

# Mini Scada System

Prof. Quereshi.A.A<sup>1</sup>, Bhosale Shivani Vajinath<sup>2</sup>, Birajdar Anisha Shivling<sup>3</sup>,  
Shinde Shraddha Saudagar<sup>4</sup>, Muskan M.D Raju<sup>5</sup>

<sup>1, 2, 3, 4, 5</sup> Dept of Electrical Engineering,

<sup>1, 2, 3, 4, 5</sup> Vishweshwarayya Institute Of Engineering And Technology, Dilip Nagar, Almala Tq. Ausa Dist. Latur - 413556,

**Abstract-** Supervisory Control and Data Acquisition (SCADA) systems are widely used in industries for monitoring and controlling various processes in real time. However, traditional SCADA systems are expensive and complex, making them unsuitable for small-scale applications and educational purposes. This project presents the design and development of a Mini SCADA System using ESP32 and IoT technology to monitor environmental parameters, electrical parameters, and control devices remotely through a mobile application.

The proposed system is built using an ESP32 microcontroller, which acts as the central processing unit. Various sensors are integrated with the system to collect real-time data. A DHT11 sensor is used for measuring temperature and humidity, an MQ135 gas sensor detects air pollution levels, a soil moisture sensor monitors soil water content, and an ultrasonic sensor measures water level. Additionally, an INA219 sensor is used to monitor electrical parameters such as voltage, current, and power consumption.

**Keywords:** Component, formatting, style, styling, insert.

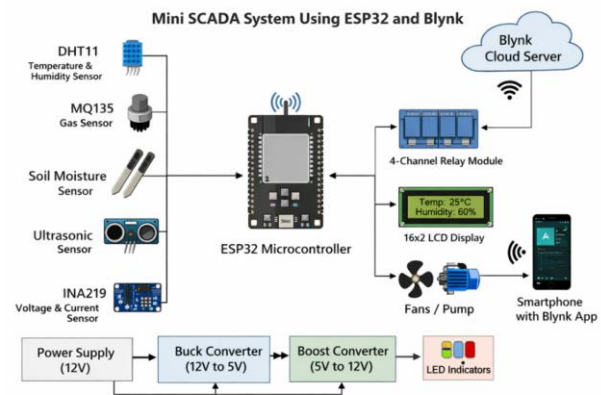
## I. INTRODUCTION

SCADA stands for Supervisory Control and Data Acquisition. It is a system used for monitoring, controlling, and analyzing industrial processes in real time. SCADA systems are commonly used in industries such as power plants, water treatment plants, oil and gas industries, manufacturing industries, and smart agriculture systems.

A typical SCADA system collects data from different sensors and devices installed in a plant or system. The collected data is sent to a central controller or computer where it is processed and displayed to the operator. The operator can monitor system conditions and control devices remotely if needed.

Traditional SCADA systems consist of components such as Programmable Logic Controllers (PLC), Remote Terminal Units (RTU), Human Machine Interface (HMI), sensors, actuators, and communication networks. These systems are very powerful but they are also very expensive and complex, which makes them difficult to implement in small industries, laboratories, and educational projects.

## II. RESEARCH METHODOLOGY



## III. METHODOLOGY

The Mini SCADA System is designed to monitor environmental conditions and electrical parameters and control devices automatically or remotely using IoT technology. The system operates using the ESP32 microcontroller, which acts as the central control unit. All sensors, output devices, and communication modules are connected to the ESP32.

The working of the system begins when the 12V 2A power adapter supplies power to the circuit. The voltage is then regulated using buck and boost converters to provide suitable voltage levels for the ESP32, sensors, and other electronic components.

Various sensors are connected to the ESP32 to measure different parameters. The DHT11 sensor measures temperature and humidity from the surrounding environment. The MQ135 gas sensor detects air quality or gas concentration in the environment. The soil moisture sensor measures the water content in the soil. The ultrasonic sensor measures the water level in a tank by calculating the distance between the sensor and the water surface. The INA219 sensor monitors electrical parameters such as voltage, current, and power consumption.

All these sensors continuously send data to the ESP32. The ESP32 reads the sensor values and processes the

data according to the programmed logic. The processed data is then displayed locally on a 16×2 LCD display for real-time monitoring.

#### IV. LITERATURE REVIEW

A literature review is an important part of any engineering project because it provides information about **previous research, technologies, and methods** used by other researchers. By studying earlier work, researchers can understand existing solutions, identify limitations, and design improved systems.

SCADA (Supervisory Control and Data Acquisition) systems have been widely used in industries for **monitoring, controlling, and collecting data from remote devices**. Traditional SCADA systems rely on **PLC controllers, RTUs, communication networks, and central control software**, which makes them expensive and difficult to implement in small-scale systems.

With the advancement of **Internet of Things (IoT)** technology, modern SCADA systems are becoming more flexible, cost-effective, and accessible. IoT devices such as **ESP32, Arduino, and Raspberry Pi** can now be used to create low-cost monitoring and control systems with remote access through cloud platforms and mobile applications.

This chapter reviews several research works and technological developments related to **IoT-based SCADA systems, remote monitoring systems, and ESP32-based automation systems**.

#### List of Component's :-

- ESP32 Microcontroller
- INA219 Voltage and Current Sensor
- DHT11 Temperature and Humidity Sensor
- MQ135 Gas Sensor
- Soil Moisture Sensor
- Ultrasonic Sensor
- Buck Converter
- Boost Converter
- LED Indicators
- CPU Fans

#### Result and Discussion :-

The Mini SCADA System using ESP32 and IoT technology was successfully designed and implemented. The system was tested using different sensors and hardware

components, and it was able to monitor and control multiple parameters effectively.

During the testing of the system, the sensors such as **DHT11, MQ135, soil moisture sensor, ultrasonic sensor, and INA219** were able to measure environmental and electrical parameters accurately. The ESP32 microcontroller successfully collected data from all the sensors and processed the information according to the programmed logic.

The measured data such as **temperature, humidity, gas level, soil moisture, water level, voltage, current, and power** were displayed on the **16×2 LCD display** and also transmitted to the **Blynk mobile application** through Wi-Fi. This allowed the user to monitor system parameters in real time using a smartphone.

#### V. CONCLUSION

The relay module was also tested successfully. The system was able to control external devices such as fans and pumps automatically based on sensor readings. For example, when the gas level increased beyond the predefined limit, the ventilation fan turned ON automatically. Similarly, when the soil moisture level became low, the irrigation system was activated.

The system also generated alert notifications on the Blynk application when abnormal conditions occurred, such as high temperature, high gas concentration, or low water level.

Overall, the Mini SCADA system performed successfully and demonstrated reliable real-time monitoring, automatic

#### Software requirement

Software plays a very important role in the development of the **Mini SCADA System**. The software is responsible for controlling the microcontroller, processing sensor data, managing communication with the IoT platform, and controlling output devices such as relays and fans.

In this project, the ESP32 microcontroller is programmed using **Arduino IDE**, and the data communication and remote monitoring are performed using the **Blynk IoT platform**. The program written for the ESP32 reads data from sensors, processes the information, displays data on the LCD screen, and sends the data to the Blynk mobile application through Wi-Fi.

The software also implements the **automation logic**, which controls relays automatically when certain conditions

such as high temperature, high gas levels, low soil moisture, or abnormal voltage are detected.

Thus, the software integrates **sensor data acquisition, data processing, communication, automation control, and remote monitoring** to make the Mini SCADA system fully functional.

#### Code :

```
#define BLYNK_TEMPLATE_ID
"TMPL3rILmcvGh"
#define BLYNK_TEMPLATE_NAME "MINI
SCADA SYSTEM"
#define BLYNK_AUTH_TOKEN
"07Wefa7ALgWbFwRYXTGNBYK-9FB4EL3k"

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <LiquidCrystal_I2C.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Arduino@21";
char pass[] = "00000000";

#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

Adafruit_INA219 ina219;
LiquidCrystal_I2C lcd(0x27,16,2);

#define MQ 34
#define SOIL 35
#define TRIG 12
#define ECHO 14

// LOW trigger relays
#define R1 16
#define R2 17
#define R3 18
#define R4 19

// Relay states
bool r1=false, r2=false, r3=false, r4=false;

// Auto/manual flags
bool auto1=true, auto2=true, auto3=true, auto4=true;
```

```
// Alert flags (avoid spam)
bool tempAlert=false;
bool gasAlert=false;
bool soilAlert=false;
bool waterAlert=false;
// Limits
float tempLimit=35;
int pollutionLimit=1500;
float voltMax=14;
float voltMin=10;
float currMax=3;

int screen=0;
unsigned long timer=0;

void setup(){
Serial.begin(115200);
Blynk.begin(auth,ssid,pass);

dht.begin();
ina219.begin();
lcd.begin();
lcd.backlight();

pinMode(R1,OUTPUT);
pinMode(R2,OUTPUT);
pinMode(R3,OUTPUT);
pinMode(R4,OUTPUT);

// OFF initially
digitalWrite(R1,HIGH);
digitalWrite(R2,HIGH);
digitalWrite(R3,HIGH);
digitalWrite(R4,HIGH);

pinMode(TRIG,OUTPUT);
pinMode(ECHO,INPUT);
}

void loop(){
Blynk.run();
readSensors();
lcdScreens();
}

//////// BLYNK MANUAL CONTROL //////////

BLYNK_WRITE(V10){ // Relay1 manual
auto1=false;
r1=param.asInt();
updateRelays();
```

```

}

BLYNK_WRITE(V11){ // Relay2 manual
auto2=false;
r2=param.asInt();
updateRelays();
}

BLYNK_WRITE(V12){ // Relay3 manual
auto3=false;
r3=param.asInt();
updateRelays();
}

BLYNK_WRITE(V13){ // Relay4 manual
auto4=false;
r4=param.asInt();
updateRelays();
}

//////// AUTO MODE SWITCH //////////

BLYNK_WRITE(V20){ auto1=param.asInt(); }
BLYNK_WRITE(V21){ auto2=param.asInt(); }
BLYNK_WRITE(V22){ auto3=param.asInt(); }
BLYNK_WRITE(V23){ auto4=param.asInt(); }

////////////////////////////////////

void readSensors(){

float t=dht.readTemperature();
float h=dht.readHumidity();
int gas=analogRead(MQ);
int soil=analogRead(SOIL);

float v=ina219.getBusVoltage_V();
float c=ina219.getCurrent_mA()/1000;
float p=v*c;

int dist=waterLevel();

// ===== ADD BELOW =====
if(dist> 10){
  if(!waterAlert){
    Blynk.logEvent("water_low");
    waterAlert = true;
  }
}else{
  waterAlert = false;
}
// =====

}

// Blynk monitoring
Blynk.virtualWrite(V0,t);
Blynk.virtualWrite(V1,h);
Blynk.virtualWrite(V2,v);
Blynk.virtualWrite(V3,c);
Blynk.virtualWrite(V4,p);
Blynk.virtualWrite(V5,gas);
Blynk.virtualWrite(V6,soil);
Blynk.virtualWrite(V7,dist);

///// AUTOMATION //////////

// GAS CONTROL + ALERT
if(auto1){
if(gas >pollutionLimit){
r1 = true;

if(!gasAlert){
Blynk.logEvent("gas_high");
gasAlert = true;
}
}else{
r1 = false;
gasAlert = false;
}
}

// TEMPERATURE CONTROL + ALERT
if(auto2){
if(t >tempLimit){
r2 = true;

if(!tempAlert){
Blynk.logEvent("temp_high");
tempAlert = true;
}
}else{
r2 = false;
tempAlert = false;
}
}

// LOAD PROTECTION (already exists)
if(auto3){
if(v>voltMax || v<voltMin || c>currMax){
r3=false;
Blynk.logEvent("power_fault");
}else{
r3=true;
}
}
}

```

```

// SOIL IRRIGATION + ALERT
if(auto4){
if(soil > 3000){
r4 = true;

if(!soilAlert){
Blynk.logEvent("soil_dry");
soilAlert = true;
}
}else{
r4 = false;
soilAlert = false;
}
}

updateRelays();
}

void updateRelays(){
digitalWrite(R1, r1?LOW:HIGH);
digitalWrite(R2, r2?LOW:HIGH);
digitalWrite(R3, r3?LOW:HIGH);
digitalWrite(R4, r4?LOW:HIGH);
}

int waterLevel(){
digitalWrite(TRIG,LOW);
delayMicroseconds(2);
digitalWrite(TRIG,HIGH);
delayMicroseconds(10);
digitalWrite(TRIG,LOW);
long d=pulseIn(ECHO,HIGH);
return d*0.034/2;
}

void lcdScreens(){

if(millis()-timer>5000){
screen++;
if(screen>2) screen=0;
timer=millis();
lcd.clear();
}

if(screen==0){
lcd.setCursor(0,0);
lcd.print("Temp:");
lcd.print(dht.readTemperature());
lcd.setCursor(0,1);
lcd.print("Gas:");
lcd.print(analogRead(MQ));

```

```

}

if(screen==1){
lcd.setCursor(0,0);
lcd.print("Volt:");
lcd.print(ina219.getBusVoltage_V());
lcd.setCursor(0,1);
lcd.print("Curr:");
lcd.print(ina219.getCurrent_mA()/1000);
}

if(screen==2){
lcd.setCursor(0,0);
lcd.print("Soil:");
lcd.print(analogRead(SOIL));
lcd.setCursor(0,1);
lcd.print("Water:");
lcd.print(waterLevel());
}
}

```

#### Advantages :-

- Low Cost System
- Real-Time Monitoring
- Remote Monitoring
- Automatic Control
- Wireless Communication
- Multiple Parameter Monitoring
- Energy Monitoring Capability9

#### Disadvantages:-

- Dependence on Internet Connection
- Limited Industrial Capability
- Sensor Accuracy Limitations
- Limited Processing Power
- Security Concerns
- Environmental Sensitivity
- Maintenance Requirement
- Installation Challenges

#### Applcation :-

- Smart Agriculture
- Industrial Monitoring Systems
- Environmental Monitoring
- Water Tank Monitoring
- Energy Monitoring Systems