

Transformer Protection Using Microcontroller Based Relay and Monitoring Using IoT

A report submitted to the Department of Electrical and Electronic Engineering, Sonargaon University of Bangladesh in partial fulfillment of the requirements for the Award of Degree of Bachelor of Science in Electrical and Electronic Engineering.

Submitted by

Md. Tanvir Khan Rono	ID: EEE1803015015
Md. Saiful Islam	ID: EEE1803015055
Md. Shoibal Hasan	ID: EEE1803015009
Md. Obaydullah Khan	ID: EEE1803015001
Md. Masud Kabir	ID: EEE1803015032



Supervised by

Md. Rias Uddin Mollah
Lecturer
Department of Electrical and Electronic Engineering
Sonargaon University (SU)
Dhaka-1215, Bangladesh

22 January, 2022

Letter of Transmittal

22 January, 2022

To

Md. Rias Uddin Mollah

Lecturer

Department of Electrical and Electronic Engineering.

Sonargaon University of Bangladesh

Subject: Submission of Project Report.

Dear Sir,

We are pleased to submit the project report on **“Transformer Protection Using Microcontroller Based Relay and Monitoring Using IoT ”**. It was a great pleasure to work on such an important topic. This project has been done as per instruction of your supervision and according to the requirements of the Sonargaon University.

We expect that the project will be accepted by the concerned authority we will remain happy to further explanation that you may feel necessary in this regard.

Thank You

Sincerely yours,

Md. Tanvir Khan Rono
EEE1803015015; Group: 3

Md. Saiful Islam
EEE1803015055; Group: 3

Md. Shoibal Hasan
EEE1803015009; Group: 3

Md. Obydullah Khan
EEE1803015001; Group: 3

Md. Masud Kabir
EEE1803015032; Group: 3

DECLARATION

We do hereby solemnly declare that, the work presented here in this project report has been carried out by us and has not been previously submitted to any University/ Organization for award of any degree or certificate

We hereby ensure that the works that has been prevented here does not breach any existing copyright.

We further undertake to indemnify the university against any loss or damage arising from breach of the foregoing obligation.

Md. Rias Uddin Mollah

Lecturer

Department of Electrical and Electronic Engineering

Sonargaon University (SU)

ACKNOWLEDGEMENT

First we started in the name of almighty Allah. This thesis is accomplished under the supervision of **Md. Rias Uddin Mollah**, Lecturer, Department of EEE, Sonargaon University. It is a great pleasure to acknowledge our profound gratitude and respect to our supervisor for this consistent guidance, encouragement, helpful suggestion, constructive criticism and endless patience through the progress of this work. The successful completion of this thesis would not have been possible without his persistent motivation and continuous guidance.

The authors are also grateful **Prof. Dr. M. Bashir Uddin**, Head of the Department of Electrical and Electronic Engineering and all respect teachers of the Electrical and Electronic Engineering Department for their co-operation and significant help for completing the thesis work successfully.

[Authors]

Md. Tanvir Khan Rono
EEE1803015015; Group: 3

Md. Saiful Islam
EEE1803015055; Group: 3

Md. Shoibal Hasan
EEE1803015009; Group: 3

Md. Obydullah Khan
EEE1803015001; Group: 3

Md. Masud Kabir
EEE1803015032; Group: 3

Abstract

The aim of the project is to design a Transformer Protection Using Microcontroller Based Relay and Monitoring Using IoT. The present monitoring system requires manpower which is time consuming and it is difficult to predict the occurrence of faults. In this project, we designed a system which Building of an automatic system using the Arduino microcontroller for the monitoring and protection of the transformer and also Making an android app fit for our purpose and synchronization of the system with the Node MCU (Wi-Fi module) for getting notifications and real-time monitoring of the parameters of the transformer. The system would also provide the real-time monitoring of the parameters of the transformer. This paper deals with real time monitoring of voltage levels, oil levels, and temperature of transformer and gas sensors. Minimum and maximum values are set for each parameter mentioned. These devices will continuously compare both set values with values at that particular instant. On the android device and get notifications in the cases of faults.

Table of Contents

Certificate	II
Declaration	III
Acknowledgement	V
Abstract	VI
List of Figures	VIII
List of Tables	IX

Chapter 1 Introduction

1.1	Introduction	01
1.2	Problem Statement	02
1.3	Project Keywords	02
1.4	Objectives	02
1.5	Methodology	02-03
1.6	Organization of the Book	03

Chapter 2 Literature Review

2.1	Literature Review	04-05
-----	-------------------	-------

Chapter 3 Equipment and Software Details

3.1	Arduino Nano	06-09
3.2	LCD Display	09-10
3.3	Node MCU	10-13
3.4	Current Snsor	13-14
3.5	Voltage Sensor	14-15
3.6	SMPS	15-16
3.7	Ultrasonic Sensor	16-19
3.8	MQ2 Sensor	19-20
3.9	Relay	21-23
3.10	Temperature Sensor	23-24
3.11	5V Regulator IC	24-25

3.12	Arduino IDE	25-34
3.13	Proteus Software	34-35
3.14	Remote XY	35-37

Chapter 4 System Construction

4.1	System Description	38
4.2	Block Diagram	39
4.3	Circuit Diagram	39
4.4	Working Principle	40
4.5	The Project Prototype	40
4.6	Results	41
4.7	Cost Analysis	41

Chapter 5 Conclusion

5.1	Advantages	42
5.2	Applications	42
5.3	Future Scope of Work	42
5.4	Conclusion	43

Reference	44-45
------------------	--------------

Appendix	46
-----------------	-----------

List of Figures

Figure 3.1	Arduino Nano	06
Figure 3.2	Arduino Nano Schematic Diagram	07
Figure 3.3	How Arduino Nano Looks Like	07
Figure 3.4	Microcontroller IC AT mega 328p	08
Figure 3.5	LCD Display	10
Figure 3.6	Node MCU	10
Figure 3.7	Node MCU Schematic Diagram	11
Figure 3.8	Node MCU Pin Out	12
Figure 3.9	AC Current Sensor	14
Figure 3.10	Voltage Sensor	15
Figure 3.11	SMPS Power Supply	15
Figure 3.12	Working of Sensor	17
Figure 3.13	Smoke Sensor	20
Figure 3.14	Relay	21
Figure 3.15	Relay Module	22
Figure 3.16	Main Voltage Connection	22
Figure 3.17	Voltage Connection	23
Figure 3.18	DHT 11 Temperature Sensor	24
Figure 3.19	5V Regulator IC	25
Figure 3.20	Arduino Software Interface IDE	26
Figure 3.21	Proteus Software Interface	35
Figure 4.1	Our Project in the Proteus Software	38
Figure 4.2	Block Diagram	39
Figure 4.3	Schematic Diagram	39
Figure 4.5	Project Prototype Image	40

List of Tables

Table 01	Ultrasonic Sensor Pin Configuration	18
Table 02	Cost of Componets with price	41

Chapter 1

Introduction

1.1 Introduction

Transformer is the most important device in the power system. Thus, it has to be continuously monitored and kept healthy. Any fault or unbalancing in the transformer may cause the serious problems in the power system. Thus, here our objective is to build a system that would continuously monitor the transformer and do the needful in the cases of unbalancing. The system would also provide the real-time monitoring of the parameters of the transformer like voltage, current, temperature etc. on the android device and personal computers and get notifications in the cases of faults. Our project involves two phases. (1) Building of an automatic system using the Arduino microcontroller for the monitoring and protection of the transformer. (2) Making an android app fit for our purpose and synchronization of the system with the Nodemcu (Wi-Fi module) for getting notifications and real-time monitoring of the parameters of the transformer. The aim of this research paper is to make system easy and convenient. Automatic systems are used instead of traditional methods. Monitoring is vital for sustenance of mankind but simultaneously it is very tedious and time consuming. By introducing advanced systems, this process can become better. There is already a shortage of V&I sources and hence it becomes essential for mankind to save it. This proposed system consumes the least amount time required for monitoring, thus saving it. This paper proposes a system which does controlling the process with the help of a IoT on mobile application, which will make easy to monitor the transformer from remote location and even decrease the requirement of manpower to look after the system as our system is automatic.

Reliability and safety issues of power system have been more important with progress and development of national economy and power system [1]. If we look back towards our daily routine, we can conclude that electricity is the inseparable part of our life and transformers plays a role of electricity carrier to us from generation stations. Transformer is the key component in electricity distribution system. Hence protection of transformer is very important. Transformers are used in many applications i.e. from small projects to the mega industries [5]. Increasing population and machinery is resulting in more and more demand of power [4]. Transformers get overloaded due to illegal use of electricity. Overload affects the

efficiency of transformer and electricity distribution system. So, the designed system involves automatic isolation of load to avoid damage to the transformer due to overloading. Therefore, a proposed method is chosen to design microcontroller-based transformer for overload protection. The microcontroller-based relay provides more adjustable characteristics, high accuracy, more flexibility, increased range of setting, and reduced size, minimum cost with many functions such as self-monitoring and checking by GSM technology

1.2 Problem Statement

Modern power system requires accurate, reliable technique for detection of faults, real time data monitoring and fast response speed. The reliable operation of the power system depends upon the effective functioning of the distribution transformer. Microcontroller based system has real time data monitoring, detection of abnormal condition, fast processing speed, reduced installation cost low maintenance cost and more flexibility.

1.3 Project Keywords

Transformer's protection, Relay, Internet of things, Voltage control, microcontroller Arduino, Real time monitoring.

1.4 Objectives

The objectives of this project are:

- To study about **Transformer Protection Using Microcontroller Based Relay and Monitoring Using IoT.**
- To design and construct a **Transformer Protection Using Microcontroller Based Relay and Monitoring Using IoT.**
- To test the performance of the transformer health monitoring through IoT.

1.5 Methodology

Our methodologies for the project:

- Creating an idea for Design and construction of **Transformer Protection Using Microcontroller Based Relay and Monitoring Using IoT** And drawing and listed of components/materials to know which components / materials need to construct it.

- Collecting the all components / materials construct the system.
- Finally, we constructed this system & checked it finally that working very well.

1.6 Organization of the Book

This project book consists of five chapter. The first chapter contains the statement of the introduction, our motivation for the project, objectives of the study, our used methodologies and the project organization. Chapter two contains literature review in details. Chapter three describes the proposed system architecture with component details and the software which we have used for our work. Chapter four deals with the circuit design, working principle and shows the complete prototype of the project that we have built. In the final chapter, we have discussed the advantages and applications of the project and also about some aspects we had to overcome while doing the project and lastly we gave the conclusion of the book.

Chapter 2

Literature Review

2.1 Literature Review

Literature review plays an important role in deciding the objectives of the study. It helps in the successful completion of the work to arrive at the desired results. Many improvements in communication and hardware technology have been developed - a promising one is the IoT technology, from home automation to industrial IoT. The IoT is an emerging technology, and it has been a great interesting topic for the past few years. Much literature is available on monitoring, controlling, and protecting the transformer, and enormous work has been done in this area.

A system based on the microcontroller to monitor and save the substation transformer from current rise due to overload is proposed [1]. A PLC-based automatic control system [2, 3] to monitor and detect the transformer's internal faults and external faults were proposed. In [4], a protective system using a temperature sensor, microcontroller, LCD, GSM, and Xbee was proposed to send the message to the electricity board. A conventional fault detecting method [5] have been employed for transformer protection. Nowadays, transformer fault monitoring based on vibration analysis drawing the attention of researchers because of added advantages. This method shows satisfactory results, but it can still be improved using computer algorithms to analyze the data and predict the fault. This work presents, development of an IoT based system for real-time monitoring and control of transformer parameters. This system is placed close to the transformer, and considered parameters are diagnosed and are transmitted to a centralized web server. Thus, the data is utilized to know the transformer's condition on a real-time basis and are stored within a server database for future analysis.

Previously many development were seen to design relay's to overcome limits such development were made to increase number of analog inputs, CT saturation countermeasure and high speed operation of differential relay [2]. Survey allowed us to study various problems with previous methods. CT saturation must be considered for differential protection. If the saturation of CT occurs during external fault, current does not flow during

the non saturation period. Hence digitized relay were used for protection. On the distribution side of the line to solve the specific protection and control problems, digital relays have been used. Digital relays have advantages over analog relays previously used. Now mostly digital multifunction relays are used as they provide reliability, versatility and have low cost [3]. Differential relays protection can sometimes malfunction. The reason of this malfunction is considered to be due to inrush current in transformer at the time of starting. Several methods like per-phase method, cross blocking method, Percent average blocking method and Harmonic sharing method were used. To overcome such malfunction and to protect the transformer, protection using fuzzy methods were introduced which was more sophisticated than the previous methods [4]. Internet of things is now used almost everywhere and industries are booming due to IOT.

Chapter 3

Equipment and Software Details

3.1 Arduino Nano

Arduino is an open source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling Lights, motors, and other actuators.



Figure 3.1: Arduino Nano

The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, Maxims'). Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a small, complete, and breadboard friendly component. It has everything that Decimal/ Duemilanove has (electrically) with more analog input pins and onboard +5V AREF jumper. Physically, it is missing power jack. The Nano can automatically sense and switch to the higher potential source of power.

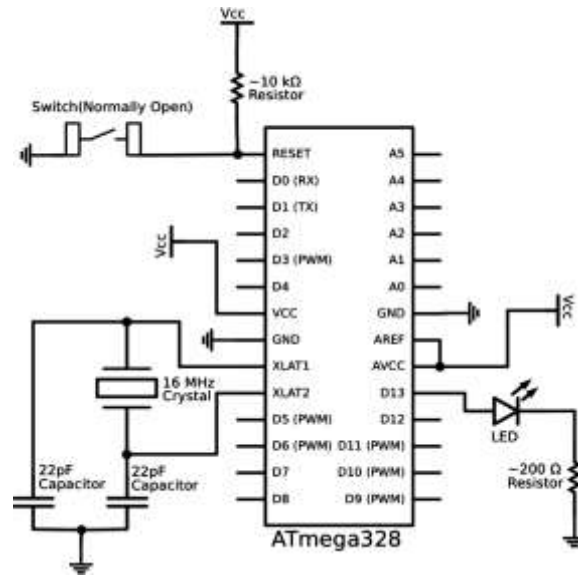


Figure 3.2: Arduino Nano Schematic Diagram

Nano's got the breadboard-ability of the Boarding and the Minibus with smaller footprint than either, so users have more breadboard space. It's got a pin layout that works well with the Mini or the Basic Stamp (TX, RX, ATN, and GND on one top, power and ground on the other). This new version 3.0 comes with ATMEGA328 which offer more programming and data memory space. It has two layers. That make it easier to hack and more affordable. One of the best features of Arduino Nano is, it's easy to use, compact and also small.

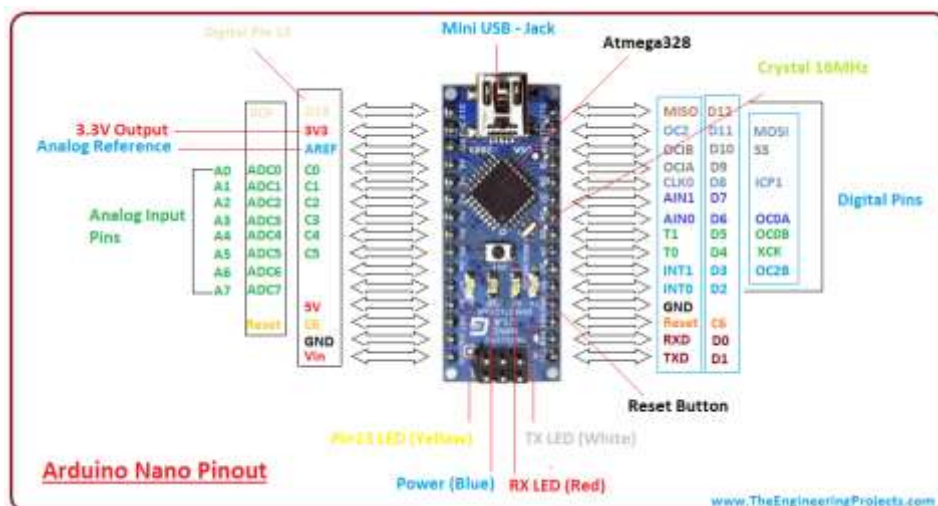


Figure 3.3: How Arduino Nano looks like

Specifications:

- Microcontroller: Atmel ATmega328
- Operating Voltage (logic level): 5 V
- Input Voltage (recommended): 7-12 V
- Input Voltage (limits): 6-20 V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 8
- DC Current per I/O Pin: 40 mA
- Flash Memory: 32 KB (of which 2KB used by boot loader)
- SRAM : 2 KB

Features:

- Automatic reset during program download
- Power OK blue LED
- Green (TX), red (RX) and orange (L) LED
- Auto sensing/switching power input
- Small mini-B USB for programming and serial monitor
- ICSP header for direct program download
- Standard 0.1 spacing DIP (breadboard friendly)
- Manual reset switch

Microcontroller IC ATmega328p



Figure 3.4: Microcontroller IC AT mega 328p

The high-performance Microchip Pico Power 8-bit AVR RISC-based microcontroller combines 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

3.2 LCD Display

LCD (Liquid Crystal Display) screen is an electronics display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits.

These modules are preferred over seven segments and other multi segment LEDs. The reasons being LCDs are economical; easily programmable, have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc.

Features of LCD Display:

Built-in controller (KS 0066 or Equivalent) + 5V power supply (Also available for + 3V)
1/16 duty cycle B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED) N.V. optional for + 3V power supply.

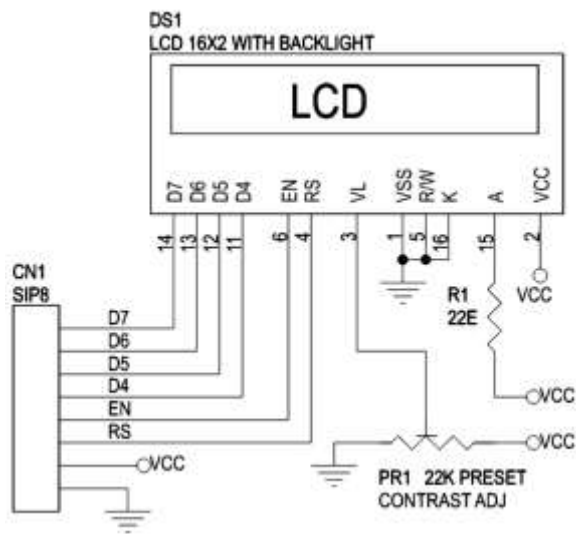


Figure 3.5: 16*2 LCD Display

3.3 Node MCU

Node MCU is an open source firmware for which open source prototyping board designs are available. The name "Node MCU" combines "node" and "MCU" (micro-controller unit). The term "Node MCU" strictly speaking refers to the firmware rather than the associated development kits. Both the firmware and prototyping board designs are open source.

The firmware uses the Lua scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.



Figure 3.6: Node MCU

The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications.

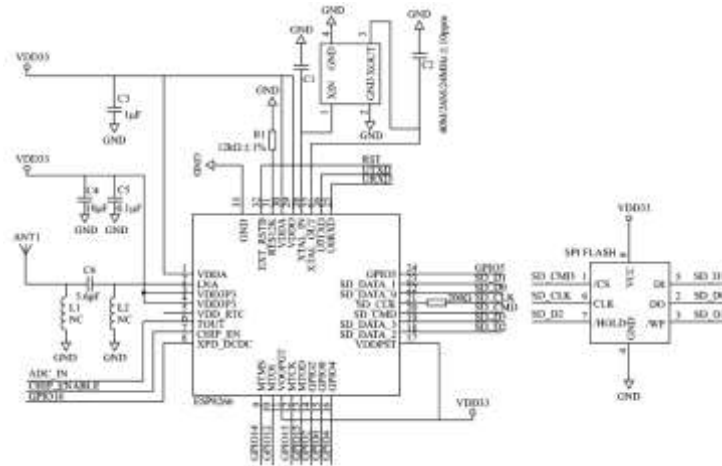


Figure 3.7: Node MCU Schematic Diagram

This is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "Node MCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications (see related projects).

NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9. Later that month, Tuan PM ported MQTT client library from Contiki to the ESP8266 SoC platform, and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib to NodeMCU project, enabling

NodeMCU to easily drive LCD, Screen, OLED, even VGA displays. In summer 2015 the creators abandoned the firmware project and a group of independent contributors took over. By summer 2016 the NodeMCU included more than 40 different modules. Due to resource constraints users need to select the modules relevant for their project and build a firmware tailored to their needs.

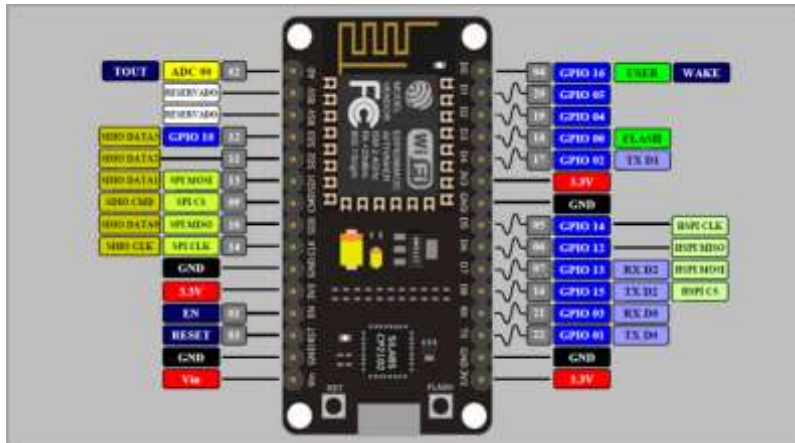


Figure 3.8: Node MCU Pin Out

Node MCU V3 ESP8266 ESP-12E is WiFi development board that helps you to prototype your IoT product with few Lua script lines, or through Arduino IDE. The board is based on ESP8266 ESP-12E variant, unlike other ESP-12E, you won't need to buy a separate breakout board, usb to serial adapter, or even solder it to a PCB to get started, you will only need a usb cable (Micro USB).

Features

1. Communication interface voltage: 3.3V.
2. Antenna type: Built-in PCB antenna is available.
3. Wireless 802.11 b/g/n standard
4. WiFi at 2.4GHz, support WPA / WPA2 security mode
5. Support STA/AP/STA + AP three operating modes
6. Built-in TCP/IP protocol stack to support multiple TCP Client connections (5 MAX)
7. D0 ~ D8, SD1 ~ SD3: used as GPIO, PWM, IIC, etc., port driver capability 15mA
8. AD0: 1 channel ADC
9. Power input: 4.5V ~ 9V (10VMAX), USB-powered
10. Current: continuous transmission: $\approx 70\text{mA}$ (200mA MAX), Standby: $< 200\mu\text{A}$

11. Transfer rate: 110-460800bps
12. Support UART / GPIO data communication interface
13. Remote firmware upgrade (OTA)
14. Flash size: 4MByte.

3.4 Current Sensor

The module is equipped with the ZMCT103C series of small high-precision current transformers and high-precision op amp circuits for accurate sampling and proper compensation of signals. It is convenient for signal acquisition of AC power within 5A. The corresponding output analog AC signal can be adjusted. The required output voltage can be adjusted according to the potentiometer (adjusting the amplification ratio, the amplification range is 0-100 times), but the maximum voltage at the output (OUT) will not exceed $1/2 VCC$

Product Description:

- On-board micro-precision current transformer
- Onboard sampling resistor
- Modules 5A can be measured within an alternating current, the analog output corresponding to 5A/5mA
- Rated input current: 5A
- Rated output current: 5mA
- Change: 1000: 1
- The linear range: 0 ~ 10A (100 ohm)
- Precision rating: 0.2
- Uses isolation voltage: 3000V Measurement
- Sealing material: epoxy resin
- Operating temperature: - 40 Celsius - + 70 Celsius



Figure 3.9: AC Current Sensor

3.5 Voltage Sensor

ZMPT101B AC Single Phase voltage sensor module is based on a high precision ZMPT101B voltage Transformer. ZMPT101B AC Voltage Sensor is the best for the purpose of the DIY project, where we need to measure the accurate AC voltage with a voltage transformer. This is an ideal choice to measure the AC voltage using Arduino /ESP8266 /Raspberry Pi like an open source platform. In many electrical projects, engineer directly deals with measurements with few basic requirements like High galvanic isolation, Wide Range, High accuracy, Good Consistency.

Onboard precision miniature voltage transformer, The active phase AC output voltage transformer module. Onboard precision op-amp circuit, the signal sampling and appropriate compensation for precise functions. Modules can be measured within 250V AC voltage, the corresponding analog output can be adjusted. It is brand new, good quality high performance.

FEATURES of ZMPT101B AC Single Phase Voltage Sensor Module:-

- Voltage upto 250 volts can be measured
- Light weight with on-board micro-precision voltage transformer
- High precision on-board op-amp circuit
- Operating temperature : 40°C ~ + 70°C
- Supply voltage 5 volts to 30 volts

ADVANTAGES of ZMPT101B AC Single Phase Voltage Sensor Module:-

- Analog output corresponding quantity can be adjusted.
- Pcb board size : 49.5 (mm) x19.4 (mm)
- Good consistency, for voltage and power measurement
- Very efficient and accuracy



Figure 3.10: Voltage Sensor

3.6 SMPS

A switched-mode power supply is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently. Like other power supplies, an SMPS transfers power from a DC or AC source to DC loads, such as a personal computer, while converting voltage and current characteristics.



Figure 3.11: SMPS Power Supply

A switched-mode power supply (switching-mode power supply, switch-mode power supply, switched power supply, SMPS, or switcher) is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently.

Like other power supplies, an SMPS transfers power from a DC or AC source (often mains power) to DC loads, such as a personal computer, while converting voltage and current characteristics. Unlike a linear power supply, the pass transistor of a switching-mode supply continually switches between low-dissipation, full-on and full-off states, and spends very little time in the high dissipation transitions, which minimizes wasted energy. Ideally, a switched-mode power supply dissipates no power. Voltage regulation is achieved by varying the ratio of on-to-off time. In contrast, a linear power supply regulates the output voltage by continually dissipating power in the pass transistor. This higher power conversion efficiency is an important advantage of a switched-mode power supply. Switched-mode power supplies may also be substantially smaller and lighter than a linear supply due to the smaller transformer size and weight.

Switching regulators are used as replacements for linear regulators when higher efficiency, smaller size or lighter weight are required. They are, however, more complicated; their switching currents can cause electrical noise problems if not carefully suppressed, and simple designs may have a poor power factor.

3.7 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

HC-SR04 Ultrasonic Sensor - Working

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

$$\mathbf{Distance = Speed \times Time}$$

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below-

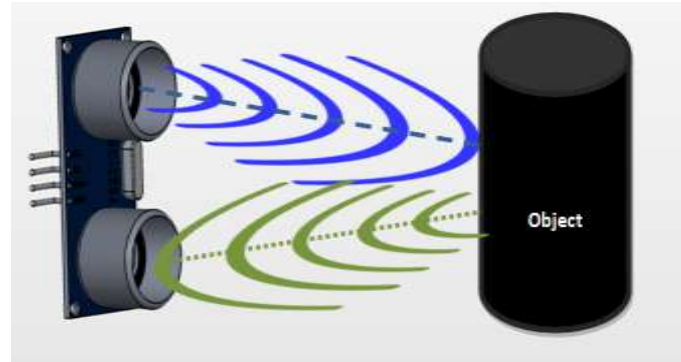


Figure 3.12: Working of sensor

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

How to use the HC-SR04 Ultrasonic Sensor

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie etc. The following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the Vcc and Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return back to the Sensor. Using this information, the distance is measured as explained in the above heading.

Applications

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

Ultrasonic Sensor Pin Configuration

Table 01: Ultrasonic Sensor Pin Configuration

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <math><15^\circ</math>

- Operating Current: <15mA
- Operating Frequency: 40Hz

3.8 MQ2 Smoke Detector Module

The utility model can be used for gas leakage monitoring devices in families and factories, and is suitable for the detection of liquefied petroleum gas, butane, propane, methane, Hydrogen, smoke, etc. This is a very easy to use low-cost semiconductor Gas sensor Module with analog and digital output.

APPLICATIONS:

- Safety of home.
- Safety of office and commercial space.
- Safety of Hospital and school.

Working Principle:

The MQ2 has an electrochemical sensor, which changes its resistance for different concentrations of varied gasses. The sensor is connected in series with a variable resistor to form a voltage divider circuit (Fig 1), and the variable resistor is used to change sensitivity. When one of the above gaseous elements comes in contact with the sensor after heating, the sensor's resistance change. The change in the resistance changes the voltage across the sensor, and this voltage can be read by a microcontroller. The voltage value can be used to find the resistance of the sensor by knowing the reference voltage and the other resistor's resistance. The sensor has different sensitivity for different types of gasses.

Features:

- Adopt high quality double panel design, with power indication and TTL signal output indication.
- It has DO switch signal (TTL) output and AO analog signal output.
- TTL output valid signal is low level. When the output is low, the signal light is on, and the microcontroller or relay module can be directly connected.
- The analog output voltage increases with the concentration, the higher the voltage.
- It has better sensitivity to liquefied petroleum gas, natural gas, urban gas and smoke.

- MQ-2 MQ2 Smoke Gas LPG Butane Methane Sensor Detector Module
- With four screw holes, easy to locate.
- Product size: 32 (L), *20 (W), *22 (H)
- With long service life and reliable stability.
- Fast response recovery features

Specifications:

- Input voltage: DC5V
- Power dissipation (current): 150mA
- DO output: TTL, numeric quantities 0 and 1 (0.1 and 5V)
- AO output: 0.1-0.3V (relatively pollution-free), the highest concentration of about 4V voltage
- Special reminder: after the sensor is energized, you need to preheat 20S or so, the data to be stable, sensor heating is a normal phenomenon, because the internal heating wire, if hot, it is not normal.

Connection mode:

- VCC: power supply positive (5V)
- GND: power supply negative pole
- DO:TTL switch signal output
- AO: analog signal output
- Functions: This version supporting test procedures
- Using chips: AT89S52
- Crystal oscillator: 11.0592MHZ



Figure 3.13: Smoke Sensor

3.9 Relay:

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a separate low-power signal, or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

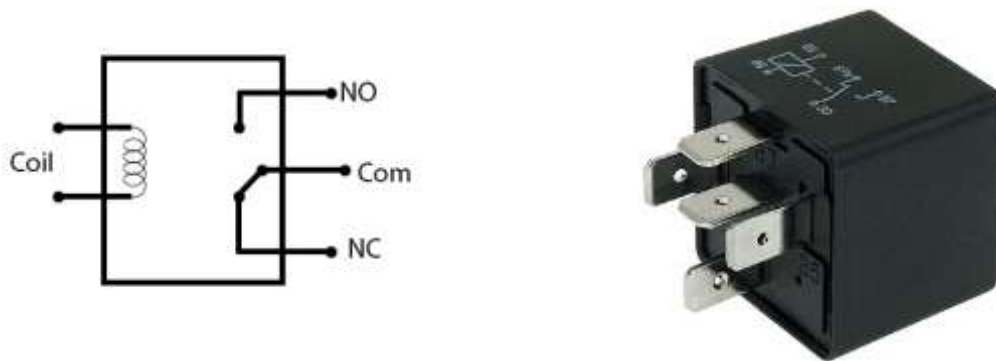


Figure 3.14: Relay.

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

Magnetic latching relays require one pulse of coil power to move their contacts in one direction, and another, redirected pulse to move them back. Repeated pulses from the same input have no effect. Magnetic latching relays are useful in applications where interrupted power should not be able to transition the contacts.

Magnetic latching relays can have either single or dual coils. On a single coil device, the relay will operate in one direction when power is applied with one polarity, and will reset when the polarity is reversed. On a dual coil device, when polarized voltage is applied to the

reset coil the contacts will transition. AC controlled magnetic latch relays have single coils that employ steering diodes to differentiate between operate and reset commands.

The relay module is the one in the figure below.



Figure 3.15: Relay Module.

This module has two channels (those blue cubes). There are other varieties with one, four and eight channels.

Mains voltage connections.

In relation to mains voltage, relays have 3 possible connections:



Figure. 3.16: Mains voltage connections.

- **COM:** common pin
- **NO (Normally Open):** there is no contact between the common pin and the normally open pin. So, when you trigger the relay, it connects to the COM pin and supply is provided to a load
- **NC (Normally Closed):** there is contact between the common pin and the normally closed pin. There is always connection between the COM and NC pins, even when the relay is turned off. When you trigger the relay, the circuit is opened and there is no supply provided to a load.

Pin wiring

The connections between the relay module and the Arduino are really simple:

- **GND:** goes to ground
- **IN1:** controls the first relay (it will be connected to an Arduino digital pin)
- **IN2:** controls the second relay (it should be connected to an Arduino digital pin if



Figure 3.17: Main's voltage connections.

3.10 Temperature Sensor

The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data.

DHT11 Specifications:

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: $\pm 1^\circ\text{C}$ and $\pm 1\%$

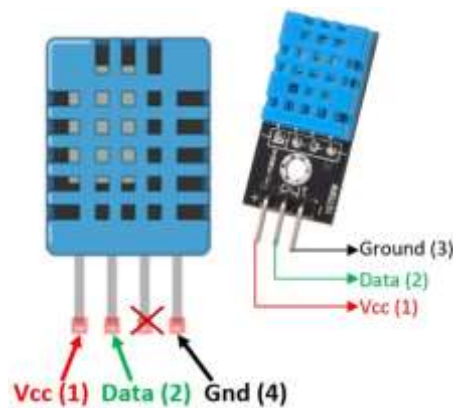


Figure 3.18: DHT11 Temperature Sensor

3.11 5V Regulator IC

Voltage sources in a circuit may have fluctuations resulting in not providing fixed voltage outputs. A voltage regulator IC maintains the output voltage at a constant value. 7805 IC, a member of 78xx series of fixed linear voltage regulators used to maintain such fluctuations, is a popular voltage regulator integrated circuit (IC). The xx in 78xx indicates the output voltage it provides. 7805 IC provides +5 volts regulated power supply with provisions to add a heat sink.

7805 IC Rating

- Input voltage range 7V- 35V
- Current rating $I_c = 1A$
- Output voltage range $V_{Max}=5.2V$, $V_{Min}=4.8V$

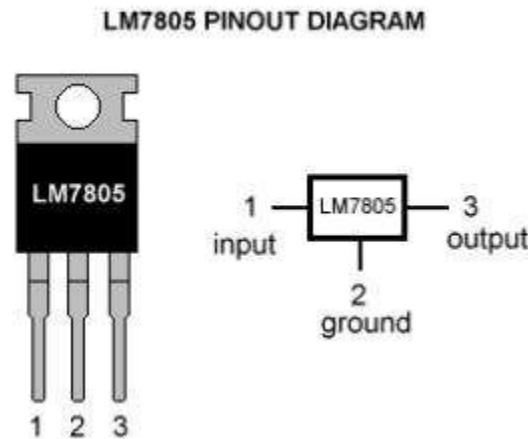


Figure 3.19: 5V Regulator IC

3.12 Arduino IDE

The digital microcontroller unit named as Arduino Nano can be programmed with the Arduino software IDE. There is no any requirement for installing other software rather than Arduino. Firstly, Select "Arduino Nano from the Tools, Board menu (according to the microcontroller on our board). The IC used named as ATmega328 on the Arduino Nano comes pre burned with a boot loader that allows us to upload new code to it without the use of an external hardware programmer.

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and programs the microcontroller through the ICSP (In Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

The Arduino Nano is one of the latest digital microcontroller units and has a number of

facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 -(RX) for receive the data and pin no.1 (TX) for transmit the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .in file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial Communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. Arduino programs are written in C or C++ and the program code written for Arduino is called sketch. The Arduino IDE uses the GNU tool chain and AVR Lab to compile programs, and for uploading the programs it uses argued. As the Arduino platform uses Atmel microcontrollers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino.

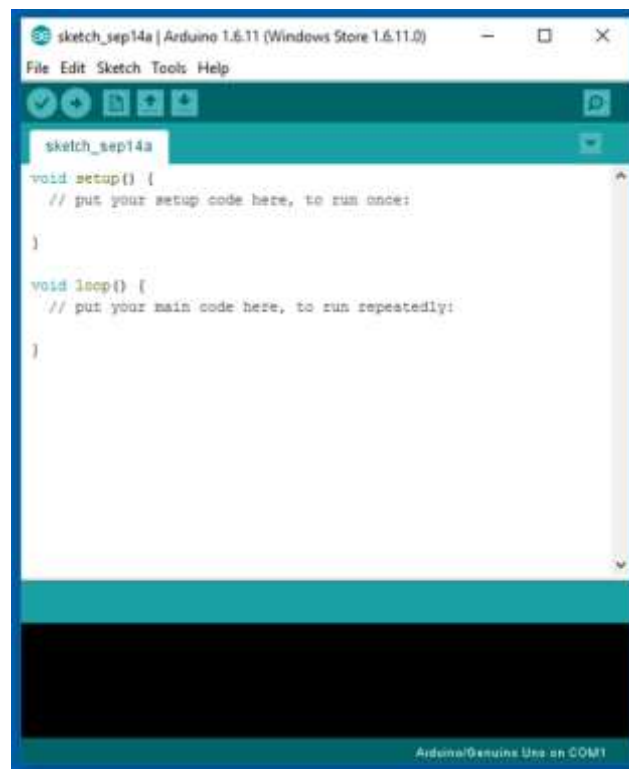


Figure 3.20: Arduino Software Interface IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

Verify

Checks your code for errors compiling it.

Upload

Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New

Creates a new sketch.

Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.

Save

Saves your sketch.

Serial Monitor

Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File**New**

Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.

Open

Allows to load a sketch file browsing through the computer drives and folders.

Open Recent

Provides a short list of the most recent sketches, ready to be opened.

Sketchbook

Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.

Examples

Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.

Close

Closes the instance of the Arduino Software from which it is clicked.

Save

Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as." window.

Save as...

Allows to save the current sketch with a different name.

Page Setup

It shows the Page Setup window for printing.

Print

Sends the current sketch to the printer according to the settings defined in Page Setup.

Preferences

Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.

Quit

Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

Edit

Undo/Redo

Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.

Cut

Removes the selected text from the editor and places it into the clipboard.

Copy

Duplicates the selected text in the editor and places it into the clipboard.

Copy for Forum

Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.

Copy as HTML

Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.

Paste

Puts the contents of the clipboard at the cursor position, in the editor.

Select All

Selects and highlights the whole content of the editor.

Comment/Uncomment

Puts or removes the // comment marker at the beginning of each selected line.

Increase/Decrease Indent

Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.

Find

Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.

Find Next

Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.

Find Previous

Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

Sketch**Verify/Compile**

Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.

Upload

Compiles and loads the binary file onto the configured board through the configured Port.

Upload Using Programmer

This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a Tools -> Burn Bootloader command must be executed.

Export Compiled Binary

Saves a .hex file that may be kept as archive or sent to the board using other tools.

Show Sketch Folder

Opens the current sketch folder.

Include Library

Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.

Add File...

Adds a source file to the sketch (it will be copied from its current location). The new file

appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side of the toolbar.

Tools

Auto Format

This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.

Archive Sketch

Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.

Fix Encoding & Reload

Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.

Serial Monitor

Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

Board

Select the board that you're using. See below for descriptions of the various boards.

Port

This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.

Programmer

For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.

Burn Bootloader

The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before

burning the bootloader on the target board. This command also set the right fuses.

Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

Find in Reference

This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog.

Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the

upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more `#include` statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its `#include` statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

Serial Monitor

This displays serial sent from the Arduino or Genuino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

You can also talk to the board from Processing, Flash, MaxMSP, etc (see the interfacing page for details).

3.13 Proteus Software

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronics design engineers and technicians to create schematics and electronics prints for manufacturing printed circuit boards.

The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for DOS in 1988. Schematic Capture support followed in 1990 with a port to the Windows environment shortly thereafter. Mixed mode SPICE Simulation was first integrated into Proteus in 1996 and microcontroller simulation then arrived in Proteus in 1998. Shape based auto routing was added in 2002 and 2006 saw another major product update with 3D Board Visualization. More recently, a dedicated IDE for simulation was added in 2011 and MCAD import/export was included in 2015. Support for high speed design was added in 2017. Feature led product releases are typically biannual, while maintenance based service packs are released as required.

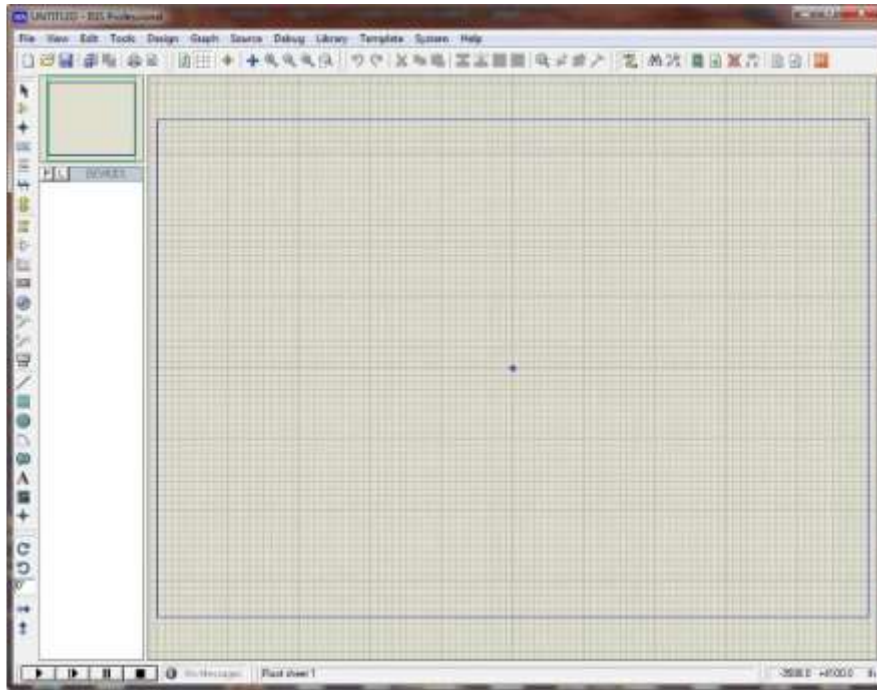


Figure 3.21: Proteus Software Interface

3.14 Remote XY:

Remote XY is easy way to make and use a mobile graphical user interface for controller boards to control via smartphone or tablet. The system includes:

- Editor of mobile graphical interfaces for controller boards, located on the site remotexy.com
- Mobile app Remote XY that allows to connect to the controller and control it via graphical interface.

Distinctive features:

- The interface structure is stored in the controller. When connected, there is no interaction with servers to download the interface. The interface structure is downloaded to the mobile application from the controller.
- One mobile application can manage all your devices. The number of devices is not limited.

Supported connection methods:

- Internet over Cloud Server;
- Wi-Fi client and access point;
- Bluetooth;
- Ethernet by IP or URL;
- USB OTG (Android only that support USB OTG);

Supported boards:

- Arduino UNO, MEGA, Leonardo, Pro Mini, Nano, MICRO and compatible AVR boards;
- ESP8266 boards;
- ESP32 boards;
- Chip KIT UNO32, Chip KIT uC32, Chip KIT Max32;

Supported communication modules:

- Bluetooth HC-05, HC-06 or compatible;
- Bluetooth BLE HM-10 or compatible;
- ESP8266 as modem;
- Ethernet W5100;

Supported IDE:

- Arduino IDE;
- FLProg IDE;
- MPIDE;

Supported mobile OS:

- Android;
- iOS;

Remote XY is easy way to make a unique graphical interface to control microcontroller device via mobile application, Arduino for example.

Remote XY allows:

- To develop any graphical management interface, using the control, display and decoration elements any combination thereof. You can develop the graphical interface for any task, placing the elements on the screen using the online editor. Online editor posted on the website .
- After the development of the graphical interface, you get the source code for the microcontroller that implements your interface. The source code provides a structure for interaction between your program with the controls and display. Thus you can easily integrate the control system into your task for which you are developing the device.
- To manage microcontroller device using your smartphone or tablet with the graphical interface. For manage used mobile application Remote XY.
- Using one mobile application, you can manage a large number of devices with different graphical management interfaces. As the interface description is stored on board the microcontroller device.

the source code for the microcontroller, for example a sketch for the Arduino, you opened it in the IDE. In the simplest case, you can download the sketch in Arduino board and test your interface immediately, connecting via a smartphone. But often you need to integrate the developed graphical interface with your problem you want to solve it using the Arduino. Next, we will show how it is easy to do.

The generated source code have a part that is starts with tag **Remote XY include library** and ends with tag **END Remote XY include**. This part of the code defines the structure of the graphical interface that you have created. In turn, this structure is also divided into parts. Consider the order.

Chapter 4

System Construction

4.1 System Description

The main processing brain of the system is the Arduino Nano and a Node MCU Microcontroller. Firstly, to run the project we need to connect it in power section. This project mainly powers from AC200V. After running this system, it will observe the transformer health condition and send it to mobile apps through IoT. All of this system maintain a microcontroller which is programmable.

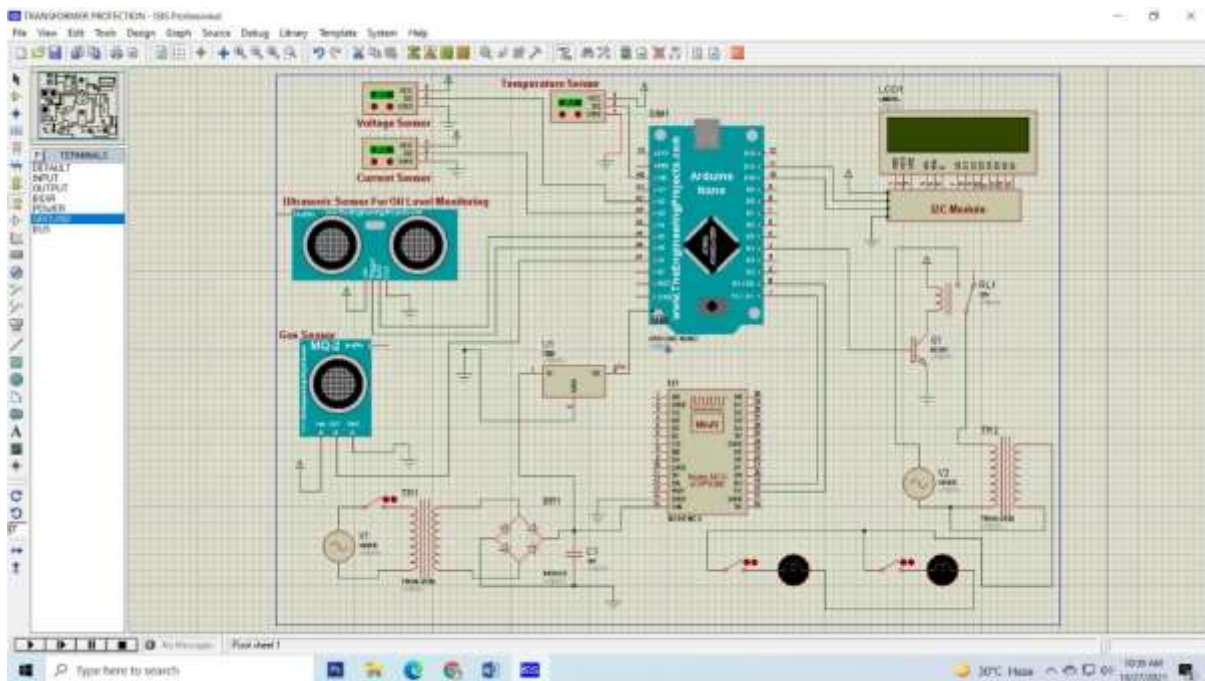


Figure 4.1: Our project in the Proteus Software

4.2 Block Diagram

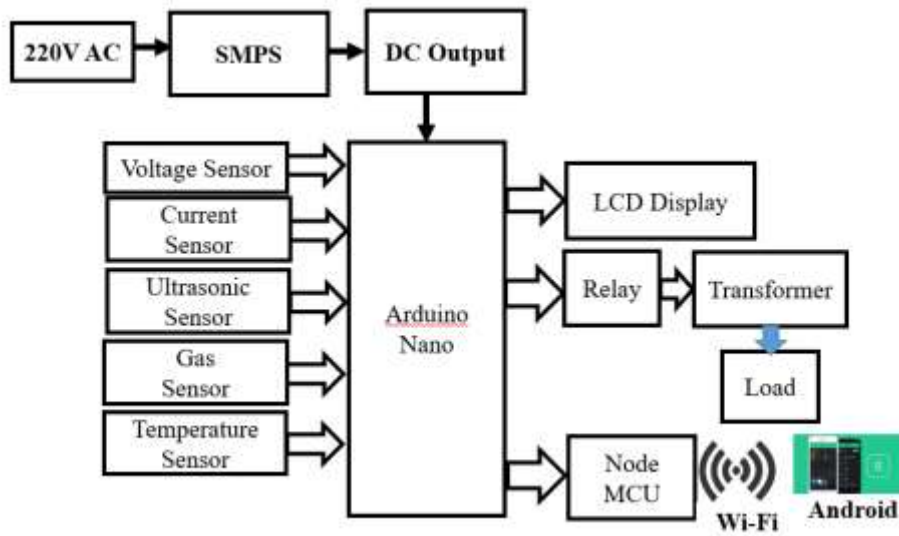


Figure 4.2: Block Diagram

4.3 Circuit Diagram

The schematic diagram here is representing the electrical circuit and the components of the hybrid power generation system. Here we have used standardized symbols and lines.

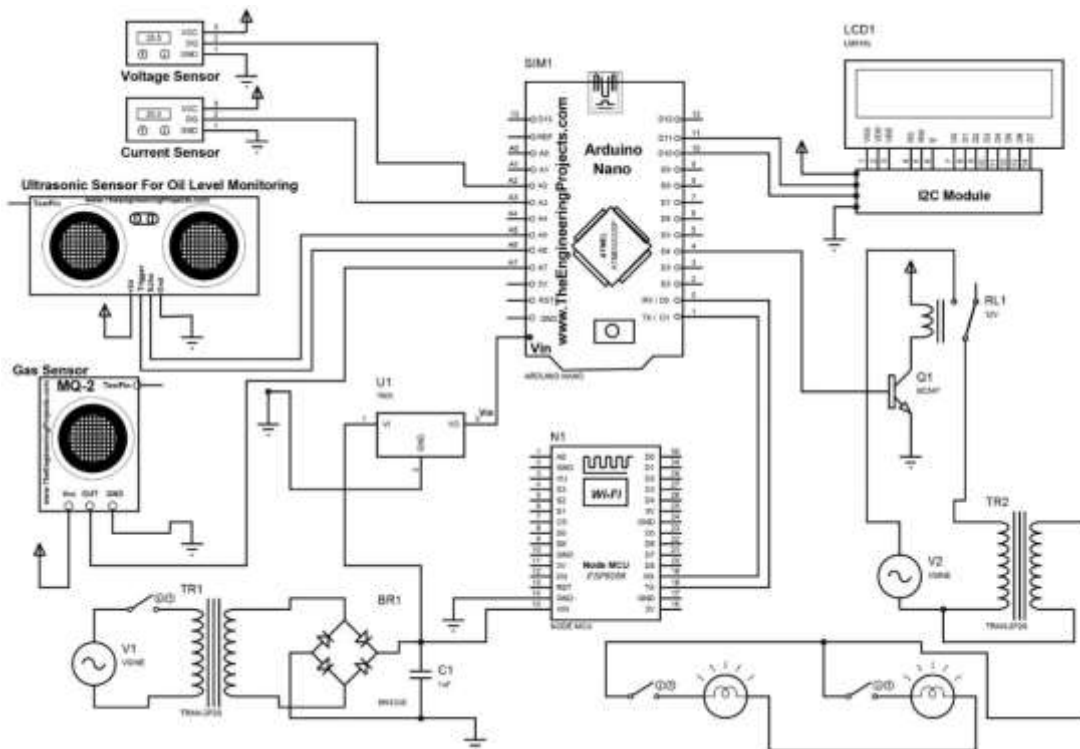


Figure 4.3: Schematic Diagram.

4.4 Working Principle

The power supply section of this circuit consists of a 220V AC from the mains supply which is stepped down by the SMPS (Switch Mode Power Supply) to 12V. The output is then feed to a filter to remove any A.C. components present even after rectification. The voltage is regulated using a voltage regulator to 5V which supplies power to the rest of the circuit. In this project we use a Arduino Nano and a Node MCU microcontroller as a main controller. Here we also use voltage sensor, current sensor, ultrasonic sensor, gas sensor, temperature sensor etc.

The main purpose of our project is to detect the transformer overall condition. When we first on the project, the project will be able to operate. Here we attached some sensor. Those are checked the transformer temperature, voltage, current and inside temperature oil condition. If there create abnormal situation the system will send a notification through an Apps. Temperature sensor measure the transformer inside oil temperature, Current and voltage sensor measure the transformer current and voltage part and Ultrasonic sensor measure the transformer inside oil quantity. Also, a flame sensor is added with this system. If there create flame it will be detect it and send a notification. It is the main purpose of this system.

4.5 The Project Prototype

The complete prototype of our project is shown below:



Figure 4.4: The Complete Prototype of the Project

4.6 Results

Now, it's time to talk about the results. We have written our commands using the Arduino IDE and the following things can happen:

- This project is connected with the AC220 Volt.
- When the system is on then it will measure the transformer oil temperature, oil condition.
- It will send in mobile apps through IoT.

4.7 Cost Analysis

In the below table we have summarized our project expenditure.

Table 02: Cost of Components with Price

Sl. no	Particulars	Specification	Qty.	Unit Price (Taka)	Total Price (Taka)
1	Node MCU	ESP8266	1	480	480
2	SMPS	5V, 5Amp	1	450	450
3	Gas Sensor	MQ2	1	120	120
4	Ultrasonic Sensor	R307	1	180	180
5	Voltage Sensor		1	220	220
6	Current Sensor		1	300	300
7	Arduino	Nano	1	400	400
8	Temperature Sensor	DHT11	1	180	180
9	Relay	5V	1	50	50
10	Others				1350
				Total	3730/=

Chapter 5

Conclusion

5.1 Advantages

There are many advantages of our project because of its accuracy. Some of the advantages are pointed out below:

- Able to take action for fire protection.
- increase the level of power system information
- improve the utilization efficiency of infrastructure in the existing power system.
- Automatic Monitoring the oil limit inside transformer.
- The whole system is automatic system.
- The system is user friendly.
- Able to detect fire.

5.2 Applications

The application areas for this project in this modern and practical world are huge and some of these are given below:

- Analyzing and locating fires
- Conducting search and rescue
- Monitoring hazardous variables
- It would be use in Substation
- In Powerplant

5.3 Future Scope of Work

We are thinking about adding many features to our project in the future to get more desirable out comes. Some of the steps that we are thinking about taking are given below:

- In future, we will try it to apply in real field.
- In future we will improve our operating system microcontroller to PLC system.
- In future we are thinking about adding more sensor to protect instant hazard.

5.4 Conclusion

A design of circuit for transformer monitoring. If transformer is in abnormal condition we can know from anywhere. No human power needed to monitor the transformer. Details about the transformer are automatically updated in webpage and notify when the transformer is in abnormal condition. We have described a remote monitoring system for distribution transformers utilizing the existing IoT, which has low investment and operation cost.

References

- [1] Walid K A hasan, Abobaker Alraddad, Abdulfatah Ashour, Yachao Ran (2019), “Design and Implementation Smart transformer based on IOT” published in IEEE explorer.
- [2] Sunitha Emi.P ,Sivasankari R, Ranjith Kumar.P Prabha R Jayageetha .J (2019), “Fault Detection in Transformer Using GSM Technology” published in 5th International conference on Advanced computing & Communication system (ICACCS).
- [3] Divyank Srivastava, M. M. Tripathi (2018) “Transformer Health Monitoring System Using Internet of Thing” published in 2nd IEEE international conference on power electronics intelligent control and Energy systems (ICPEICS-2018).
- [4] Rohit R. Pawar, Priyanka A. Wagh, Dr. S.B.Deosarkar (2017), “Distribution Transformer Monitoring System Using Internet of Things (IOT)” published in International conference on Computational Intelligence in Data Science (ICCIDS).
- [5] Rohit R. Pawar, Priyanka A. Wagh, Dr. S.B.Deosarkar (2017), “Health Condition Monitoring System for Distribution Transformer using IOT” published in International conference of computing Methodologies and Communication(ICCMC).
- [6] Mohamed Ahmed Eltayeb Ahmed Elmustafa Hayati Sherief F. Babiker (2016), “Design and Implementation of Low-Cost SMS Based Monitoring System of Distribution Transformers” published in Conference of Basic Sciences and Engineering Studies (SGCAC).
- [7] Xiaojiao Tang, K. Kobayashi, Y. Sonobe and M. Okazaki, "Development of 765kV transformer protection relay," 2011 International Conference on Advanced Power System Automation and Protection, Beijing, 2011, pp. 210-214

- [8] R. Grabovickic, C. Labuschagne, N. Fischer and O. Glynn, "Protection of transformer-ended feeders using multifunction relays," PES T&D 2012, Orlando, FL, 2012, pp. 1-10.
- [9] M. S. Deshmukh and V. T. Barhate, "Microcontroller based differential relay using fuzzy logic for transformer protection," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, 2017, pp. 712-717.
- [10] M. S. Deshmukh and V. T. Barhate, "Transformer protection by distinguishing inrush and fault current with harmonic analysis using fuzzy logic," 2016 IEEE International Conference on Control and Robotics Engineering (ICCRE), Singapore, 2016, pp. 1-5.
- [11] C. Lee and A. Fumagalli, "Internet of Things Security - Multilayered Method For End to End Data Communications Over Cellular Networks," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 2019, pp. 24-28.
- [12] Dharanya, S., M. Priyanka, R. Rubini, and A. Umamakeswari. "Real time monitoring and controlling of transformers." *Journal of artificial intelligence* 6, no. 1 (2013): 33-42.
- [13] Sun, Jialong, Shunbin Hu, Chenwei Hou, and Yanhong Wang. "Monitoring and control system of transformer temperature rising test." In 2011 International Conference on Electrical and Control Engineering, pp. 2365-2367. IEEE, 2011

Appendix

- Code used in the project is -