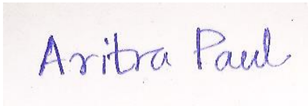
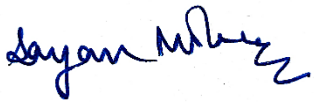
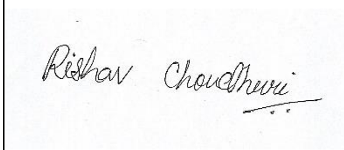
Remote monitoring and control of cooling system for an oil cooled transformer

Group No.	SL No.	Name	Roll No	Signature
XV	1	Aritra Paul	510619008	
	2	Ayan Kamar	510619020	
	3	Sayan Mitra	510619031	
	4	Rishav Choudhuri	510619052	

Forwarded by

Prof. D. Ganguly
(Convenor, DUGC)

Countersigned by

HOD,
E.E. Department

Acknowledgments

We wish to express our sincere appreciation to our advisor, Prof. Abhinandan De, who convincingly guided and encouraged us throughout the duration of what we have achieved so far in the project and still let us have our own free space of thoughts and decisions. Without his persistent help, the progress we have made so far in this project would not have been realized.

We would also like to extend our sincere thanks to Prof. P. Syam, the honourable HOD of our department, and Prof. K. Mukherjee for their valuable input and views about our project which helped us identify and solve some significant issues.

Lastly, we would like to thank our parents, friends, and seniors whose support kept us going throughout the project.

Remote monitoring and control of cooling system for an oil cooled transformer

Mini-Project Team 15
Aritra Paul, Ayan Kamar, Rishav Choudhuri, Sayan Mitra
(510619008, 510619020, 510619052, 510619031)
Advisor - Dr. Abhinandan De
Department of Electrical Engineering
Indian Institute of Engineering Science and Technology, Shibpur

Abstract:

The project aims to make a low-cost monitoring system for distribution transformers which can be extended to power transformers (with limitations in monitoring capabilities). We plan to use multiple sensors which will measure certain parameters like oil temperature, oil level, secondary voltage and current, winding temperature, etc., and then feed data into a microcontroller. The microcontroller will process the signals and display the data in an LCD screen. If any parameter goes above/below a certain threshold, then an alarm will go off. All the data will be uploaded into a cloud server which will be used for remote monitoring of the transformer as well as controlling the cooling system. It can also be used for data logging, data analysis and predictive maintenance.

I. Introduction:

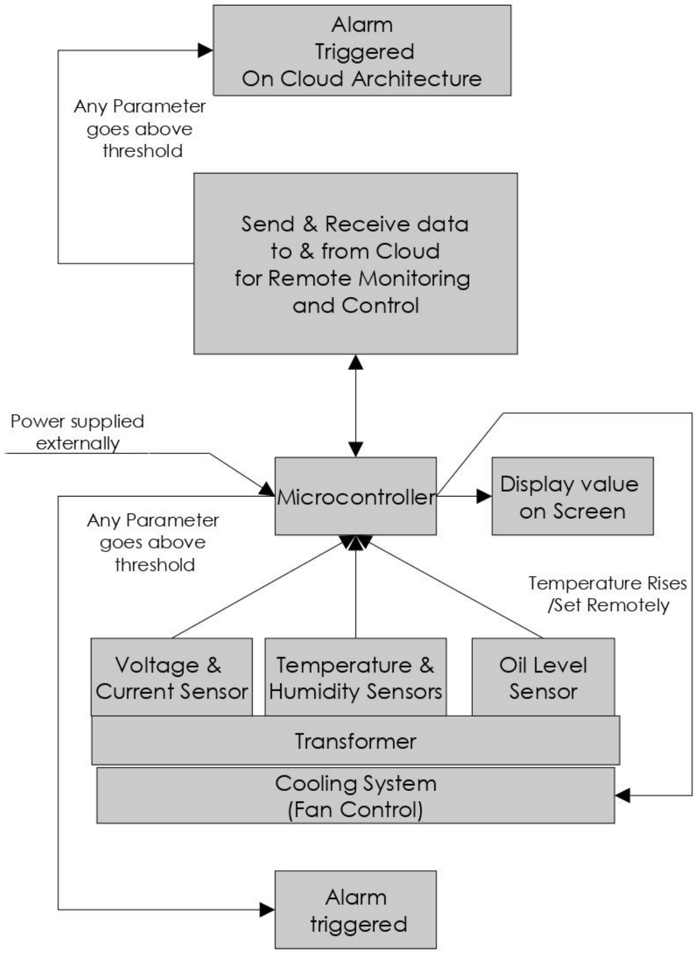
Transformers are arguably most important components of any power system. Transformer is the main reason for the widespread popularity of AC systems over DC systems. It will be safe to say that almost entire world production of electrical energy is transformed twice, thrice, or more times before utilisation.

A distribution transformer or service transformer is a transformer that provides the final voltage transformation in the electric power distribution system, stepping down the voltage used in the distribution lines to the level used by the customer. Usually, Primaries are provided power at the standard distribution voltages used in the area; these range from as low as 3300 volts to about 33,000 volts depending on local distribution practice and standards (as per BIS) at 50 Hz, but many other voltages are standard. 415V (Line voltage) is given out on the secondary side. The primary coils are wound from enamel coated copper or aluminium wire and the high current, low voltage secondaries are wound using a thick ribbon of aluminium or copper. The windings are insulated with resin-impregnated paper. The entire assembly is baked to cure the resin and then submerged in a powder coated steel tank which is then filled with transformer oil (or other insulating liquid), which is inert and non-conductive. The transformer oil cools and insulates the windings, and protects the transformer winding from moisture, which will float on the surface of the oil. The tank is temporarily evacuated during manufacturing to remove any remaining moisture that would cause arcing and is sealed against the weather with a gasket at the top.

Distribution transformers are the backbone of any modern power distribution system and a vital asset of power utility companies. Vast sum of capital expenditure is spent to install these transformers. So, it is important to detect any unnatural behaviour of transformer as early as possible. To increase reliability and achieve optimum operation of distribution network, it is necessary to monitor the transformer conditions and act accordingly. Operation of distribution transformer under rated condition (as per specification in their nameplate) guarantees their long service life. However, their life is significantly reduced if they are subjected to overloading, heating, low or high voltage/current resulting in unexpected failures and loss of supply to many customers thus effecting system reliability. Abnormality in distribution transformer is accompanied with variation in different parameters like Winding temperature, Oil temperatures, Ambient temperature, Load current, Oil flow (pump motor), Moisture and dissolved gas in oil, LTC monitoring, Oil level, Bushing condition [1]. Overloading, oil temperature, load current and ineffective cooling of transformers are the major causes of failure in distribution transformer. When a transformer fails, an adverse effect occurs in the continuity of transmission and distribution systems resulting in increase of power system cost and decrease of reliability in electric delivery.

Since knowing the transformer's condition is the key to determine the proper maintenance action, the utility should be able to have such knowledge. The condition can be well defined if the data acquired is valid. According to the above requirements, we need a distribution transformer real-time monitoring system to monitor all essential parameters operation and send to the monitoring centre in time. It leads to online/remote monitoring of main functional parameters of distribution transformers which will provide necessary information about the health of distribution transformers. This project aims to build such capability.

The aspect of remote monitoring is also important in the context of Industry 4.0 and concept of Internet of Things. As we move toward a more IoT centric world, it is only natural to incorporate such ideas with electrical utilities also. As disperse energy generation (mostly wind and solar) become more prevalent, we need to incorporate software controlled smart transformer in smart grids for smooth grid operation [2]. While such aspect is beyond the scope of our understanding at this stage, but remote monitoring of transformer may be the first step of such a system.



Block Diagram

II. Parameters monitored

There are numerous different parameters which can be assessed for monitoring purpose of a transformer and to determine aging and loss of life calculation. These are namely winding and oil temperature, ambient temperature and humidity, load profile (current, power factor, power), Oil level, Oil flow rate, Vibration and noise level, oil quality and dissolved gas analysis, Insulation property, Load tap changing condition, harmonic analysis etc.

Among these, we have chosen properties which are under the scope of remote monitoring and most important. Here is a short description of how they affect transformer health.

i. Winding and Oil temperature:

The rating of a transformer has an almost exclusive thermal basis, the limitation being the maximum temperature of working for which the insulation will have a reasonable economic life. The winding I^2R losses, the core losses and the stray losses in the tank and metal support structures are the prime sources of heat that cause the oil and winding temperature to rise. It is natural that the winding / coil temperature where the I^2R losses are taking place is higher than the oil temperature. Accordingly, often a parameter called

Hotspot temperature (HST) is introduced. Hotspot temperature is the hottest temperature inside the winding of the transformer. The hotspot location on winding is depended to the design of the transformer. The traditional insulation system consists of oil and paper. Over time, the paper insulation used in transformer winding loses mechanical and electrical strength and becomes brittle when exposed to elevated operating temperature. Various investigators have not agreed on life-duration at any given temperature. However, they do agree, that between 353 K(800C) to 413 K(1400C), the rate of loss of life due to aging of transformer insulation is "doubled" for every 6 °C rise in temperature [3]. Thermal aging of transformer insulating material is related with the chemical reactions – pyrolysis, oxidation, and hydrolysis. As also pointed out by case study in reference [3], Overheating is a prime cause of transformer failure. So, it is important to monitor this parameter and to control cooling systems accordingly.

ii. **Ambient Temperature:**

As pointed out earlier also the temperature of a transformer, often considered as hot spot temperature (HST), is primarily controlled by ambient temperature and load, where ambient temperature is influential factor. On one hand, some types of load are directly affected by ambient temperature, such as the cooling load in summer: the higher the ambient temperature is, the greater the load will be. On the other hand, the load capability of a transformer is generally governed by ambient temperature. Transformers must operate under the prescribed limit of HST, and ambient temperature is an uncontrollable factor in the influential factors of HST (ambient temperature and load); therefore, if the ambient temperature is high, the load capability of a transformer is always low, From these aspects, it can be seen that ambient temperature is an important factor in estimating the transformer life.

iii. **Humidity:**

Humidity in external environment may introduce moisture in transformer oil which in turns can be harmful for transformer. While a small portion is found in the oil, some 98 to 99 percent of it becomes diffused in the of paper (cellulose) insulation [4] which in turn makes transformer increasingly exposed to electrical malfunction. While breather is employed to tackle the issue, it may be till interesting to understand other effects of humidity on transformer by recording humidity.

iv. **Load profile (Secondary Voltage, Secondary Current, Frequency):**

Loads for transformer, specially loads for distribution transformer varies widely over a typical 24-hour day or in different seasons of year. So, it may be important for utility to register continuous load profile for a certain transformer. As load increases, transformer losses and hence temperature also increases. But this relationship is not linear. the long thermal time constants of a transformer make the relationship between load and transformer temperature highly dynamic. This means that the temperature is dependent not only on the present load, but also on the loading in the previous hours. As pointed out in reference case study [3], in India, most failures of distribution transformers occur even before the life of 3 years due to overloading. So, it is important for utility companies to keep a track of loading patter, and hence load profiling is important.

v. **Oil Level:**
In oil cooled liquid immersed transformer, oil plays a very important part in both cooling and insulation. The oil in the ducts, and at the surfaces of the coil and cores, takes up heat by conduction and rises, cool oil from bottom of tank rising to take its place. A continuous circulation of oil is completed by the heated oil flowing to the tank sides or to the radiator tubes/ fins, depending on tank design, where cooling in ambient temperature occurs. Conservators are required to take up the expansion and contraction of the oil with changes of temperature in service without allowing the oil to come in contact with the air.
An oil level monitoring system is required so that the correct oil level can be maintained. Maintaining the proper oil level is extremely important because if the oil level falls below the level of the radiator inlet, flow through the radiator will cease and the transformer will overheat. Also, reduction of oil level indicate leak in tank or radiator, making the oil in direct contact with air and hence introducing moisture to oil. So, it important to periodically monitor oil level.

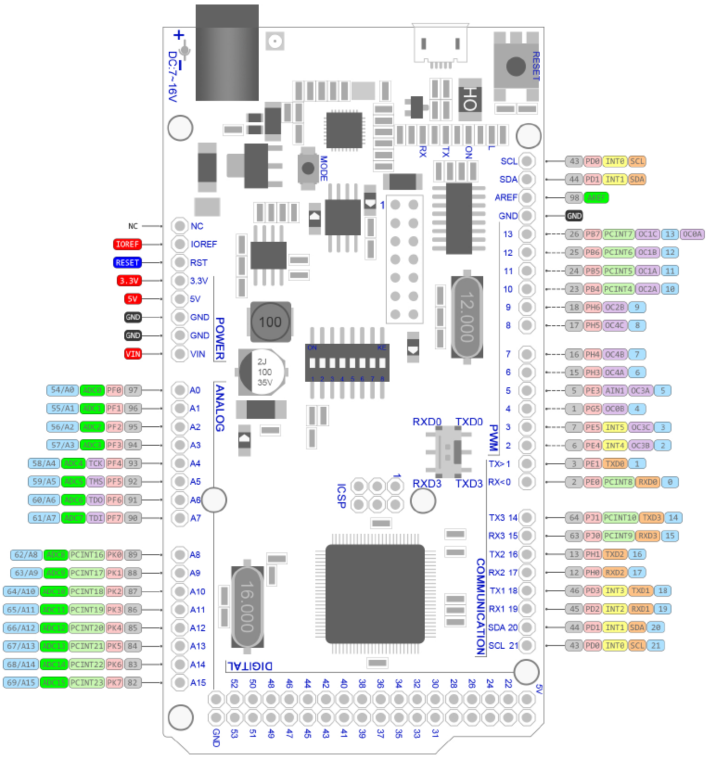
III. Monitoring System

The monitoring system will consist of a microcontroller that will suit our needs along with a range of required sensors/modules (may be further added upon) which are listed below:

a. Microcontroller: Arduino Mega + NodeMCU ESP8266

About: It is a customized version of the classic ARDUINO MEGA R3 board. Full integration of Atmel ATmega2560 microcontroller and ESP8266 Wi-Fi IC, with 32 Mb (megabits) of flash memory, and CH340G USB-TTL converter on a single board. All the modules can work together or separately. And everyone has their own pinout headers. The operating mode is selected by means of DIP switches on-board.

Reason for using: We required a microcontroller that had many I/O pins and could also connect to the Internet using Wi-Fi and send and receive data. This microcontroller served both purposes and was available at a reasonable cost, hence we decided on using this microcontroller board.



Microcontroller Pinout Diagram

b. Temperature sensor: **MAX6675 Module**

About: The MAX6675 performs cold-junction compensation and digitizes the signal from a type-K thermocouple. The data is output in a 12-bit resolution, SPI-compatible, read-only format. This converter resolves temperatures to 0.25°C, allows readings as high as +1024°C.

Reason for using: This sensor is used to measure the temperature of the oil and the windings inside a transformer. The alternatives to this sensor were using thermistors or sensors like LM35 both of which are impractical to use in this scenario. Dipping a thermocouple junction in the oil makes much more sense, hence we decided to use this sensor.

Pin details:

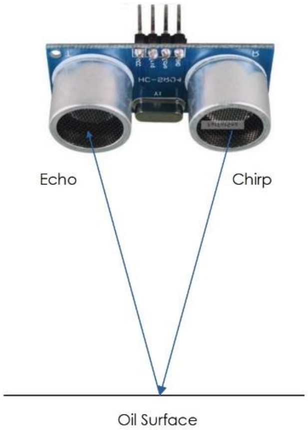
PIN NAME	FUNCTION
VCC	Positive Supply (-0.3 to +6V)
GND	Ground
SCK	Serial Clock Input
CS	Chip Select
SO	Serial Data Output
T+	Alumel Lead of Type-K thermocouple
T-	Chromel Lead of Type-K thermocouple

#The placement of the sensors in the transformer will vary according to the design and the hotspot position of the transformer.

c. Oil level sensor: Ultrasonic Ranging Module (HC-SR04)

About: Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The module includes ultrasonic transmitters, receiver, and control circuit. The Module automatically sends eight 40 kHz chirps and detects whether there is a pulse signal back.

Reason for using: This sensor is used to measure the oil level. This can accurately measure distances in the range of 2-400cm and it is a cheap and easy to use sensor. On testing this sensor, we found out that it can accurately measure the water level and hence it can be expected to measure oil levels as well. It will be set with the sensor faced perpendicular to the oil surface for best and most accurate results.



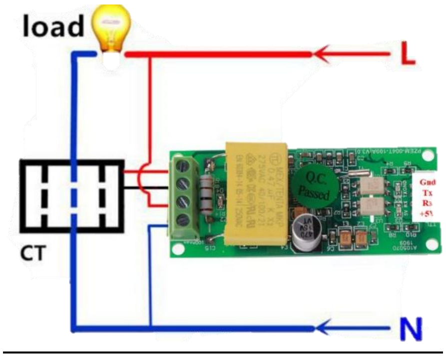
Pin details:

PIN NAME	FUNCTION
VCC	Positive Supply (5V)
GND	Ground
Trig	Trigger Pulse
Echo	Receive Pulse

d. Voltage and Current sensor: PZEM-004T V3.0

About: This PZEM-004T module is capable of measuring AC (RMS) voltage, current, and power (single-phase). The unit easily interfaces with microcontrollers like Arduino and other hardware using the code library.

Reason for using: It is used to measure the secondary voltage, current and frequency. This sensor is capable of measuring voltages from 80-260V AC, current up to 100A, frequency from 45-65 Hz. It is one of a kind sensor module that can measure such high values of voltage and current and send out the data via Serial Interface. It can be connected easily to microcontrollers using the Serial Interface and even be directly connected to computers using a TTL-USB convertor.



Pin details:

PIN NAME	FUNCTION
VDD	Positive Supply (5V)
GND	Ground
Rx	Receive Pin of Serial Communication
Tx	Transmit Pin of Serial Communication

Note – There are 4 more terminals. 2 of them connect to the 2 CT terminals and the other 2 connect to the two terminals across which voltage is to be measured.

e. Ambient Temperature and humidity sensor: DHT11

About: The DHT11 features a temperature & humidity sensor complex with a calibrated digital signal output. It ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component.

Reason for using: This sensor is used to measure the ambient temperature and humidity. This is a carefully calibrated sensor; hence it is extremely accurate in measuring the temperature and humidity.

Pin details:

PIN NAME	FUNCTION
VCC	Positive Supply (3 - 5.5V)
GND	Ground
Data	Data Output Pin

f. Display:

In addition to the data being uploaded to the cloud for remote monitoring, the values can also be viewed onsite on 2 LCDs. One of the LCDs will display the critical parameters such as voltage, current, oil temperature, winding temperature. The other LCD will display the less critical parameters such as oil level, ambient temperature, frequency and humidity.

The LCDs will be connected to a I2C Serial Interface Adapter Module. This will reduce the number of pins from 16 to just 4. This module is cheap and in addition to reducing the number of pins, it also reduces the number of wires used. It communicates using the I2C protocol which is simple to implement using the microcontroller we have chosen.

g. Alarm System:

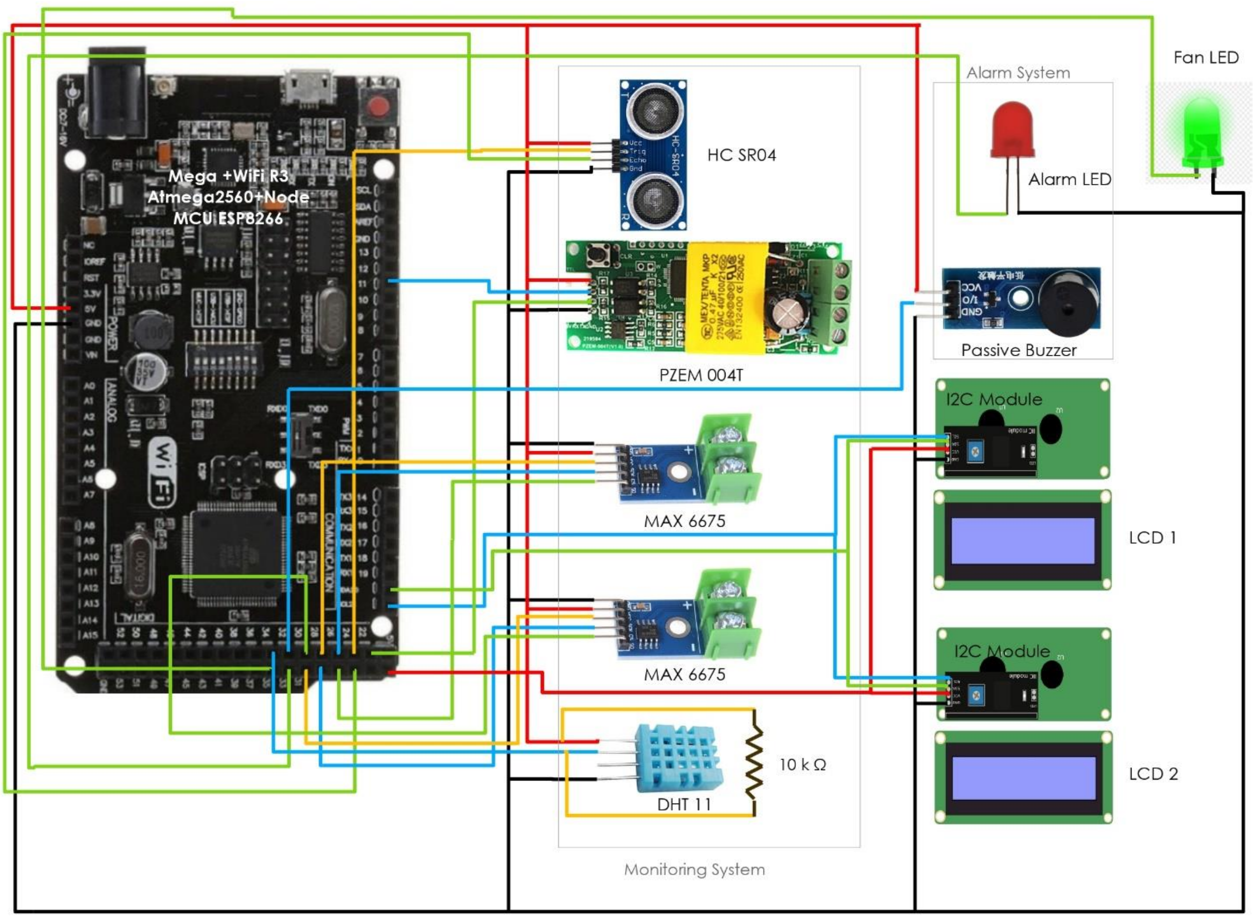
In the prototype that we have developed, we have also made the provision of an ON-SITE Alarm System, which consists of an LED and a Piezo Buzzer. The alarm gets triggered when certain conditions are met by the parameters.

h. Fan Control:

The transformer cooling fans are sturdy, weatherproof fans for accelerating the cooling process in oil-cooled electrical transformers. They circulate air around the exterior of the transformer's radiator as oil works it way through the inside to dissipate heat and reduce the operating temperature of the transformer. These fans can be centrifugal (forcing out hot air) or Axial (Forcing in cool air on the radiator) and are usually run by induction motor, be it single phase or three phases.

In Our model we wish to ON or OFF the fan with the help of a 5 V DC control AC Power relay module/Switch. Whenever the conditions are met to start the fan (as discussed earlier) OR the command is given from remote server, the relay is excited with 5 v DC, and switch goes to ON state to run the fans.

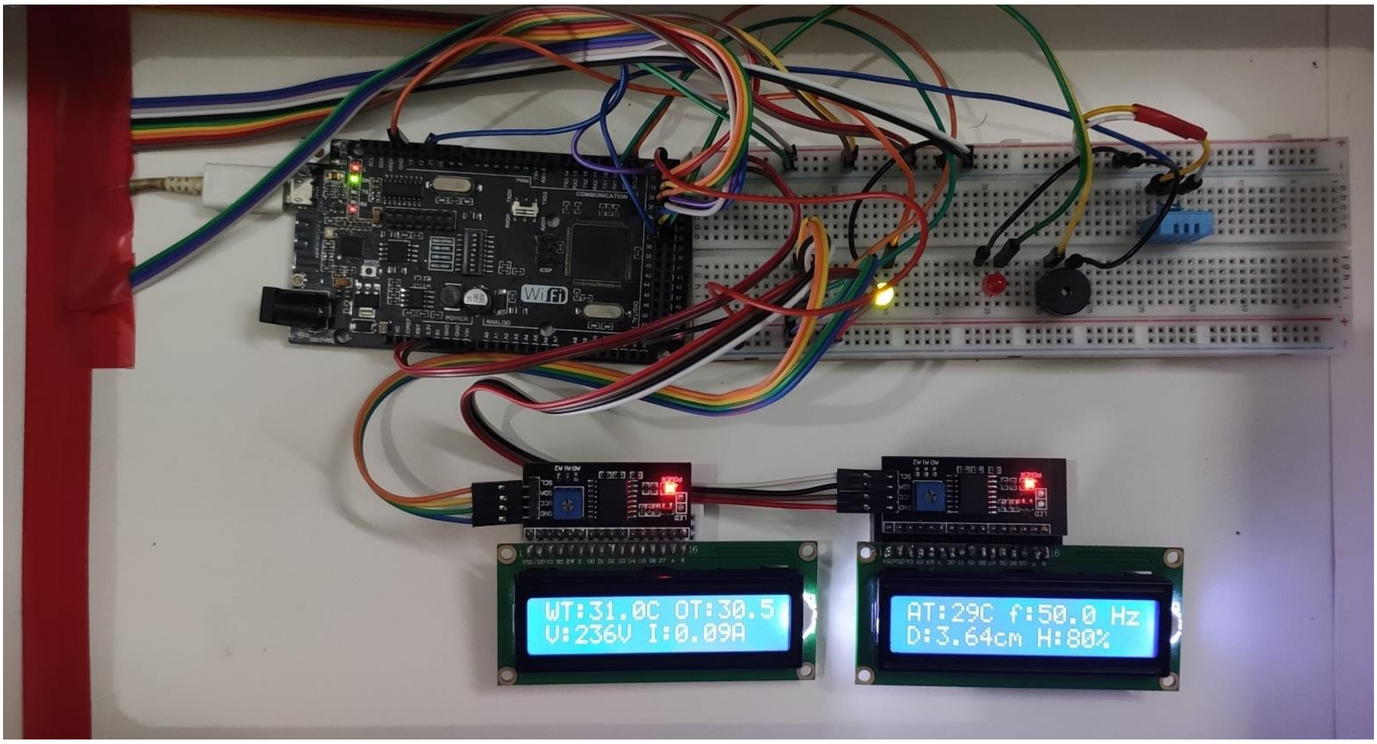
IV. Connection Diagram



Connection Diagram

SENSORS	PIN SELECTION	
PZEM 004T	Tx	22
	Rx	11
MAX 6675 (for winding temp.)	SCK	28
	CS	26
	SO	27
MAX 6675 (for oil temp.)	SCK	31
	CS	29
	SO	30

SENSORS	PIN SELECTION	
HC SR04	Trig	24
	Echo	25
DHT 11	Data	34
I2C Module/LCD	SDA	SDA (20)
	SCL	SCL (21)
Passive Buzzer	I/O	32
Alarm LED	I/O	33
Fan LED	I/O	35



Final Circuit

V. Remote Operation:

Choosing Cloud Servers:

The cloud servers are an integral part of our project. For the *'remote monitoring and control'* aspect, we need cloud servers wherein all the data acquired from the sensors are sent to it. The alternative to this was setting up servers of our own and develop web software, which is complicated, time consuming, a more expensive option and not under our skill set.

The requirements of the cloud server for the ***'remote monitoring'*** aspect were:

- **Have enough data fields** to store the multiple data it is being sent periodically.
- Be **compatible** to receive data from our microcontroller.
- Can **store data for long time periods.**
- **Have a dashboard** to visualise all relevant data.

After researching about many platforms and trying them out, the cloud server we finally chose for the *'remote monitoring'* part is **ThingSpeak**. ThingSpeak is an ***"IoT analytics"* platform** service that **allows us to aggregate, visualize, and analyse live data streams in the cloud.** We can send data to ThingSpeak, create instant visualizations of live data, and send alerts using web services like Twitter and Twilio. **With MATLAB analytics inside ThingSpeak, we can write and execute MATLAB code to perform pre-processing, visualizations, and analysis.** ThingSpeak enables to prototype and build IoT systems without setting up servers or developing web software. **The platform matched all our requirements.** It can receive data up to 8 data fields, was compatible with our microcontroller, was easy to interface and has many other useful features.

For the *'remote control'* aspect however, we decided to go with a different platform since ThingSpeak does not have the necessary features required to communicate instructions back to the microcontroller.

The requirements of the cloud server for the ***'remote control'*** aspect were:

- Able to **send instructions to our microcontroller** in near real time.
- Possess **at-least 3 data fields** to receive data periodically.
- Be **compatible** with our microcontroller.

After going through the various platforms available as open source, we decided to go with the **Blynk** platform. **Blynk is a hardware-agnostic IoT platform** with white-label mobile apps, private clouds, device management, data analytics, and machine learning. It was one of the few platforms that was compatible with our microcontroller board. On testing the platform, **we were able to send the critical data necessary** (oil, winding, and ambient temperature) to monitor and **let the user decide the fan state and control it correspondingly**. We also were able to indicate the fan state on the platform. The control of the fan state could be done in near real time with an approximate latency of about 10ms.

Setting up ThingSpeak:

To use ThingSpeak as a cloud server, we need to establish a connection between the server and the client (our microcontroller/device) so that both parties can communicate with each other. The interface of ThingSpeak is from its website. Widgets with limited functionality are also usable on Android Phones for ease in data viewing and alarm.

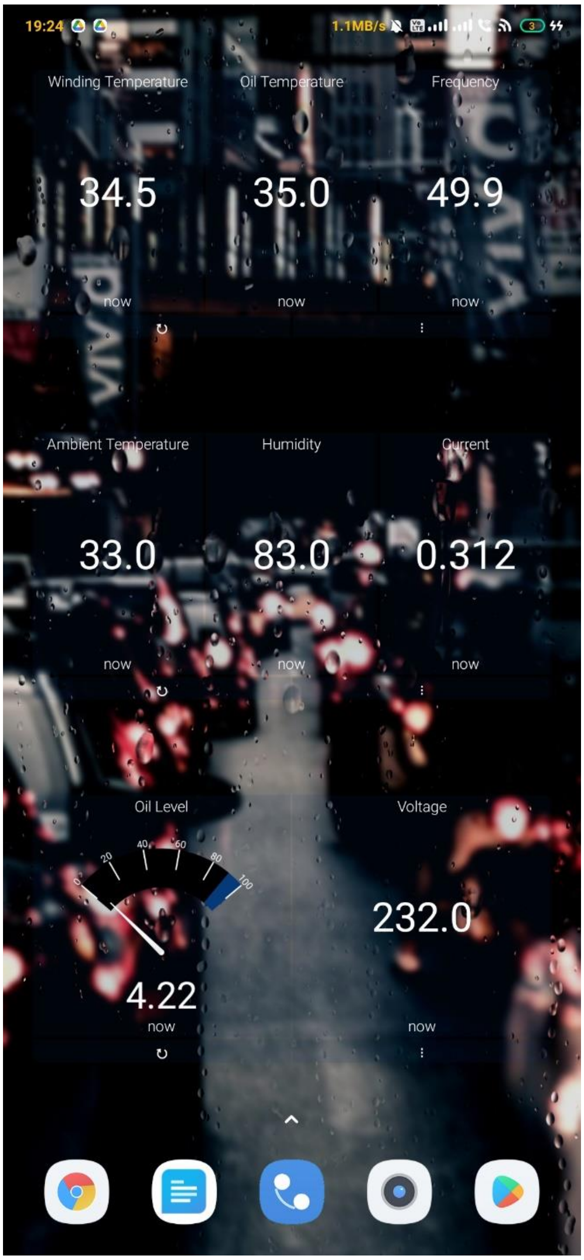
The process of setup of the server was:

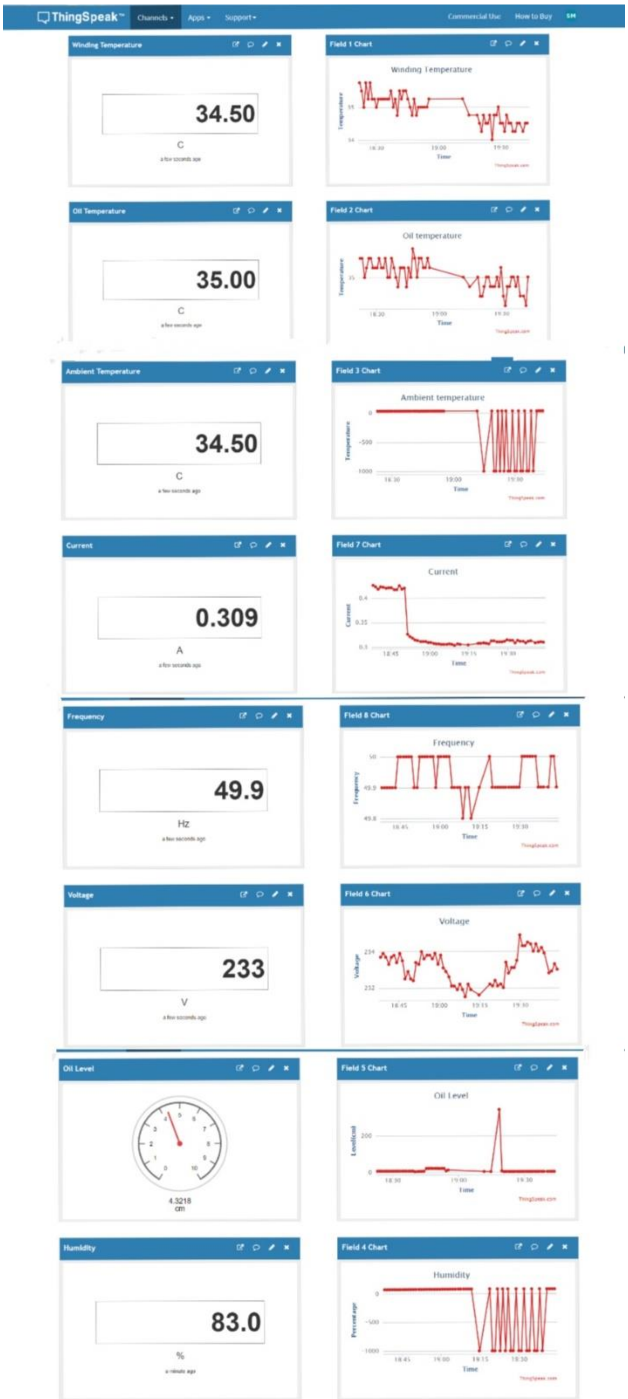
- On ThingSpeak, after creation of a new project, Channel ID, Read and Write API keys are provided for secure communication between the ThingSpeak servers and the microcontroller.
- On the website, a dashboard has to be built which shows the required data effectively and in meaningful way with each widget in the dashboard linked with a respective Field as set in the Arduino(microcontroller) code.
- **On the client side (i.e., the microcontroller/device side):**
- We used the in-built library **<ThingSpeak.h>**
- While programming the microcontroller, we had to **mention the unique *Channel ID, ReadAPIKey and WriteAPIKey*** as described before. The following lines of code do that:

```
char* writeAPIKey = "DHNDHNM9Z9G7ZPWV";          // Write API Key
char* readAPIKey = "9S7OHAFY7AZ9768C";           // Read API Key
const long channelID = 1274361;                  // Upload Channel ID
```

- **To write data from device to ThingSpeak server**, we used the following line to upload the data through various fields defined by their Field numbers and all of the data is uploaded simultaneously using the function write2TSData()

```
ThingSpeak.setField( TSField1, field1Data );
```





ThingSpeak Dashboard

Setting up Blynk:

To use Blynk as a cloud server, we need to establish a connection between the server and the client (our microcontroller/device) so that both parties can communicate with each other. The interface of the Blynk is from its app present on a mobile phone. We can send and receive data to and from our device using the app. The app can be installed on any Android/iOS running smartphone.

The process of setup of the server was:

- On the Blynk app, we need to create a "new project". In that, we selected the device we are using - Arduino Mega and the connection type – Wi-Fi.
- On creating a new project, a ***"Auth Token"*** was sent to the email ID that was used to register/login on the Blynk App. **This is a unique ID designed for authorisation and security purposes.** We program the device in such a way that it only sends and receives data to and from a Blynk server with that particular "*Auth Token*".
- On the app, we then enter a dashboard, where we can add "*widgets*" of our own choice which suits our requirements. We chose to add the following widgets:
  1. **"Labelled Value"** (x2) – **Displays the data value** entered to its "*virtual pin*".
  2. **"SuperChart"** (x1) - **Displays the graphs** (charts) of the data streams selected.
  3. **"LED"** (x1) – **Basically behaves as a virtual LED**. Lights up when there is a HIGH signal provided to its "*virtual pin*" and stays off on LOW signal.
  4. **"Button"** (x1) – When this button is pressed, it sends a signal back to a microcontroller pin which **controls the fan state.**

On the client side (i.e., the microcontroller/device side):

- **For simplicity**, we used the in-built library **<BlynkSimpleShieldEsp8266.h>**
- While programming the microcontroller, we had to **mention the unique *"Auth Token"*** as described before. The following line of code does that:

```
char auth[] = "Hq3OHNa1u_eubVi6XmX5RIEuAE0NcWRD";
```

- To **initialise and establish a connection** between the Blynk server and our device, we write the following piece of code:

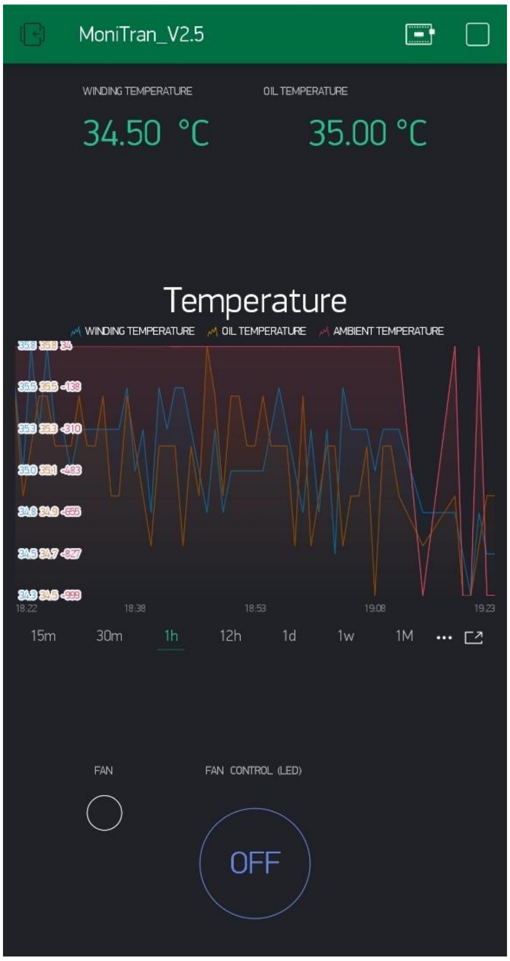
```
Blynk.begin(auth, wifi, ssid, password);
```

- **To write data from device to Blynk server**, we used the following lines of code in a function named as BlynkWrite()

```
Blynk.virtualWrite(V5, celsiusW);          //Sends Winding Temperature through Virtual Pin 5
Blynk.virtualWrite(V6, celsiusO);          //Sends Oil Temperature through Virtual Pin 6
Blynk.virtualWrite(V7, DHT.temperature);   //Sends Ambient Temperature through Virtual Pin 7
```

- **To keep the communication channel open to Blynk server open and functional**, we wrote the following line of code:

```
Blynk.run();
```



Blynk Dashboard

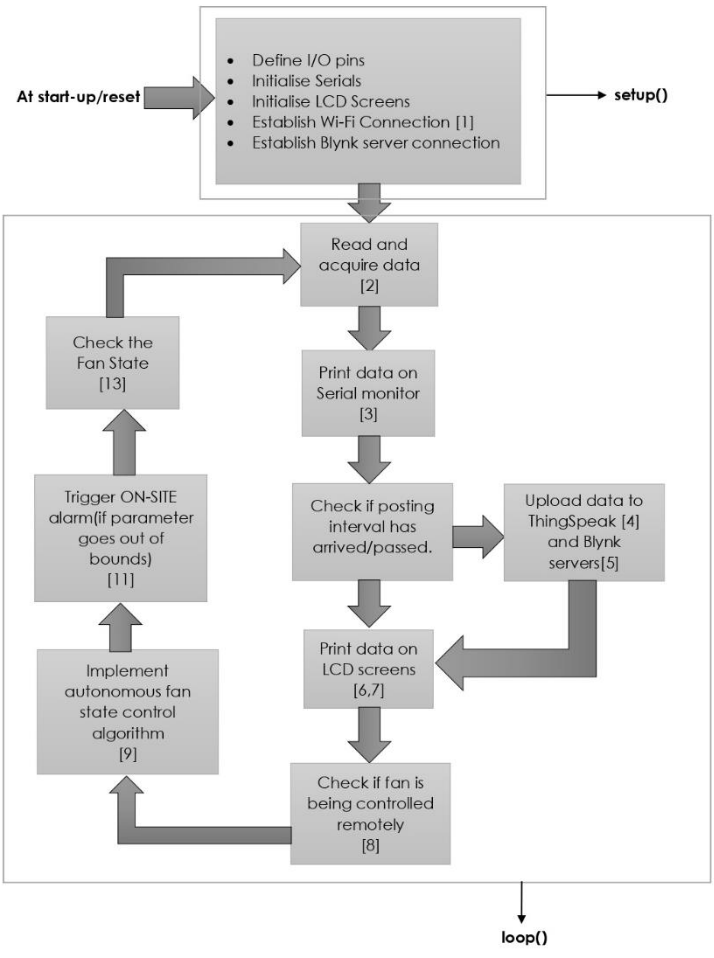
VI. Code Developed:

To keep the code simple and easily readable, we majorly did the following things:

- **Used in-built libraries** – The in-built libraries are used to interface with our multiple sensors, establish Wi-Fi connection, and responsible for data transfer functions to the cloud servers.
- **Divided sections of code into functions** – The various aspects were subdivided into separate functions and then called whenever they were required. Besides the mandatory setup() and loop() functions, there are in total of 13 functions. They are:
    1. **connectWifi() –** This function is responsible for **establishing the internet connection** at start-up as well as whenever the connection is lost. It also **establishes a connection between our device and the Thingspeak Server.**
    2. **DataReadAcquire() –** This function is **responsible for reading and acquiring the data from the multiple sensors** used. The data is stored in their relevant variables which are declared globally to be accessible from any function.
    3. **Serial_Print() –** This function is responsible for **printing all the data out on the serial monitor.**

4. **write2TSData() –** This function is **responsible for uploading data to the ThingSpeak server**. It takes 17 arguments and returns a number (0 or !) indicating the nature of the upload.
5. **BlynkWrite() –** This function **is responsible for writing data to the Blynk servers** on their respective virtual pins.
6. **LCD1_PRINT() –** This is **responsible for printing to the 1st LCD Screen.**
7. **LCD2_PRINT() -** This is **responsible for printing to the 2nd LCD Screen.**
8. **Fan_Flag() -** This function **checks if the fan is being controlled remotely via Blynk.**
9. **Auto_FAN() –** This function **implements the algorithm for autonomous fan state control.**
10. **FanParametersOutOfBound() –** This function **checks if the parameters which decide the fan state are within normal range** or not.
11. **Alarm_Trigger() –** This function is **responsible for triggering the ON-SITE alarm system** in case the parameters cross their normal range of operation.
12. **AlarmParametersOutOfBound() –** This function **is responsible for checking if all the relevant parameters are within their normal range of operation.** Returns a Boolean output after checking the data.
13. **Fan_State() –** This **function checks the state of the fan**. This is used to avoid conflicts with the autonomous control and the remote control of the fan state.

Code Structure:



NOTE – The number indicated in the brackets refer to the function used to implement them.

VII. Algorithm for Active Cooling System Triggering

a. Based on temperature: When the difference between ambient temperature and oil temperature OR difference between oil temperature and winding temperature exceed a certain value, the fan will be triggered. The certain values will depend on construction, type, rating, and particular thermal modelling of transformer
b. Based on Loading: If load current increases a certain value the, the fan will be started
c. Based on Predictive analysis: From previous load profile or temperature data, it can be predicted when a particular transformer will need to be forced cooled

VIII. Use of Collected Data

1. Predictive Maintenance (PdM):

Any plant, system or Electrical Installation require regular maintenance for reliable operation. However, the difference between normal maintenance and predictive maintenance (Also called condition-based maintenance) is that in predictive maintenance we monitor the performance and condition of equipment during normal operation to reduce the likelihood of failures. The goal of predictive maintenance is the ability to first predict when equipment failure could occur (based on certain factors), followed by preventing the failure through regularly scheduled and corrective maintenance.[22]
Predictive maintenance cannot exist without condition monitoring. And there lies the use of our system. In the case of transformer, we can make a model based on condition monitoring to schedule the maintenance. For example, we can say if a transformer has operated a certain amount of time with the oil temperature being above 600C, then probably the oil quality degradation needs to be checked and oil need to be replaced. Similar maintenance action may be required depending on other parameters.
If we have large scale data of many transformers, then it is possible to use advance algorithms and machine learning to predict the maintenance schedule based on past equipment failure history.

2. To assess remaining Useful Life and to allow planned operation beyond continuous design limits:

By monitoring the thermal parameters of the transformer, it is possible to assess the remaining transformer life. Moreover, it is possible to identify when the unit is capable of operation more than its nominal design rating and thus utilise the plant to its optimum capability. It is also possible to identify any operating mode likely to cause damage or premature ageing of the insulation and take timely preventive action.

3. Load profile of Transformer:

Transformer plays a fundamental role in the process of transmission and distribution and may cater to an industry, industrial complex, or a residential area. If we have real time data on electrical power flow/ load current through transformer then it is possible to realise the load profile of the transformer and to understand power demand from a particular area. In this way it is possible to make a power usage map of different area over time, which may be used and analysed for various engineering or economic interest. It helps to plan new transformer installation, to optimise the location of transformer based on forecasting of customer load and helps in power demand management.

4. Optimising Cooling and optimised use of cooling fan

As for the ONAF or OFAF type transformer, it may be waste of energy to always run the fan even in the low load period. We have already proposed an algorithm to run the cooling fan based on the difference between ambient and oil temperature and other parameters. However, with the past data on temperature and load profile in hand, we can use this data to predict when the fan needs to be started, thus optimising cooling process and reducing the energy cost of cooling operation. Same analogical operation can be employed for oil circulation application also.

5. Analysis and Modelling of Data

As we are using the ThingSpeak platform (a MATLAB product) as our remote server, All the data can be used in different MATLAB modelling and analytical research purpose.

IX. Project Cost:

We have been successful in building a substantially cheap monitoring system which can be employed on a large scale for distribution transformers without adding much to the already high capital cost of transformers. We have even successfully managed to limit the project cost to less than our agreed budget of INR 4000.

Component	Quantity	Quoted Rate	Price (in INR)
Arduino Mega+NodeMCU ESP8266	1	1090	1090
Ultrasonic Sensor (HC-SR04)	1	75	75
MAX6675 Module (Temp sensor)	2	325	650
PZEM-004T	1	899	899
IIC/I2C Serial Interface Adapter Module	2	79	158
LCD1602 Parallel LCD Display	1	106	106
Male to Female Jumper Wires 40 Pin 30cm	1	65	65
170 pts Mini Breadboard SYB-170 Black	1	49	49
Components already existing (Jumper wires, breadboard, resistors)	N/A	N/A	150
		TOTAL	**3242**

X. Conclusion:

- We have researched extensively about distribution transformers and identified the parameters that affect its health, and which can be remotely monitored.
- Learnt about different cooling mechanism of transformers and narrowed down our focus on ONAF/OFAF type.
- Appropriate sensors have been decided upon to measure the parameters.
- A proper microcontroller has been identified that serves all the purposes including uploading and receiving data via Wi-Fi.
- A final connection diagram has been made which will be followed to make the hardware.
- Financial planning has been done by fixing an appropriate budget keeping in mind contingency measures and making a Bill of Materials.
- Cloud architecture has been studied upon and two open-source platforms (ThingSpeak and Blynk) have been identified and tested as the host.
- Sensors and microcontroller were procured, and a prototype is built. Required code is developed to integrate the sensors with microcontroller and successfully tested
- The remote server is configured and successfully connected with the system.
- All the data field can be observed in real time visually in remote server and in mobile application also.
- Whenever any parameter goes above specified value, an alarm is triggered in transformer end and remote server is notified
- An algorithm is made for automatic control of cooling fan based on temperature and loading data as well as predictive analysis of past information.
- Separate provision is made for remote control of cooling fans. For this another cloud platform "Blynk" is used, as "ThingSpeak" does not allow remote command operation.
- Thus, we have effectively made a data acquisition system which help us in-
    1. Predictive analysis
    2. Detecting transformer failure
    3. Transformer life assessment and many more applications
- In addition, remote operation of cooling fans is also achieved.
- Thus, we propose that if implemented in large scale, the system can bring more reliability and efficiency in power distribution system by optimised operation of transformer.

XI. Constraints and Shortcomings

1. In this remote condition monitoring of transformer model, we have excluded some critical parameters such as partial discharge, Dissolved gas in oil, etc which are either beyond the scope or may not be possible to monitor remotely and effectively.

2. The model is based on Wi-Fi technology and it is pre-assumed that Wi-Fi connection is available. Although It can be modified in accordance with GSM technology also. Nevertheless, whatever be the connection type, a stable connection with the cloud server should be maintained, which may be difficult in some remote terrain.
3. Our Model utilises the ThingSpeak / Blynk server, which are free IoT server, and it has a constraint on how many data field can be utilised. It is expected for commercial implementation of such model Utilities should have their own server.
4. As the transformer is effectively connected to the internet, via which it is sending the data to remote server, possible cyber security implication should be carefully understood.

XII. Future Aspects:

1. Hardware and Software Improvements:
Though we have made our system capable and efficient to monitor the transformer conditions, we can improve its efficiency and reliability with better hardware components like sensors and microcontrollers. With sophisticated sensors (subject to future market availability) we can monitor the parameters more precisely and cost effectively.
Moreover, several distribution transformers in a certain area can be merged and taken under our system with such capable controllers.
Moving to the software part, we are currently using free cloud server utilities like ThingSpeak/Blynk server which constraints us to the certain fields for monitoring. But an exclusive server made for such system will help more in desired data consumption, remote controlling and analysing. The concept will help in smart grid technology also.

2. Inclusion of more parameters:
As evident we are measuring most of the basic and crucial parameters. But there are also some other parameters considering transformer health, which are left out due to unavailability of necessary sensors and economic constraints. These are Dissolved gases in oil, humming noise and vibration, Transformer oil impurities, oil flow rate, bushing condition. With proper technology and sensors, these can be also included to give a clearer picture about transformer health.

3. Load Prioritization:
From the data acquired from our system, a protection system can be designed to separate loads based on priority in case of minor fault in transformer happens. Disconnect the less important load (workshops, residence etc) and it will keep the loads of high importance (hospitals, substations etc.). If the transformer available rating is unable to run loads of high importance, in this case system will separates all loads and stay in the no-load status where the monitoring system monitor its parameters by itself, if all major parameters of the transformer return to the desired normal level, the monitoring system will automatically reconnect the loads in priority order.

4. Addition of GSM technology:
As we are using a Wi-Fi based wireless communication system, so we need a stable Wi-Fi connection in both ends. Though here we presumed that the Wi-Fi connections

are available, but in remote areas it is difficult to find a stable Wi-Fi connection on the distribution transformer end. So, we can integrate GSM (Global Service Mobile) technology and modify our system accordingly to overcome such situations.

XIII. References:

1. Vivek Landage, Ashish Khsrche, Vadirajacharya. K, Harish Kulkarni, 'Transformer health condition monitoring through GSM Technology", International Journal of Scientific &Engineering. Research Volume 3, Issue 12, December-2012,
2. https://electrical-engineering-portal.com/download-center/books-and-guides/electricity-generation-t-d/intelligent-transformer
3. PREMATURE FAILURE OF DISTRIBUTION TRANSFORMERS –A CASE STUDY- Ranjana Singh, A S Zadgaonkar, Amarjit Singh, International Journal of Scientific & Engineering Research, Volume 5, Issue 6, June-2014, ISSN 2229-5518
4. https://www.drykeep.com/how-moisture-reduces-transformer-life/
5. https://electrical-engineering-portal.com/transformer-temperature-oil-level-and-pressure-gauges#:~:text=Maintaining%20the%20proper%20oil%20level,and%20the%20transformer%20will%20overheat
6. https://ieeexplore.ieee.org/document/8534849
7. https://ieeexplore.ieee.org/document/8897325
8. https://ieeexplore.ieee.org/document/8521065
9. http://electrical-engineer-guide.blogspot.com/2012/10/transformer-oil-and-winding-temperature.html
10. http://www.utilityscanir.com/papers/Oil_Level_Power12.pdf
11. https://www.usbr.gov/tsc/techreferences/mands/mands-pdfs/Trnsfrmr.pdf
12. https://forum.arduino.cc/index.php?action=dlattach;topic=480767.0;attach=220845
13. https://datasheets.maximintegrated.com/en/ds/MAX6675.pdf
14. https://thingspeak.com/pages/learn_more?Education
15. https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf
16. https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf
17. https://in.mathworks.com/help/thingspeak/
18. https://en.wikipedia.org/wiki/Transformer_oil
19. https//bu.edu.in
20. https://www.researchgate.net
21. https://www.itma.org.in/uploads/1546411477_6824_-_Copy.docx
22. https://www.reliableplant.com/
23. https://ieeexplore.ieee.org/document/369583