reference voltage source for analog inputs, and Reset, which, when set to low, initiates a reset of the microcontroller.

Remember, for more detailed pin mapping, refer to the Arduino pins and ATmega328 ports mapping. This mapping is consistent for Atmega8, 168, and 328 microcontrollers."

* 1. **LCD:-**

"LCD, which stands for Liquid Crystal Display, is a technology found in laptops and smaller computers. It's used in displays for its thinness, consuming less power than other types of displays like LED or gas displays. Instead of emitting light, LCDs work by blocking light. There are two types of LCDs: passive matrix and active matrix. Active matrix, also known as thin film transistor (TFT) display, has a transistor at each pixel intersection, needing less current to control brightness.

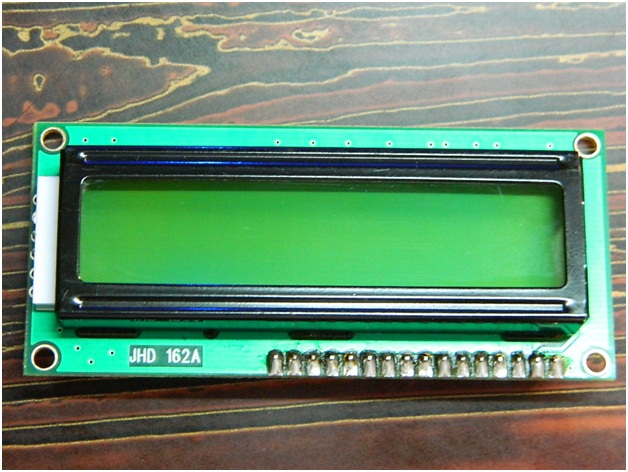
Now, let's talk about how LCDs handle data, signals, and execution:

LCDs accept two types of signals: data and control. These signals are identified by the LCD module through the RS pin status. By pulling the R/W pin high, data can also be read from the LCD display. When the E pin is pulsed, the LCD reads and processes data at the pulse's falling edge, as well as in the transmission case.

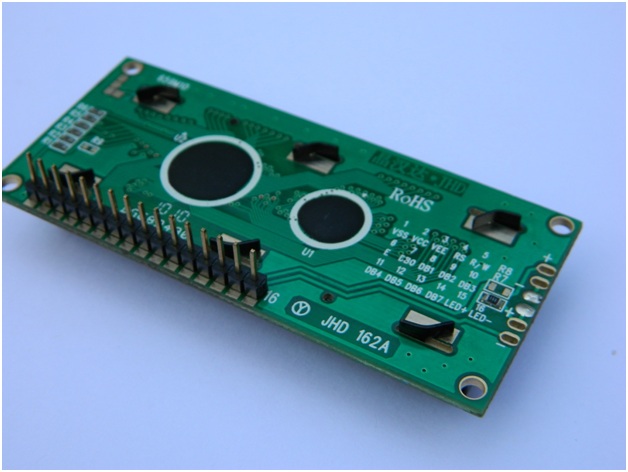
It takes around 39-43 microseconds for the LCD display to position a character or execute a command. For certain actions like clearing the display or moving the cursor to the home position, it takes 1.53 to 1.64 milliseconds. Trying to send data before this time might lead to reading failures or improper execution.

LCD displays have two types of RAM: DDRAM and CGRAM. DDRAM keeps track of where characters appear on the screen based on the ASCII chart. Each DDRAM byte represents a position on the LCD screen. The LCD controller reads DDRAM data and displays it on the screen. CGRAM allows users to create custom characters. Users can display these personalized characters on the LCD screen once CGRAM is configured.

*6.2.2. Images of LCD Display:-*

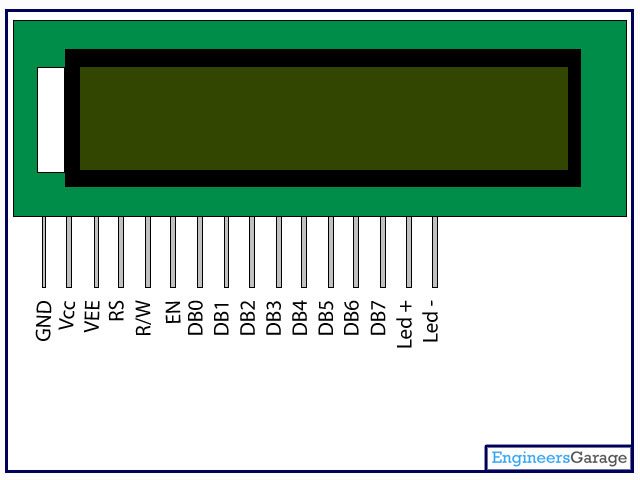
[](http://www.circuitstoday.com/wp-content/uploads/2012/02/LCD-Display-Front-Side.jpg)

**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:*

In the LCD, there exist predefined commands that must be conveyed to the LCD through a microcontroller. Here are several crucial 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:**

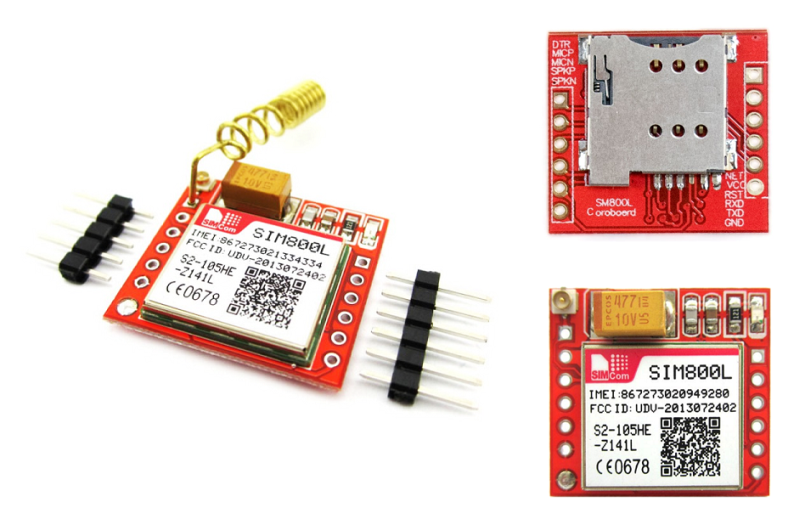
GSM, short for Global System for Mobile Communication, is a mobile communication technology developed at Bell Laboratories in 1970. It's a widely used system for transmitting mobile voice and data services, operating at frequencies like 850MHz, 900MHz, 1800MHz, and 1900MHz.

GSM uses digital technology with a technique called time division multiple access (TDMA) for communication. It digitizes and sends data in time slots through a channel. This digital system can handle data rates from 64 kbps to 120 Mbps.

GSM networks have different cell sizes: macro, micro, pico, and umbrella cells, each tailored to specific environments. The architecture consists of components like the Mobile Station (a mobile phone), Base Station Subsystem (linking mobile stations and the network), and Network Subsystem (providing network connections).

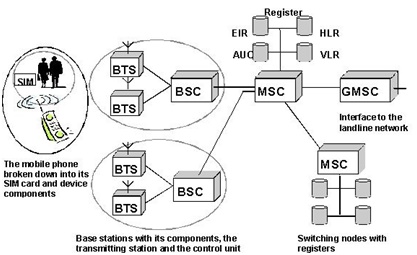
Features of GSM modules include improved spectrum efficiency, international roaming, compatibility with services like ISDN, support for new services, SIM phonebook management, and more. GSM ensures secure communication through encryption and also offers features like short message service (SMS).

A GSM modem is a device that enables communication over a network. It can be a mobile phone or a separate device. It requires a SIM card and can connect to computers through serial, USB, or Bluetooth connections. GSM modems have various applications like transaction terminals, security, weather stations, and remote data logging."

****

**Fig. 6.8: GSM**

The system uses a SIM (Subscriber Identity Module) card similar to mobile phones to establish network communication, featuring an IMEI (International Mobile Equipment Identity) number for identification. The GSM/GPRS modem facilitates tasks like managing SMS messages, handling phonebook entries on the SIM card, and engaging in voice calls. Communication with the modem involves AT commands conveyed via serial communication, allowing interaction with the GSM and GPRS cellular network. The GSM architecture encompasses the Radio Subsystem, Network and Switching Subsystem, and Operation Subsystem, with the radio subsystem containing the Mobile Station and Base Station Subsystem. The mobile station, often a phone, houses a transceiver, display, and processor, and it embeds a distinct identity within a SIM (Subscriber Identity Chip) module—a microchip storing mobile station data.



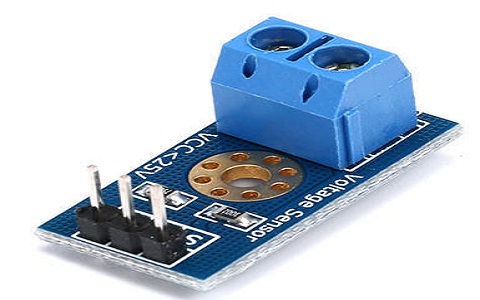
**Fig. 6.9: GSM Architecture**

## **Voltage Sensor:-**

This sensor is utilized for monitoring, calculating, and determining voltage supply levels. It can discern both AC and DC voltage levels. The sensor takes voltage as input and provides outputs such as switches, analog voltage signals, current signals, or audible signals. Some sensors yield sine or pulse waveforms, while others generate outputs like AM (Amplitude Modulation), PWM (Pulse Width Modulation), or FM (Frequency Modulation). These sensors' measurements can rely on voltage dividers.

.**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 features both input and output functionalities. On the input side, it comprises two pins: positive and negative. These pins can be connected to the corresponding positive and negative pins on the sensor. Similarly, the positive and negative pins of the device can be linked to the positive and negative pins of the sensor. The sensor's output consists of supply voltage (Vcc), ground (GND), and analog output data.

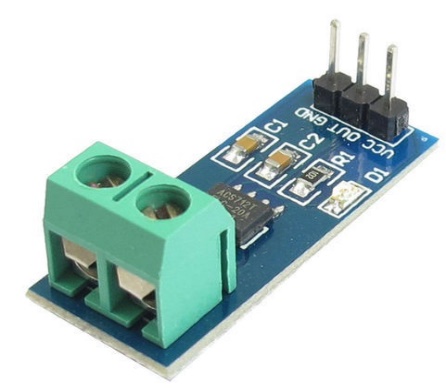
## **ACS712 Current Sensor:-**

Electric current passing through a conductor results in a decrease in voltage, as described by Ohm's law. When dealing with electronic devices, exceeding the necessary current can result in overload, potentially harming the device.

Accurate current measurement is crucial for device functionality. Voltage measurement is a non-intrusive task that can be carried out without impacting the system. However, current measurement is considered intrusive, as it cannot be directly detected like voltage.

**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**

To measure current in a circuit, a specialized device known as the ACS712 Current Sensor is utilized. This sensor accurately gauges and computes the current passing through a conductor without disrupting the system's performance.

The ACS712 Current Sensor is a fully integrated sensor IC that employs Hall-effect technology. It possesses a voltage isolation of 2.1kV RMS and features a low-resistance current conductor.

*Working Principle:*

The ACS712 Current Sensor functions by detecting the current passing through a wire or conductor and generating a corresponding signal representing the measured current. This signal can be in the form of analog voltage or digital output. Current sensing can be approached in two ways: Direct Sensing and Indirect Sensing.

In Direct Sensing, voltage drop caused by current flow across a wire is measured using Ohm's law. In contrast, Indirect Sensing involves assessing the magnetic field created by a current-carrying conductor, with Faraday's or Ampere's laws quantifying this magnetic field. Various components such as transformers, Hall effect sensors, or fiberoptic current sensors are employed to sense this magnetic field.

The ACS712 Current Sensor utilizes the Indirect Sensing technique. It incorporates a low-offset Hall sensor circuit positioned on a copper conduction path on the integrated circuit's surface. When current flows through this copper path, it generates a magnetic field, which is then detected by the Hall effect sensor. The sensor subsequently produces a voltage proportional to the magnetic field, thereby facilitating accurate current measurement.

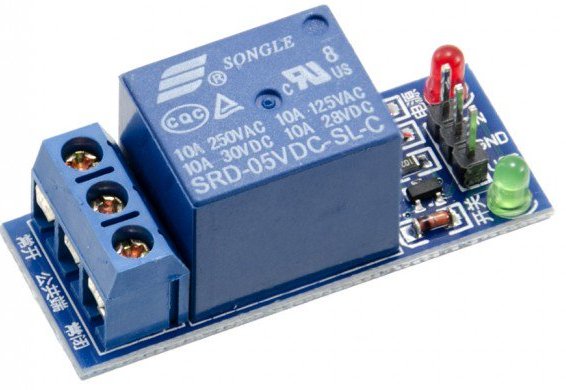
The precision of the device relies on the closeness of the magnetic signal to the Hall sensor, as closer signals result in enhanced accuracy. Encased in a compact SOIC8 package, the ACS712 Current Sensor features a conduction path formed by Pin-1 to Pin-4. Operating at a 5V supply voltage, it generates an output voltage proportional to both AC and DC current. Additionally, the sensor boasts minimal magnetic hysteresis, further enhancing its performance.

Among its pins, Pin-5 serves as the signal ground, Pin-6 as the FILTER pin to set bandwidth through an external capacitor, Pin-7 as the analog output, and Pin-8 as the power supply. Notably, this IC doesn't require additional isolation techniques as its conduction path terminals are electrically isolated from the IC leads.

**6.6 Relay:-**

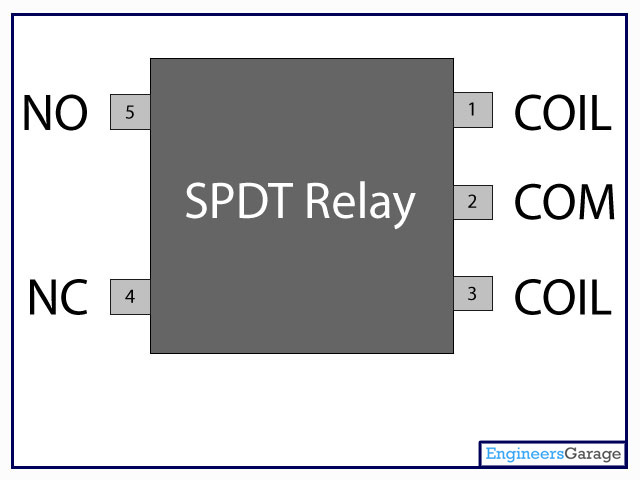
A relay serves as an electromagnetic switch employed to activate or deactivate a circuit using a low-power signal. It's especially useful when managing multiple circuits with a single signal.

In numerous advanced industrial devices, relays play a crucial role in ensuring optimal functionality. These relays function as basic switches that can be operated both electrically and mechanically. Each relay comprises an electromagnet along with a set of contacts. The switching action is executed through the electromagnet's influence. Different operational principles can be utilized, varying based on specific applications. Relays find extensive application across a multitude of devices.



**Fig. 6.12: Relay**

### *6.6.1 Pin Diagram:-*



**Fig. 6.13: Relay Pin Diagram**

6.6.2 Reasons for Using Relays

Relays find their primary use in scenarios where a circuit needs to be controlled using a low-power signal. They also come into play when a single signal is required to manage multiple circuits. The origins of relay applications trace back to the early days of telephone development. In telephone exchanges, relays played a pivotal role in call switching. They were similarly employed in long-distance telegraph communication.

The core purpose of relays was to redirect signals from one source to another destination. With the advent of computers, they found applications in executing Boolean and other logical operations. In more advanced applications, where higher power is needed to drive components like electric motors, specialized relays known as contactors are used.

* 1. **GPS:-**

The Global Positioning System (GPS) uses satellites and ground stations to determine its position on Earth. It's also known as Navigation System with Time and Ranging (NAVSTAR) GPS. For accuracy, a GPS receiver needs signals from at least 4 satellites. Notably, a GPS receiver doesn't send data to satellites. These receivers find applications in various areas like smartphones, taxi services, and fleet management systems.



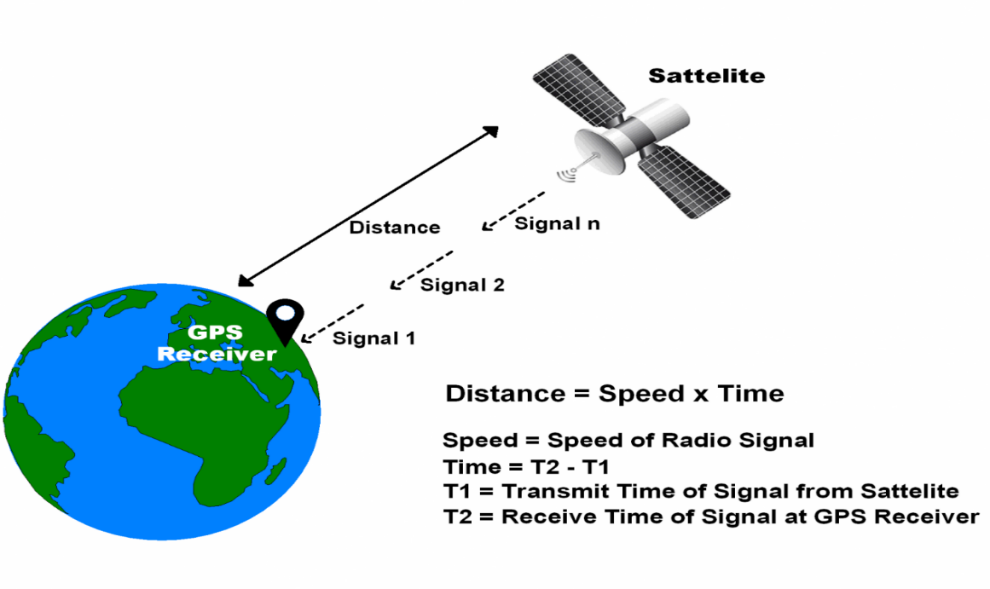
**Fig. 6.13: GPS**

*6.7.1 Understanding GPS Operation:*

GPS receivers make use of an extensive network of satellites and ground stations to precisely determine their geographical coordinates, regardless of their current location. These GPS satellites transmit information signals via radio frequencies, typically falling within the range of 1.1 to 1.5 GHz, which are received by the GPS receiver. By processing this received data, a ground station or GPS module can accurately compute its specific position and the precise time.

6.7.2 Computation of Position and Time by GPS Receiver:

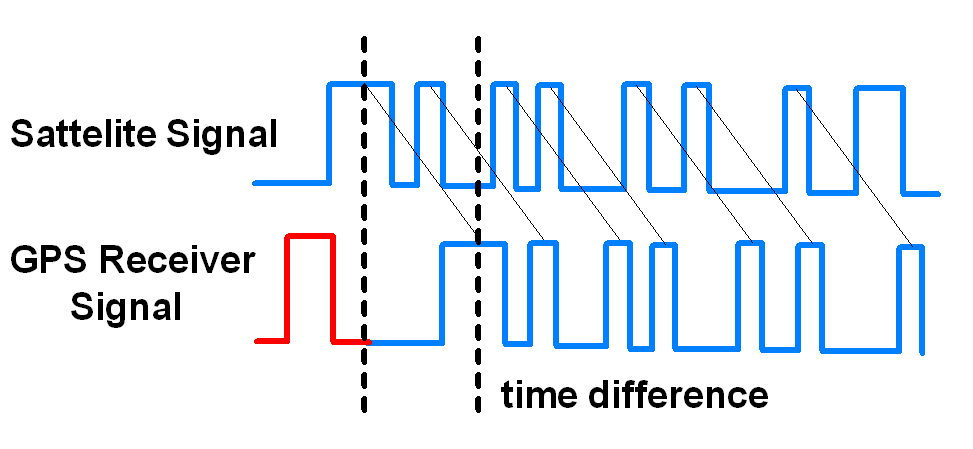
The GPS receiver captures and deciphers the information signals transmitted by GPS satellites, leveraging them to calculate the distance separating the receiver from these satellites. This calculation is accomplished by measuring the time it takes for the signal to travel from the satellite to the receiver.



**Fig. 6.14: GPS Distance Calculation**

 The equation "Distance = Speed x Time" can be understood as follows:

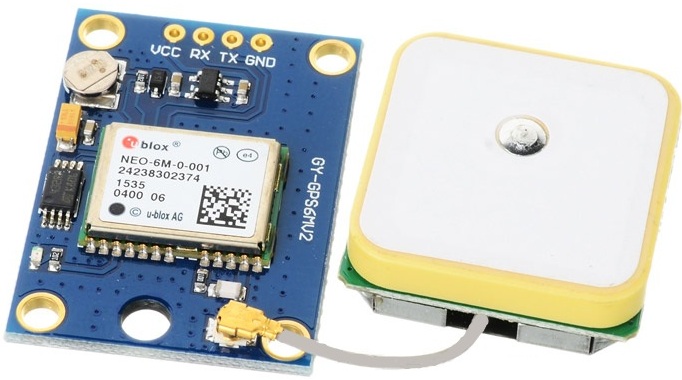
* Speed refers to the velocity of the radio signal, which is approximately equal to the speed of light.
* Time represents the period taken for the signal to travel from the satellite to the receiver. By calculating the difference between the transmitted time and the received time, we can determine the duration of the signal's journey.



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

To measure distance, both the satellite and GPS receiver generate matching pseudocode signals at the same time. The satellite transmits the pseudocode signal, which the GPS receiver picks up. Comparing these signals reveals the time taken for travel. When the receiver knows the distance from at least 3 satellites and their positions (sent by the satellites), it can pinpoint its own location using the Trilateration method.

# **GPS Module:-**



**Fig. 6.16: GPS Receiver**

The GPS receiver module generates output in the NMEA (National Marine Electronics Association) string format. This output is transmitted serially via the Tx pin, using a default baud rate of 9600. The NMEA string produced by the GPS receiver comprises different parameters, like longitude, latitude, altitude, and time, which are separated by commas. Each string begins with a '$' sign and ends with a carriage return/line feed sequence.

For instance:

$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

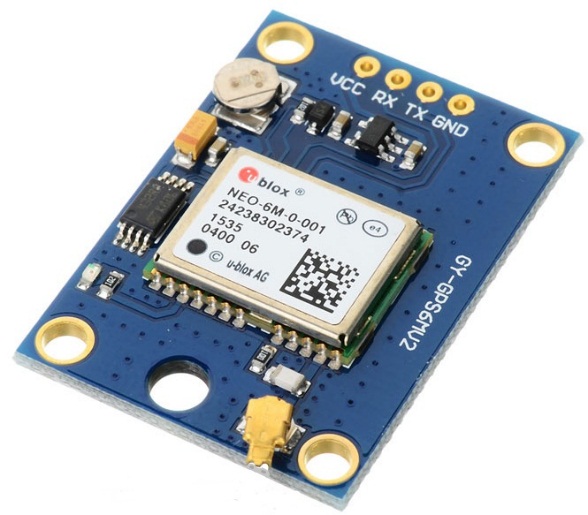
$GPGSV,3,1,11,15,47,133,46,25,44,226,45,18,37,238,45,26,34,087,40\*72

$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 Source 3.3 – 6 V

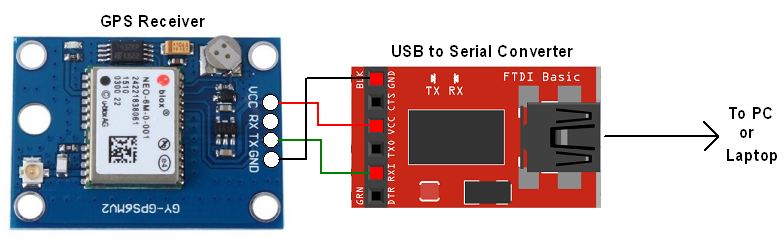
GND: Ground

TX: Transmits data serially, conveying details about location, time, etc.

RX: Receives data serially. Required for configuring the GPS module.

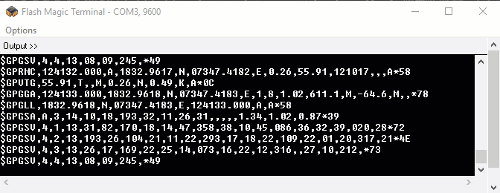
# ***6.8.2 Check GPS module:-***

Prior to connecting the GPS module to the PIC18F4550 microcontroller, it's prudent to verify the module's output. Extracting information like longitude, latitude, and time from the output string can assist in determining location and timing. This can be achieved by opening a serial terminal (e.g., Realterm, Hyperterminal, Putty) on a PC or laptop, configuring it to the designated port at a 9600 baud rate, and observing the data streaming from the GPS receiver module.



**Fig.6.18: GPS Serial Interface**

The serial terminal displays the output data from the GPS receiver module, as depicted.



Among the given strings, the "$GPGGA" NMEA string is frequently utilized. This string offers Time, Longitude, Latitude, Altitude, and directional details, serving as valuable information for establishing both time and location.

1. **SOFTWARE REQUIREMENTS**

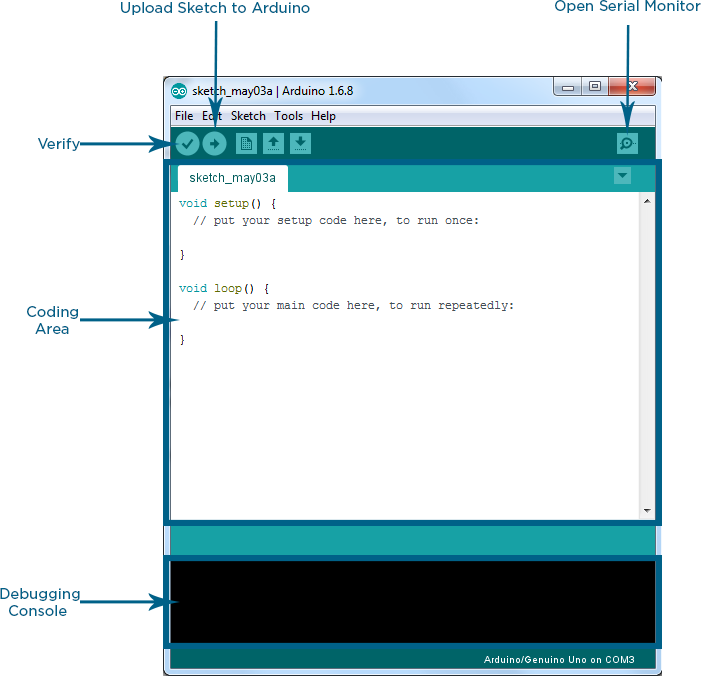
**7.1 Arduino IDE:-**

The Arduino IDE, short for Integrated Development Environment, is the official software developed by Arduino.cc. Its core function is to facilitate the writing, compiling, and uploading of code to Arduino devices. As open-source software, it is compatible with a variety of Arduino modules, providing an accessible way to learn programming. The IDE is available for Mac, Windows, and Linux operating systems, running on the Java Platform. It offers support for different Arduino boards such as Uno, Mega, Leonardo, and Micro. Each Arduino board houses a microcontroller that receives code (known as a sketch) created using the IDE. The IDE converts the sketch into a Hex File, which is then loaded onto the microcontroller of the board. The environment is composed of two main components: the Editor, where code is written, and the Compiler, which handles code compilation and uploading. Additionally, the environment supports both the C and C++ programming languages.

7.1.2 How to Install Arduino IDE:

You can download the software from the official Arduino website. Make sure to choose the version compatible with your operating system (Linux, Windows, or macOS). Note:

* The Windows app version requires Windows 8.1 or 10; it's not compatible with Windows 7 or older.
* The Arduino IDE consists of three sections: Menu Bar, Text Editor, and Output Pane.
* When you open the IDE, it will display a layout like the image below.



**Fig.7.19: Arduino IDE**

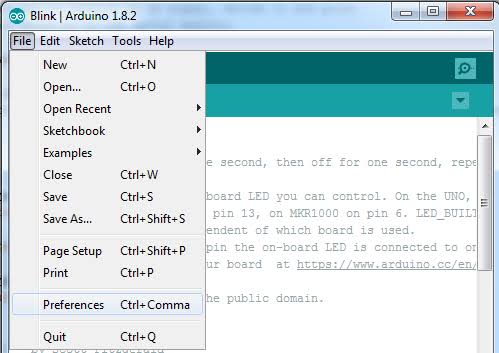
***Menu Bar in Arduino IDE:***

The top bar is known as the Menu Bar, offering five distinct options:

* File: This option allows you to create a new code window or open an existing one. It includes further subdivisions as shown in the table below.

File Menu Subdivisions:

|  |  |
| --- | --- |
| **Subdivision** | **Description** |
| New | Create a new sketch or project. |
| Open | Open an existing sketch or project. |
| Open Recent | Access recently opened sketches or projects. |
| Examples | Browse and load example sketches to learn from. |
| Save | Save the current sketch or project. |
| Save As | Save the current sketch or project with a new name. |
| Save Copy As | Save a copy of the current sketch or project. |
| Preferences | Configure settings for the Arduino IDE. |
| Page Setup | Adjust page setup settings for printing. |
| Print | Print the current sketch or project. |
| Exit | Close the Arduino IDE. |

****

When you navigate to the "Preferences" section within the Arduino IDE's Menu Bar and access the "Compilation" settings, any code compilation activity triggered by clicking the "Upload" button will be displayed in the Output Pane. This area provides real-time feedback on the compilation process, including information about any errors, warnings, or successful compilation messages that occur during the upload process. The Output Pane is a valuable tool for developers to identify and address any issues in their code before uploading it to the Arduino board.

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);

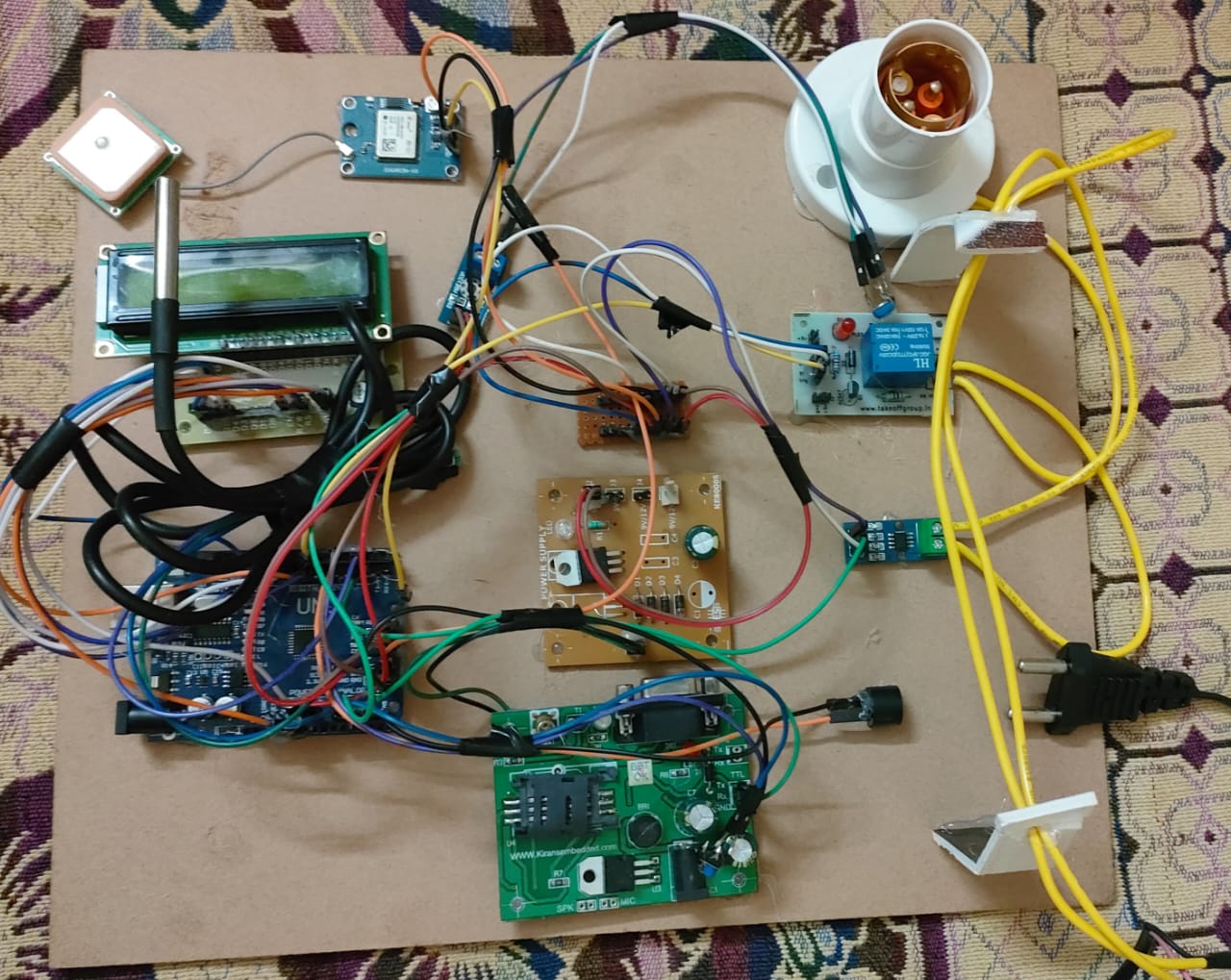
Serial.println("DONE..!!");

delay(3000);

}

1. **WORKING PRINCIPLE**

The project integrates an Arduino microcontroller with a range of sensors and components to develop an advanced monitoring and notification system. This system is designed to accurately identify cable faults and measure distances. Using a set of resistors symbolizing cable lengths in kilometers, along with strategically positioned switches for simulating faults, the system ensures accurate distance calculations. The voltage sensor captures the voltage drop across the feeder resistor, and this data is precisely digitized by an Analog-to-Digital Converter (ADC). The microcontroller then processes this data, calculating distances based on the reference resistors, and displays the results on a 16x2 LCD screen. The system also incorporates fault detection, recognizing simulated faults triggered by the switches and indicating the fault location and corresponding phase on the LCD. To enhance safety, the project constantly monitors temperature and activates an alarm through a buzzer and LCD notification if the temperature surpasses a predetermined threshold. Additionally, the integration of a Real-Time Clock (RTC) ensures accurate timestamping of events. The recorded data is sent to the internet via Internet of Things (IoT) technology, allowing remote access and monitoring. Overall, this project offers a comprehensive solution for cable monitoring, combining accurate measurement, fault detection, temperature safety, event timestamping, and remote access.



**Fig.9.20: Circuit Diagram**

****

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

In the event of a fault occurring in the distribution line, such as a pin insulator failure or conductor breakage, the Current Transformer (CT) plays a crucial role. The CT senses the interruption or break in the line, promptly generating a signal. This signal is then directed to an Analog to Digital Converter (ADC), which transforms the analog signal into digital format. Subsequently, the signal is transmitted to the Raspberry Pi Microprocessor.

The Raspberry Pi processor swiftly processes the received signal and initiates its transmission through the cloud. The intended recipients of this signal are the operators managing the control room and the linemen responsible for the specific area affected. These notifications are conveyed directly to their respective mobile phones. The information shared includes the precise location of the detected fault.

Equipped with the accurate fault location, the linemen can promptly respond and take necessary corrective actions. This streamlined process eliminates the time-consuming task of manually identifying the fault location, ensuring swift rectification. Ultimately, this proactive approach guarantees uninterrupted power supply to consumers, while also enhancing the efficiency of fault management within the distribution network.

**Advantages:**

* Wireless Communication Enablement: The utilization of wireless communication empowers seamless connectivity among devices, eliminating the need for physical wiring and enhancing flexibility in deployment.
* Expanded Coverage Area: This system offers a substantially larger coverage area in comparison to existing alternatives, ensuring a wider reach for monitoring and management.
* Simplicity and Efficiency: With fewer components required and reduced reliance on manual observation, the system is not only operationally efficient but also economically reliable, leading to cost savings.

Applications:

* Transmission Line Monitoring: The system finds valuable application in the monitoring of transmission lines, where it aids in fault detection, prevention, and seamless maintenance.
* Textile Mills: The technology proves beneficial in textile mills by facilitating remote monitoring of machinery and processes, optimizing operations and productivity.
* Food Industry: In the food industry, this system can enhance quality control, ensuring the optimal functioning of equipment and processes critical to food production and safety.

1. **CONCLUSION**

The model is meticulously designed with the primary aim of addressing consumer-related issues. Through this innovative approach, the detection and resolution of faults become considerably streamlined. This model exhibits a remarkable level of reliability, enabling the accurate identification of faults in three-phase transmission lines. Moreover, it also incorporates a robust data storage capability.

One of the standout features of this model is its real-time functionality. By operating in real time, it ensures the continuous monitoring and recording of relevant data. This proactive approach to data collection empowers us to maintain comprehensive data sheets that not only aid in the prompt detection of faults but also contribute to the prevention of potential future problems in the transmission line. In essence, the model serves as a comprehensive solution that not only resolves current issues efficiently but also lays the foundation for enhanced transmission line management in the long term.

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.