Automatic Control of Home Appliances Based on SCADA and IoT System

By

Riffat Ahmed
15321017
Khandaker Towfiqul Islam Ananta
15321007
Mohammed Fardin Anwar
15221007
Samanta Afrin
13221022

A thesis submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical & Electronic Engineering

Department of Electrical and Electronic Engineering
Brac University
June 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our original work while completing the degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material that has been accepted or submitted, for any other degree or diploma at a university or other institution.
4. I/We have acknowledged all main sources of help.

Student's Full Name & Signature:

Riffat Ahmed
15321017

Samanta Afrin
13221022

Khandaker Towfiqul Islam Ananta
15321007

Mohammed Fardin Anwar
15221007

Approval

The thesis/project titled "Automatic Control of Home Appliances Based on SCADA and IoT System" submitted by

1. Riffat Ahmed (15321017)
2. Samanta Afrin (13221022)
3. Mohammed Fardin Anwar (15221007)
4. Khandaker Towfiqul Islam Ananta (15321007)

Of Spring, 2021 has been accepted as satisfactory in partial fulfilment of the requirement for the degree of B.Sc. in Electrical and Electronic Engineering on 10 June 2021.

Examining Committee:

Supervisor:
(Member)

____

A. S. Nazmul Huda PhD
Assistant Professor,
Department of Electrical and Electronic Engineering
Brac University

Thesis Coordinator:
(Member)

____

Abu S. M. Mohsin PhD
Assistant Professor
Department of Electrical and Electronic Engineering
Brac University

Departmental Head:
(Chair)

____

Md. Mosaddequr Rahman PhD
Chairperson
Department of Electrical and Electronic Engineering
Brac University

Abstract

This Project aims to use a micro-scale SCADA system to control various electronic appliances connected to the grid through a remote medium. The system will use the internet to communicate from a remote location. This system can help reduce electricity waste in a case where a person might forget to turn off a light or fan when going out for vacation. The novelty lies in the aspect of remote-control features with integration of the internet, as the internet is becoming an integral part of our life. It can help us control electrical devices in our home with help from a smartphone, PC, laptop, etc.

Keywords: IoT; ESP32; Automation; SCADA; Grid Control.

Acknowledgement

First of all, we would like to thank Dr. A.S. Nazmul Huda sir for his immense support and advice in our work. This project would not have been possible without his guidance and supervision.

Lastly, we would like to thank the family members of all the members of this theses team, for their support and prayer for us throughout this research period.

Table of Contents

List of Figures

List of Acronyms

EEE	Electrical and Electronic Engineering
GPIO	General Port of Input/Output
HMI	Human Machine Interfaces
HTML	Hypertext Markup Language
IoT	Internet of Things
LAN	Local Area Network
OLE	Object Linking and Embedding
OPC	OLE for Process Control
PCB	Printed Circuit Board
PLC	Programmable Logic Controllers
ROM	Read Only Memory
RTC	Remote Time Clock
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Accusation
SRAM	Static Random-Access Memory
TSMC	Taiwanese Semiconductor Manufacturing Company
WAN	Wide Area Network

Chapter 1

Introduction

1.1 Introduction

No one could have predicted our current state two decades ago. Heavy and cuboid mobile phones gradually evolved into sleeker, slimmer forms. Technology advancements have raised our spirits to the point that we can now work with silicon chips that are nanometer-sized. Not only have these processors improved gadget performance, but they've also enabled a slew of new features, such as a touch-sensitive keypad in place of traditional buttons. Smarter applications, such as the use of gyroscopes and motion sensors, are also at the forefront of the technical advances we see on our phones. Aside from the changes that have occurred with our mobile devices, PCs have also undergone significant changes. We've gone a long way from massive cubical designs capable of supporting only a few bits of data to thinner displays with greater processing rates. Our way of life has been completely transformed by technological advancements. We can only speculate on the astounding and futuristic alterations that the future will bring. On that point, let us turn our attention to a new and impending technology that will not only improve our lives, but that once implemented, we may never look back.

Remote Connectivity is the amazing “next” innovative step towards a fully automatized future which as of the moment seems just science fiction for us. Being able to unlock the door when you are near your house’s gates and vice versa when leaving, switching the lights off (in case you have forgotten) while being in the office, switching on the heater or the air conditioner depending upon the temperature of your room before you arrive from your work, having a meal cooked just when you set your destination towards home, automatically managing to water the plants in your garden upon monitoring the water level of the soil and humidity of the air, etc.

are just the tip of the iceberg for this new technology. Not only are we becoming more efficient in our lifestyle, but we are also relieving ourselves of mundane daily chores and having more time to boost our intellectual prowess. Smart monitoring and switching off the appliances and other electrical utilities from the comforts of our smartphone will not only reduce the toll on our electricity bill but on the power plants that generate them. Wasting fewer electricity results in lesser to be provided than the current amount. This ends up burning fewer fossil fuels and ultimately reducing the number of greenhouse gases that we are emitting into the atmosphere. This is the beneficial cycle created by the snowball effect from just initiating a minor change in our lifestyle. Therefore, remote connectivity is not only futuristic and trendy but is just the initial point for us to truly solve the global warming issue. This is why this project is important and we should truly explore the possibilities that has the ability to touch and transform lives for the better.

Creativity is the ability we possess to imagine and invent new possibilities. However, innovation is when we use that creativity and apply it to the existing technology. Similarly, just dreaming about the future of remote connectivity without attempting to incorporate it with our existing knowledge and technological capabilities will surely not bear any result. Hence, we inspect a few key contenders that do possess some unique features that are very favorable to our demands. The first one is SCADA and the other one is IoT. SCADA has various features which include the ability to store vast amounts of data, portraying that data in the format the user requires, obtaining fast responses and real-time data simulations [1]. However, the crucial factors that work in favor of SCADA are the monitoring capabilities and the ability to control everything from a master control unit. The monitoring capabilities houses over a thousand sensors spread widely across various RTUs (Remote Terminal Unit) and the master control unit gives the power to operate everything from a single place. Additionally, it is very flexible and scalable which is why it is widely used in the telecommunication, energy, transportation, military, meteorological, oil and gas and water industries. In contrast, SCADA is extremely expensive as the initial cost is rather high and it is dependent upon the existing software and hardware of the PLC. Another factor that is working against it is that the unemployment rate will rise due to it and job requirements will get much tougher since skilled operators, analysts and programmers will be required to operate the system [2]. Next, the concept of IoT opens up various opportunities in the field of home security systems where the user can locally and/or remotely control and observe via the ease of their smartphones. The IoT network is also very instrumental for anyone who is using it. It boosts the quality of life for the individual, stakeholders and society. But alas, it also possesses some grave disadvantages as well. The IoT network consists of various technologies employing even more diverse architectures. Thus, it makes the entire system very complex and any failure in the system will be very hazardous. It will require a great amount of time for a skilled worker to fix the system and get it back running again. IoT also increases the chance of the unemployment rate rise since the workers will need to keep on learning new technologies each time the system gets upgraded. As for the unskilled workers that cannot cope with the new software and hardware knowledge will surely face a job crisis. Also, as multiple vendors are involved, privacy is a huge concern. Furthermore, multiple vendors also incur extra costs for the manufacturers and service providers during launch since the system's interoperability needs to be checked properly [3].

Hence, as we noticed in the previous paragraph that both SCADA systems and IoT possess some very lucrative features that shall aid us in our goal to achieve remote connectivity. However, the systems alone cannot boast perfection. Thus, the reason why we aim to build this project that utilizes the best of both while reducing their negatives.

1.2 SCADA

1.2.1 Introduction to SCADA

SCADA is an acronym that stands for Supervisory Control and Data Acquisition [4]. It is a set of both hardware and software components that aid in the supervision and control of equipment in industries such as telecommunications, water and waste management, electricity, oil and gas processing, and transportation - primarily systems with a large geographical footprint. The SCADA system does not have any predetermined protocol or standard. It is a software-only package that sits on top of hardware with which it communicates, usually through Programmable Logic Controllers (PLCs) or other commercial hardware modules. As aforementioned, the features which are available from SCADA are the local or remote control of industrial processes, real-time data acquisition and observation, a Human Machine Interface that allows the user to interact directly with valves, motors and sensors. The users operating the Master Control Unit can make quick decisions to all the Remote Terminal Units without the hassle of understanding the complex mechanisms of the machine-to-machine interaction.

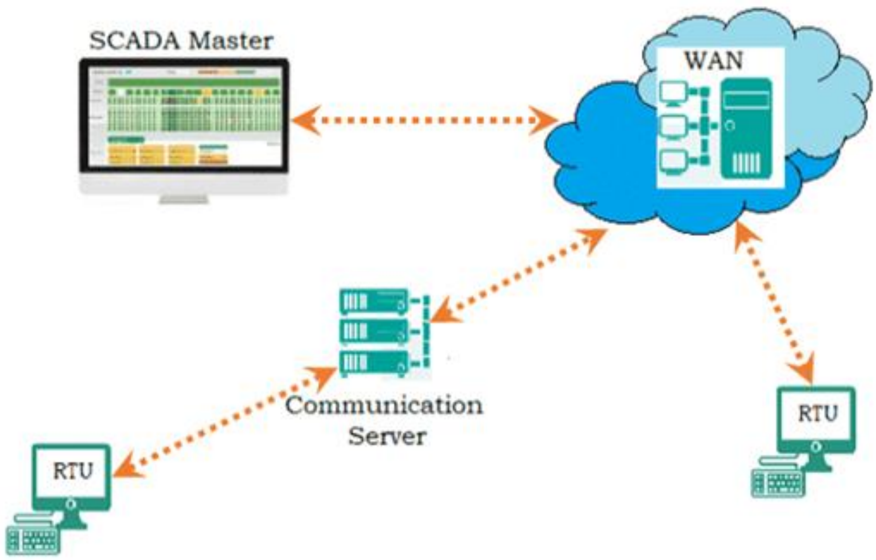

Figure 1: SCADA System [5]

Not only does it reduce downtime, it also boosts the efficiency and overall quality of work done [6].

1.2.2 History of SCADA

Electric systems were used to power industrial processes in the early 1960s. The term SCADA was coined in the mid-1970s to describe automatic control and data acquisition. Before the 1950s, every other process relied heavily on manpower. It was not until the end of the 1950s that SCADA was first conceptualized. The very first form of SCADA was telemetry SCADA relying on telephone wires. It was after the invention of transistors in the late 1960s that gave rise to minicomputers and microprocessors which allowed vendors to make more efficient and cost-effective PLC. These made SCADA become real-time and possess boosted efficiency and reliability [7].

Generation I: Monolithic SCADA: The manufacturers were swift to grasp the benefits which SCADA introduced and started to build applications of process control and sold them to specific industries. Hence, the rise of the first architecture of SCADA was introduced. Since networks were non-existent, the control systems had no means to interconnect and thus acted as standalone systems. Despite the interconnectivity, they were still able to form WAN (Wide Area Network). However, it wasn't as advanced as our modern models and was only able to scan, control and exchange data from different RTUs (Remote Terminal Units) to the master computer [8].

Generation II: Distributed SCADA: After the wide acceptance of the initial model, the suppliers knew that they must introduce a more efficient and improved model for the new booming industry. This phenomenon led to the evolution of the SCADA model. The core components of this generation were system miniaturization and LAN (Local Area Network).

Other key components such as communication processors, HMI (Human Machine Interfaces), RTUs and databases were all part of the distributed SCADA which ultimately made it much more reliable and cost-effective.

However, a few shortcomings of this generation aside from sharing data with multiple other stations, it was only limited to the systems connected in that network. Also, the suppliers played a major role when providing the software, hardware and other protocols.

Generation III: Networked SCADA: SCADA evolved into the third generation and the current architecture model after industries adapted and opened the doors to an open system architecture. It is very similar to the second generation. The only drastic difference is the usage of WAN (Wide Area Network) and various protocols like IP (Internet Protocol). This opened up the ability to control all other stations just from the master station via an Ethernet connection. During 1996, a standard system known as OPC (Object Linking and Embedding (OLE) for Process Control) Foundation was created to set a standard for the various suppliers of SCADA to follow as the years passed [9].

But alas, great advancement in technology increases the danger a few folds as well. Being connected to an IP means without a proper security system, anyone can hack into the system and alter files or steal data from the servers. If it was classified data that may harm the company, the hackers then pose an even greater risk. For such reasons, having the practice of strong and efficient internet security is a necessity [10].

1.2.3 Architecture of SCADA

SCADA architecture is the combination of bother software and hardware components. The hardware segment consists of the RTU, MTU, sensors and actuators [11]. Meanwhile, the software components include HMI, Historian, PLC and more software that makes the interaction and communication between the hardware and the software smoother. The sensors and actuators are assigned to collect data which using their communication link pass to the RTUs which in turn feeds off to the MTU. The user can now freely observe and control the entire SCADA framework based on the information provided to him thus enabling faster and more accurate responses [12]. Now a detailed explanation of each component follows.

RTU – Remote Terminal Units are assigned to show the current status of the machine they are connected with and pass any data and information to the MTU from the sensors and actuators via the communication link i.e., LAN/WAN [13].

MTU – Master Terminal Unit can be regarded as the central hub or the brain of the system. It receives all the information from the RTU regarding the status of the connected machines. Here onwards, it can perform the decisions regarding the best course of action that is required. If any immediate problems require special attention, the MTU can command the RTU to halt that machine so an operator can work on it. If the situation is not of dire repercussion, then the data which arrived via the communication link can just be stored for later usage [13].

HMI – Human Machine Interface is the bridge that connects both the hardware and software components of the SCADA system. It provides the interface for the user to understand the data obtained from the RTU and MTU in the form of statistical and graphical models or as plain texts. The HMI is responsible for transforming the obtained data allowing the user to have full control over the SCADA system [13].

Historian – The Historian's main task is to store all the data acquired from both the RTU and the MTU. It stores all the data in a remote location thus is dubbed as the central database of the system. The historian also has another function which is to provide the HMI with the graphical models of the acquired data.

Communication Network – It is the main network utilized by the SCADA framework to send data and commands to and from the MTU, RTU and the historian. The network model can be either wired or wireless. However, recently wireless model is favored due to its effortless reach [14].

1.3 IoT.

1.3.1 Introduction to IoT.

IoT is an acronym that stands for “Internet of Things”, a term coined by Kevin Ashton. It is a system of interconnected computing devices, mechanical and digital machinery, items, animals, and/or people having unique identifiers (UIDs). It can communicate data over a network without requiring human-to-human or human-to-computer interaction. ORACLE, the open-source software giant, describes IoT as a “network of physical objects— “things”—that are embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet”. The range of applications is massive since it includes simple household objects to industrial machineries. They strongly believe the IoT industry to boom into a staggering 22 billion growth rate by the year 2025. [15]

1.3.2 IoT Architecture

IoT architecture can be divided into four very distinct layers. Each of these layers seamlessly combines its unique properties to produce the interoperability effect. The feeling of being connected to almost anything is the desired outcome stemming from that seamless operation.

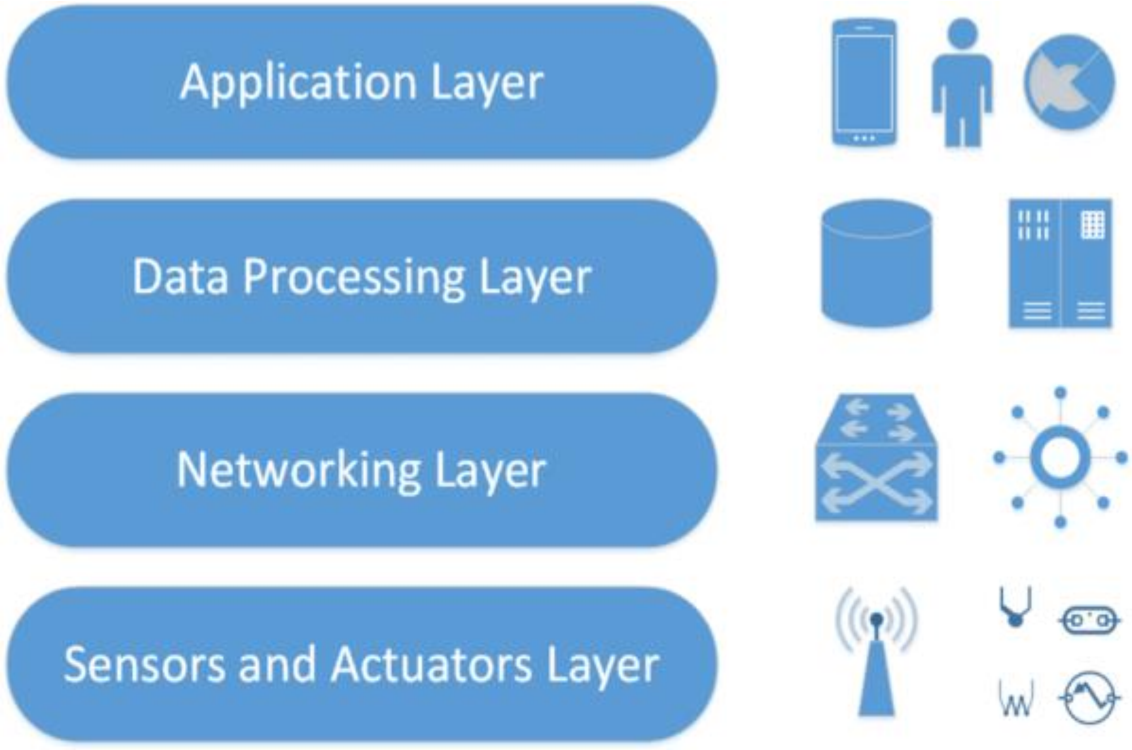

Figure 2:IoT Architecture [16]

i. Communication Layer

The very first layer comprises IoT Devices such as sensors and actuators. Their core functions include sensing, computing and connecting other devices. Sensors are responsible for picking up real-time telemetric data from the physical environment such as temperature, soil pH, etc. Meanwhile, actuators can control and rectify any immediate required changes such as turning off a switch, controlling the water flow rate, etc. This is the main layer that utilizes Bluetooth to connect mobile phones, Zigbee to connect Zigbee gateways and Raspberry Pi to directly connect the communication layer via Wi-Fi or Ethernet. The communication layer includes the REST, HTTP, MQTT and other application protocols [16].

ii. Networking Layer

The second layer also known as the Bus Layer aggregates all the data that the sensors provide. It bridges the sensors' data to the cloud and converts them to their designated protocols. The layer is important for three different reasons namely: it supports the HTTP server and MQTT broker, it aggregates and combines information via Gateway and it bridges entrance form data between different protocols [16].

iii. Data Processing Layer

This is the third layer which is very crucial for connecting multiple devices over the Web. The cloud stores and processes all the data in the database which are received from the sensors. It drives the data and provides transformation for the generated data [16].

iv. Application Layer

The last layer is the application layer which bridges the infrastructure to the third-party application. It creates web-based engines to interact with external APIs which are then fed into the API management systems. This layer creates a dashboard and helps view the analytics event processing. Aside from that, this layer also helps communicate the systems outside the network using Machine to Machine (M2M) communication [16].

1.3.3 IoT Characteristics

After exploring the architecture of IoT, let us proceed to the characteristics. Among the various characteristics IoT possess, these five stands at the very core. These are as follows:

- Dynamic
- Self-configuring

- Interoperable communication protocol
- Unique Identity
- Integrated into the information network

i. Dynamic – They are incredibly fast to adapt to any changes in the environment, conditions or user's command and respond accordingly to it.
ii. Self-configuring - They can upgrade on their own without the need for human intervention. They can also easily add new devices to the existing network on their own.
iii. Interoperable communication protocol – They allow upcoming and brand-new architectures and networks to seamlessly connect with the existing ones.
iv. Unique Identity – The connected devices of the IoT network possess a unique identifier that allows users to have control from a remote area.
v. Integrated into the information network – The devices which are connected to the network constantly keep feeding the information network making them smarter and more efficient. Each connected devices have their visibility maintained by their unique SSID. Thus, the information network allows other devices to locate your connected device. Meanwhile, you benefit from this collaboration by receiving valuable information present in the information network which other connected devices have fed in previously [17].

1.4 Project Objective

The main objective of the paper is to highlight the importance of remote connectivity. Due to its immense potential in shaping our future lives, we should care about its growth. Additionally, as aforementioned, the prime contenders we saw fit to utilize for remote connectivity were SCADA and IoT. The SCADA design has the tools, stability, and adaptability to maintain multiple systems running concurrently from a single master control unit. What it lacks are security systems that can cause major harm to users if it falls into the wrong hands, such as the security code for the door lock, which is just one of countless issues. Meanwhile, the IoT system is incredibly secure since it has numerous levels of security barriers placed on top of each other. The architecture, on the other hand, lacks interoperability with non-IoT devices. SCADA can bring non-IoT equipment under the radar of IoT interoperability, paving the way for a smarter future. Another important consideration is that SCADA systems are exceedingly costly to construct. IoT solutions, on the other hand, demand far fewer financial resources. In contrast, IoT systems, are quite complicated and need a vast knowledge base in numerous technologies, whereas SCADA systems are relatively straightforward to run provided the user is familiar with PLC software and hardware. Therefore, we shall continue with our project by combining their simplicity of use and adaptability.

1.5 Summary

In the second chapter of the paper, we had our literature review covering the core components and functions of SCADA. Additionally, the focus shifted towards the core function of IoT. Moving onwards, the third chapter introduced a key component of the project model, i.e., ESP32. It also elaborated on the hardware and software components along with the results. The fourth chapter dealt with application and safety whereas the fifth dealt with challenges. Finally, the sixth chapter spoke regarding the conclusion and all that was possible as future work regarding this project.

Chapter 2

Literature Review

2.1 Introduction

We now live in a different world from a few years ago because of COVID-19. In this new world, we find ourselves stuck in our home, not able to go outside without risking our family's health. Thus, the need to supervise and control various equipment is also becoming more prevalent. Most remote monitoring systems are being used in the industrial sector.

One of these remote monitoring systems is the SCADA system. Systems like SCADA can be helpful to achieve this remote connectivity. Manufacturers use the SCADA system in large-scale operations where they need real-time views into the operation. SCADA systems are generally expensive to build and maintain. The SCADA system has limited computing power and is not capable of artificial intelligence. Therefore, the SCADA system is considered to be a dumb system. Modbus, DNP3, Profibus, and Conitel are some of the proprietary languages used by the systems to interact. The data from the system's components are generally recorded and kept on the system. These are usually closed-loop systems managed locally and are often tailored for the industrial processes they regulate [17].

IoT is also a remote controlling system that is more flexible than SCADA. SCADA is used on an industrial scale, IoT can be used in industrial, residential, and consumer scales. SCADA and IoT have many similarities; both involve sensors and data acquisition. The two systems also differ in any way. In the end, the systems have one common goal: the optimization of use and, in the long run, better control over some devices or processes. IoT system developed separately from SCADA. We can trace the root of IoT to Information technology or IT. They started with mainframe computers more than 50 years ago and progressed to today's client/server/router systems. The term "internet" refers to the interconnection of various systems. Transmission Control Protocol/Internet Protocol (TCP/IP) is the standard communication language used on the internet. These platforms operate all-important business software, websites, email, databases, and so on. IT systems have grown exponentially due to general use and now include "cloud computing" systems, which combine many computing devices to improve processing speed and power. Artificial intelligence (AI) breakthroughs enable computers to learn by analyzing massive data, identifying trends, and creating solutions. When objects other than computers connected and interacted on the internet, the Internet of Things (IoT) was born. The most prevalent IoT device is certainly smartphones. Surveillance cameras, doorbells, swimming pool controllers, building thermostats, and alarm systems have become internet gadgets in the past ten years.

SCADA and IoT both have their unique capability. Research communities around them are trying to integrate SCADA and IoT to improve the capabilities of SCADA and IoT. SCADA systems are generally expensive to build and operate. It has high data security, as the system uses its communication channel. The system cannot process and collect a vast amount of data, has data storage limitations. Compare to SCADA, IoT systems are much cheaper to build. Unlike SCADA, IoT systems DATA security is deficient. IoT systems can be easily hacked as the system uses an open internet connection for communication. IoT can be made capable of collecting and processing a vast amount of data efficiently. DATA storage limits on IoT systems are virtually non-existent as the data storage can be increased quickly and is very cost-effective [18].

In this project, we have also tried to do something similar. We have tried to use both the concept and working principle of SCADA and IoT to create a remote monitoring system capable of transforming a non-smart appliance into a smart appliance with some simple modification. Using the core principle of SCADA and IoT, the system can turn the appliance on or off with a specific command. It is also capable of showing appliance is turned on or off in real-time condition

2.2 Components of SCADA:

SCADA systems are made up of various hardware and software components that collect, translate and display information. Data collection devices, such as sensors and relays, data processing devices, such as a PLC or RTU, and data display devices, such as an HMI or monitor, are standard SCADA hardware components. On the other hand, SCADA software analyzes and converts the collected data into useful information distributed to operators [19]. SCADA software is also in charge of carrying out pre-programmed control and alarm functions. SCADA software can also be used for predictive maintenance and machine learning, depending on the system. A full-fledged SCADA system has four below-mentioned parts.

i. **Hardware**: Input and output relays and electrical signals that interact directly with managed elements in remote sites. The SCADA operations begin at this point. These sensors and control relays collect data, but they cannot decipher communication protocols on their own. However, the SCADA system will still require a mechanism to collect this information and present it helpfully. RTUs are used in this situation.

ii. **Remote Telemetry Unit (RTU):** A remote telemetry unit (RTU), sometimes known as a remote terminal unit (RTU), is a device that collects data from sensors and sends commands to control relays. They are minor computerized units that are deployed in the field in certain areas and places.

iii. **Communication network:** It is the network that connects RTUs and SCADA at a remote site

iv. **Human Machine Interface (HMI):** HMI is software that provides human operations information and allows human operators to execute commands.

2.3 Core functions performed by SCADA:

SCADA systems are critical for industrial organizations because they help maintain efficiency, process data to make better decisions, and communicate system issues to reduce downtime. PLCs (programmable logic controllers) or remote terminal units are the foundation of the SCADA architecture (RTUs). PLCs and RTUs are microcomputers that communicate with various objects, including factory machines, HMIs, sensors, and end devices, and then route data from those objects to computers using SCADA software. The SCADA software analyzes, distributes, and displays data, assisting operators and other employees in making important decisions [20]. For instance, the SCADA system can quickly alert an operator if a product batch has a high rate of errors. To determine the cause of the problem, the operator pauses the operation and uses an HMI to view the SCADA system data. When the operator examines the data, he discovers that Machine 4 was not working correctly. The ability of the SCADA system to alert the operator to a problem aids him in resolving it and preventing further product loss. SCADA system does it by performing these four-core functions

i. **Data acquisition:** Here, data is acquired from various sensors. Often, the data require analog to digital conversion [21].

ii. **Data Communications**: The acquired data is sent to an upstream consolidator or supervisor either by itself or in response to a request for data. Analog (T202, POTS) or digital (RS485, TCP/IP) transmission channels are available. In addition to content certification, most SCADA network topologies contain some form of transit validation [21].

iii. **Data Presentation**: Data is collected, structured, and displayed to system operators so they can ensure suitable response and control decisions. For example, the presentation format might range from a statistical display of logged events to a graphical display against a mapping or graphic background [22].

iv. **Control**: When particular operational or configuration modifications are required, and the system allows output, corresponding instructions can be delivered. RTUs and PLCs handle the majority of the control functions [21].

A basic SCADA diagram is given below:

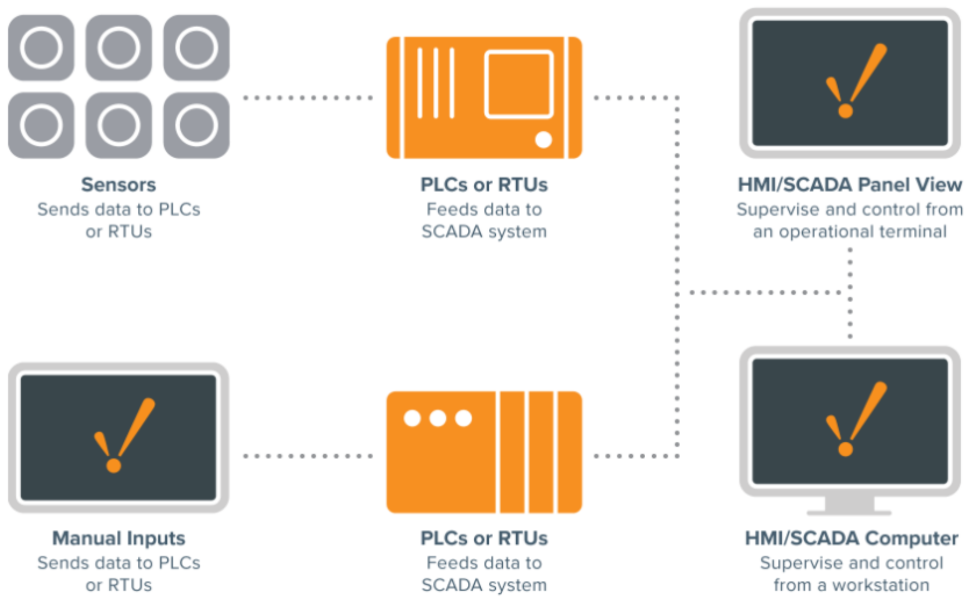

Figure 3:Basic Diagram of SCADA [23]

However, implementing an Industrial SCADA system for household application is not practical. Therefore, various open-source SCADA solutions, each with different costs and functionality, have been developed by research communities worldwide to address this problem.

2.4 Components of IoT

The IoT is a network of interconnected computing devices, mechanical and digital machines, objects, animals, and people with unique identifiers (UIDs) and the ability to transfer data without requiring human-to-human or human-to-computer interaction. As an example, A person with a heart monitor implant, a farm animal with a biochip transponder, an automobile with built-in sensors to alert the driver when tire pressure is low, or any other natural or artificial object that can be assigned an Internet Protocol (IP) address and can transfer data over a network can be considered as things in the internet of things [24]. An IoT ecosystem comprises web-enabled intelligent devices that collect, send, and act on data from their surroundings using embedded systems such as processors, sensors, and communication hardware. IoT devices can share sensor data sent to the cloud for analysis or analyzed locally by connecting to an IoT gateway or other edge device. These devices may occasionally communicate with one another and act on the information they receive. Although people can interact with the devices to set them up, give them instructions, or access data, they do most of the work without human intervention. Four distinct components make up an IoT system. They are:

i. **Sensors/Devices:** Sensors or devices gather information from their surroundings. It could be as simple as a temperature reading or as complex as a full video feed. Sensors can be bundled together or integrated into a device that does more than just sense things. A phone, for example, is a device with multiple sensors (camera, accelerometer, GPS, and many more.), but it is not just a sensor. Whether it is a standalone sensor or a complete device, data is collected from the environment first.

ii. **Connectivity:** Cellular, satellite, Wi-Fi, Bluetooth, low-power wide-area networks (LPWAN), or connecting directly to the internet via ethernet are all options for connecting sensors/devices to the server. Power consumption, range, and bandwidth are all trade-offs in each option. The best connectivity option depends on the IoT application, but they all accomplish the same goal: getting data to the cloud, a massive, interconnected network of powerful servers that provide services to businesses and individuals.

iii. **Data Processing:** Collected data is processed in the cloud. This data processing can be as simple as double-checking that the temperature reading is well within acceptable limits. It could also be highly complicated, such as identifying objects using computer vision on video.

iv. **User Interface:** Here, the processed data is sent and displayed for the user to decide. This could be done by sending a notification to the user (email, text, notification, etc.). For example, if the company's cold storage temperature is too high, a text alert will be sent. A user may also have access to an interface that allows them to check in on the system in advance. A user might, for example, use a phone app or a web browser to check the video feeds in their home. It is not always a one-way street, in any case. The user may be able to perform an action and influence the system depending on the IoT application. For example, the user could use an app on their phone to adjust the temperature in the cold storage. Some tasks are carried out automatically. Instead of waiting for a user command, the system could do it using pre-programmed rules. In addition, rather than simply notifying the user, the IoT system could automatically take appropriate measures [25].

2.5 Difference of SCADA and IoT

The IoT system has played a crucial role in upgrading the SCADA system. Both systems may look similar to each other, but they differ in some key areas. Here is the evaluation of the contrast between each other.

i. **Device interconnectivity:** In the SCADA system, it is not easy to integrate devices from different manufacturers. Furthermore, even when the same company makes the devices, it might be challenging to use them interchangeably since their models differ. They also need horizontal software that can work on various devices, regardless of the manufacturer or model [26].

   The concept of interconnection underpins the Internet of Things. The main goal is to make communication easier across several devices. Nothing is dependent on the appliance's model or manufacturer. It employs protocols like MQTT to provide accessible communication among all components in a system [27].

ii. **Operating expenses and costs:** Companies have to spend expenditures for new servers to store additional data in a SCADA system. SCADA users must also purchase separate licenses for new services and pay for system upgrades regularly when it comes to software licensing or getting extra capabilities [26].

   By utilizing IoT, organizations may significantly reduce their hardware and software expenses while eliminating the need for physical software licensing and upgrades [27].

iii. **DATA Insights:** SCADA systems have flaws in analyzing and interpreting historical data; organizations frequently face these challenges when analyzing and interpreting old data. SCADA does not prioritize collaboration or interpretation of the data that organizations generate regularly. The data does not provide users with any valuable insights or vital assistance to corporate decision-makers [26]

IoT outperforms its opponents in this area. It gathers and consolidates data from various operations and can use big data analytics to forecast efficiency and minimize future swings. Corporations can anticipate unanticipated equipment issues and maintenance requirements because of IoT's unrivaled insight [27]

iv. **SCALABILITY:** There are devices in SCADA software that offer essential information but have never been used because they demand an overhead expenditure. Additionally, when the number of users grows, the bandwidth must grow to accommodate the growth, raising security concerns and mitigating most difficulties. In addition, it takes a long time to set up [26].

IoT allows customers to easily connect and handle massive volumes of data while also connecting new equipment. All of this information is stored in the cloud and accessed using login credentials. An HMI system that links them to the clouds allows them to see and access information or data from anywhere globally [27].

2.6 Summary

The world is slowly adopting remote connectivity and is gradually becoming an essential part of our life. A hybrid system of SCADA and IoT shows a vast possibility for further improvement of the remote-control system. The system can make our everyday life easier. It can also be achieved very cost-effectively.

A SCADA system is expensive, but IoT systems are cheap. IoT can easily integrate with SCADA to make it more reliable. By merging the two systems, we perform the SCADA function and the IoT function very efficiently.

Chapter 3

Project Description

3.1 Project Objective

The objective of this project is to create a prototype of a system for controlling home appliances through SCADA using the ESP32 microcontroller. The goal is to introduce a system that is economic, effective, easy to use, and one that can be used both centrally and remotely. The components used in this project are readily available and extremely inexpensive.

In this system, relays are controlled by the ESP32 microcontroller which is in turn controlled wirelessly via a user interface that can be accessed from anywhere utilizing a Wi-Fi router or mobile hotspot. Alternatively, the microcontroller can also be programmed to be controlled via ethernet, in which case the microcontroller cannot be controlled remotely.

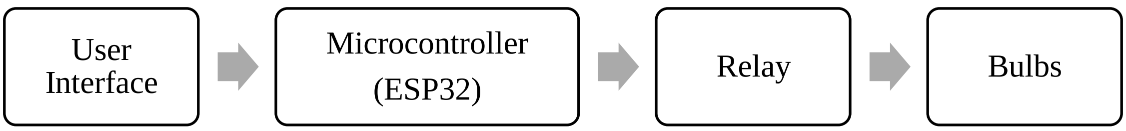

Figure 4:Flow of command of the system.

In this project, the power source is a residential power supply of Bangladesh standard, which is 220 to 240 volts and consequently relays used are the ones that operate in said voltage range. The load is represented by light bulbs. The relays are connected between the source and the loads. When a user sends a command to the microcontroller to switch on or switch off a certain appliance, the corresponding relay breaks or closes the circuit, depending on the command.

When the system is operated via the internet, the operator can be anywhere, as long as they are connected to the internet. Meanwhile, for operation via ethernet, the operator must be connected directly to the ESP32 board.

3.2 Hardware and Circuit

3.2.1 Components

The following hardware components are used in this project:

- Relay Module
- Bulb holder
- Bulbs
- ESP32 Microcontroller
- 4 Relay Module
- 1.3 mm Wires
- Jumper cables
- Wi-fi Router

Apart from the components mentioned above, a PC is needed for coding purposes and also to run the system, if the system is to be controlled remotely and a micro-USB power adapter is required if the project is to be controlled via ethernet. Some other basic components like data cable, plugs, multiplug are also used. Details about some major components and how they are used are discussed later in the chapter, and details about ESP32 is discussed in the previous chapter. All of the components enlisted above can easily be found in electronic and hardware stores in Dhaka.

3.2.2 Hardware Description

Esp32:

Esp32 is a series of low budget “system on a chip” microcontrollers developed by Espressif Systems and manufactured by TSMC (Taiwanese Semiconductor Manufacturing Company) using their 40nm technology. The 2.4GHz microcontroller comes with integrated Wi-Fi and Bluetooth and many other features that make it a powerful tool for IoT projects and such. Esp32 is a successor to ESP8266 and as of yet have four variants:

- ESP32-S2
- ESP32-C3
- ESP32-S3ESP32-C6

Figure 5:A ESP32-WROOM-32 PCB

Various Surface-mount PCBs can be found with integrated ESP32 SoC. ESP32-WROOM-32 is one such PCB, which we will be using for this project. ESP32-WROOM-32 can also be found under the name ESP-WROOM-32

CPU and Memory specifications of ESP32 are as follows:

- Xtensa® single-/dual-core 32-bit LX6 microprocessor(s), up to 600 MIPS

- 448 KB ROM
- 520 KB SRAM
- 16 KB SRAM in RTC
- QSPI supports multiple flash/SRAM chips

ESP32 employs a TCP/IP and full 802.11 b/g/n Wi-Fi mac protocol. Among other features, the Wi-Fi of ESP32 has a data rate of up to 150 Mbps and transmitting power of up to 20.5 dBm.

The microprocessor comes with many other features like Bluetooth, Clocks, Timers and many more. More details about the specifications of the ESP32 family of chips can be found in the ESP32 Datasheet by Espressif Systems. A link to the datasheet is given here [28]. The diagram below, courtesy of Random Nerd Tutorial, shows the pin layout of an ESP-WROOM-32 PCM, including all GPIOs, ADCs, Ground, V-input, etc. Micro-USB port, EN and Boot button can also be seen here.

Relay module:

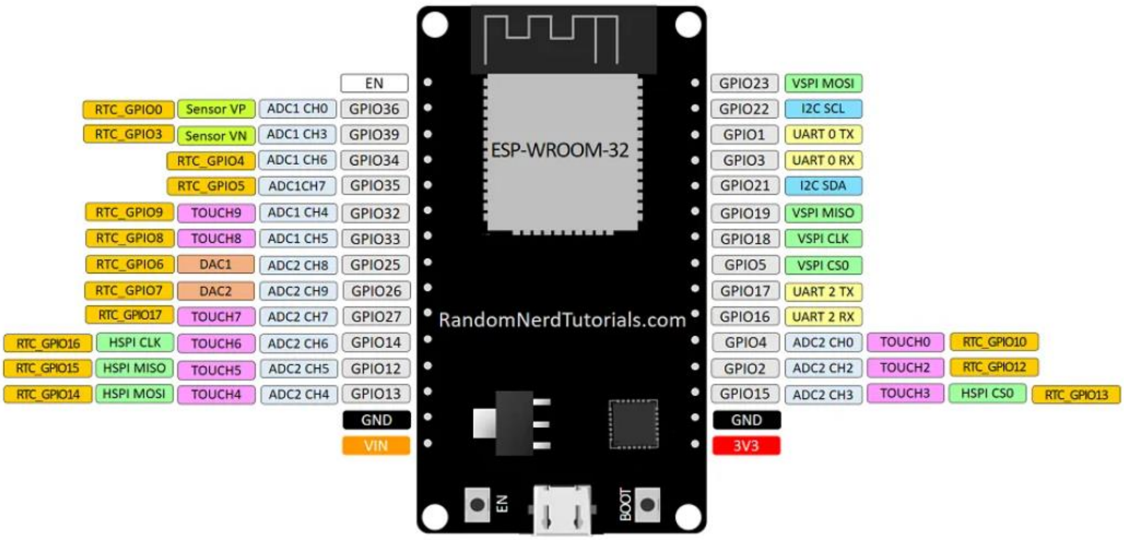

Figure 6:Pin Layout of ESP-WROOM-32 PCB [28]

The relay module used in this project is 5V Four-Channel Relay. It is one of the most used modules for home automation and high current load switching. The four-channel relay module has four 5V relays as well as the accompanying operating and separating components, allowing for simple interaction with a microcontroller or sensor with the fewest possible components and connections. There are two terminal blocks, each with six terminals, and two relays share each block. The terminals are screw style, allowing for convenient and straightforward connection to mains wiring [29].

Specification [29]:

- Input Voltage: 3.75V-6V
- Activating Current: 5mA
- Active Relay Current: 70mA (single), 300mA (all four)
- Relay Maximum Voltage: 250VAC, 30VDC
- Relay maximum Current: 10A

Figure 7:A Four Channel Relay Model [29]

Light Socket:

It is an electrically connected device that mechanically supports and connects compatible electric lighting. In addition, sockets make it possible to replace lamps safely and conveniently (re-lamping). Two types one light sockets can be found in the market. Pin-type and Screw-type. We have used an E27 Screw-type socket.

Light bulb:

There are four different types of light bulbs available for residential use: Incandescent, Halogen, Fluorescent and LED. We have used an LED-type light as it the most energy-efficient, environmentally friendly, and has the most extended lifespan [30].

Specifications [30]:

- Power: 10W
- Input Voltage: 100-240 VAC
- Luminous Flux: 780-810lm
- Color Temperature: White
- Power Factor: 0.5 min
- Life: 30,000 hours

Figure 8:A LED bulb [30]

WIFI Router:

A wireless router, commonly known WIFI-Router, is a device that combines the capabilities of a router and a wireless access point into one device. It is used to connect to the Internet or a private network of computers. It can work on a wired local area network, a wireless-only LAN, or a hybrid wired and wireless network, depending on the manufacturer and model. We have used a ‘TP-LINK TL-WR840N 300Mbps Wireless N Router’ as it is one of the most readily available routers on the market [31].

Specification:

- Power Supply: 9VDC/0.6A
- Wireless Standard: IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
- Frequency: 2.4-2.4835 GHz
- Protocol: IPv4 and IPv6

Figure 9:A Wi-Fi Router [31]

3.2.3 Circuit

First of all, the code is uploaded to ESP32. Before making the full circuit, and high ac voltages, it is a good idea to test the microcontroller and the code uploaded in it first, by using LEDs. For the test, instead of connecting the relay module to ESP32, four LEDs are connected to the four ports which are programmed to go “High” upon command, in this case, D15, D2, D4 and D5. And the ground is connected accordingly. First, the user interface is checked, if it is loading properly. After checking, when the commands are given from the server, if the appropriate LEDs turn on, then the microprocessor is working fine, and the coding is done right. The test circuit is shown in the figure below.

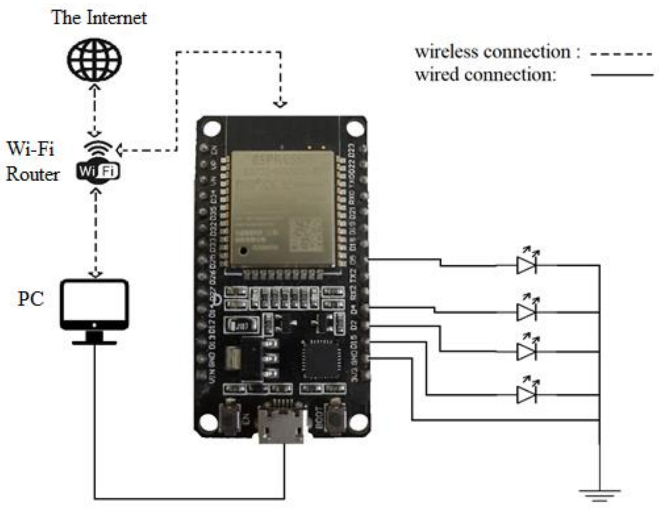

Figure 10:Circuit diagram for test circuit

After the successfully testing the user interface server, the microprocessor and the programming of the microprocessor, the actual circuit is implemented. The circuit diagram for the main circuit is shown below.

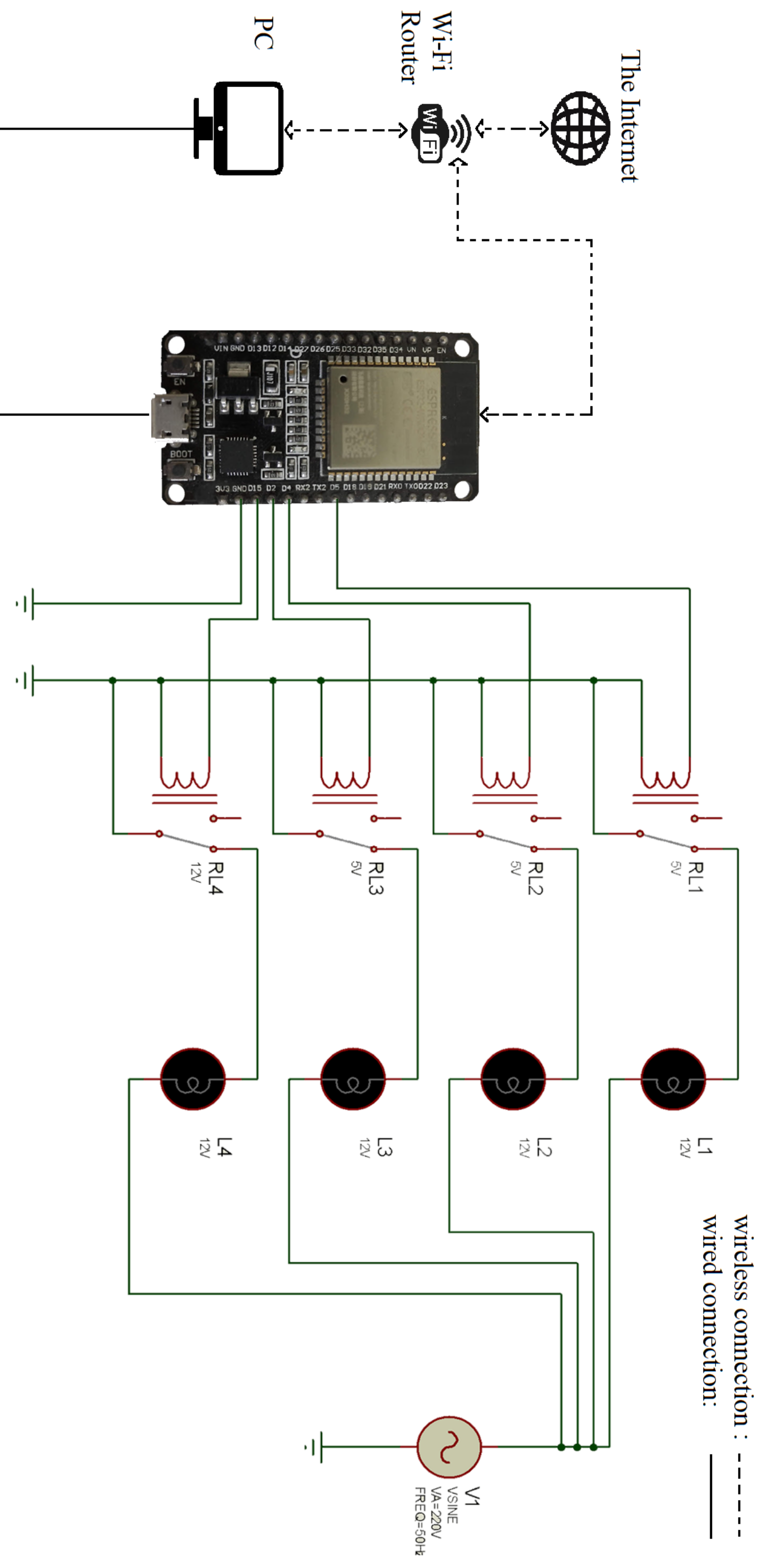

Figure 11:Full circuit for the project for remote operation

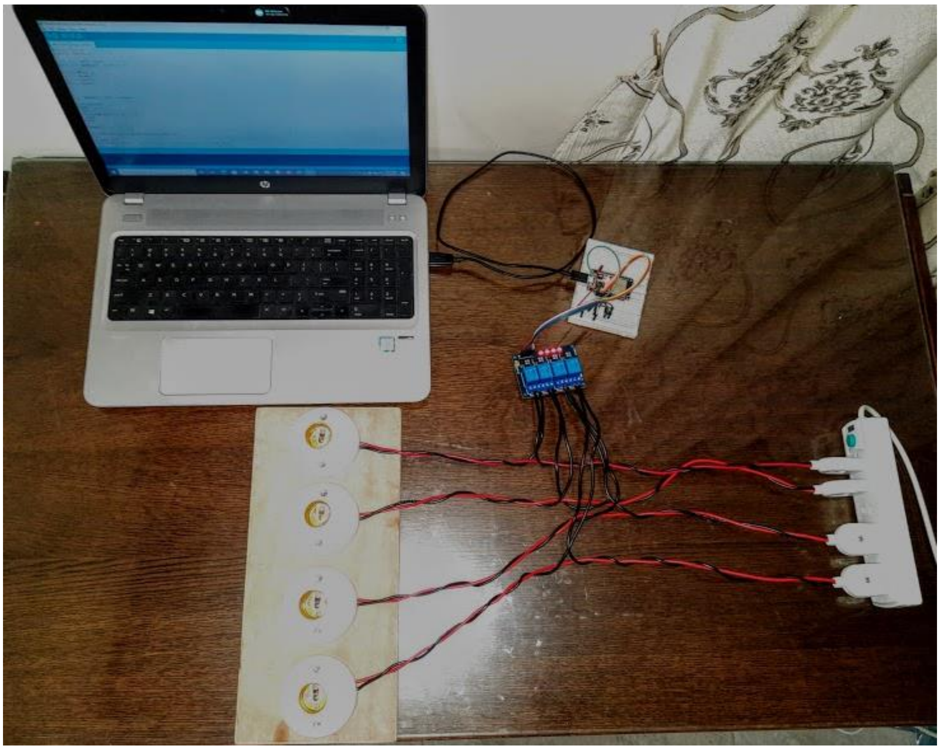

Figure 13:Photograph of the implemented circuit.

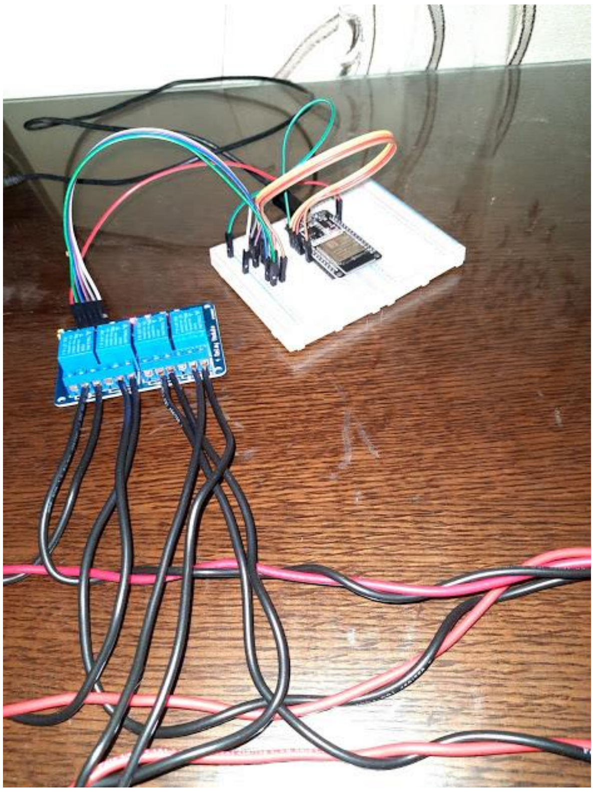

Figure 12:Photograph showing ESP32, Relay Module and their connections.

3.2.3 Hardware Setup

ESP32 is more than a microcontroller in this project, it also acts as the host for the website that is the user interface. Four GPIO of the microcontroller is programmed to become high on command. These GPIO is connected to the four-input port of the relay module. When ESP32 is powered through its micro-USB port, its V-in port act as a 5-volt output, which is connected to the V- in of the relay module. And the grounds were connected accordingly. The relay module used in the project is a 4-channel relay module. Each relay of the module is designed to perform under 5volt DC voltage and up to 250-volt AC voltage. The 1.3mm wires are connected to the relays in “normally closed” mode.

When the system is to be operated remotely, the ES32 board is connected to a computer via USB, to power the board and to initiate the connection between the Wi-Fi router and the microcontroller. However, when the system is operated in ethernet mode, it is not necessary to use a pc and router. A mobile phone charger sufficient to power the device.

3.3 Software and Codes

3.3.1 Software

Arduino IDE: Arduino Integrated Development Environment or Arduino IDE is a software developed by Arduino Software to write codes and upload them to Arduino boards. However, it can also be used to run and upload code to other boards like ESP32, with some modifications. Arduino IDE is free to use and can be run on Windows, Mac, Linux Raspberry Pi devices [32].

To use Arduino IDE with ESP32, first, the ESP board tool has to be installed to Arduino IDE then the AsyncTCP library is installed which allows us to host the server. The entire coding for this project is written and uploaded using Arduino IDE. Arduino IDE also has a 'serial monitor' feature, using which the board is connected to the Wi-Fi router. The serial monitor is only required when the system is to be done remotely.

Ngrok: Ngrok is a tunnel service application that is used to expose a server on a local device to the internet. Ngrok is free to use and can be downloaded from their website. This application can make a server, which is locally hosted, appear as a subdomain of Ngrok. If someone has their domain, then they can use that instead of Ngrok [33].

3.3.2 Codes and Programming:

The codes for the project are shown in the figures below:

```
#include <WiFi.h>
#include <ESPmDNS.h>

const char* ssid = "RAKA";/*ssid of WiFi router goes here*/
const char* password = "01727522673";/*passwordd of WiFi router goes here*/

#define Relay1 15
#define Relay2 2
#define Relay3 4
#define Relay4 5


char webpage[] PROGMEM = R"=====(

<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<center>


<h1>Power Grid Control Through SCADA with ESP32</h1>
<hr>
<h3>Load 1</h3>
<button onclick="window.location = 'http://'+location.hostname+'/s1/on'">ON</button>
<button onclick="window.location = 'http://'+location.hostname+'/s1/off'">OFF</button>
<h3>Load 2</h3>
<button onclick="window.location = 'http://'+location.hostname+'/s2/on'">ON</button>
<button onclick="window.location = 'http://'+location.hostname+'/s2/off'">OFF</button>
<h3>Load 3</h3>
<button onclick="window.location = 'http://'+location.hostname+'/s3/on'">ON</button>
<button onclick="window.location = 'http://'+location.hostname+'/s3/off'">OFF</button>
<h3>Load 4</h3>
<button onclick="window.location = 'http://'+location.hostname+'/s4/on'">ON</button>
<button onclick="window.location = 'http://'+location.hostname+'/s4/off'">OFF</button>
</center>
<hr>
<p>This website is part of a project submitted to the Depertment of EEE,BRACU.</p>

</body>
</html>

)=====";
```

Figure 14:Code for the project (part 1)

```
#include <ESPAsyncWebServer.h>
AsyncWebServer server(80); // server port 80



void notFound(AsyncWebServerRequest *request)
{
  request->send(404, "text/plain", "Page Not found");
}



void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);

  pinMode(Relay1,OUTPUT);
  pinMode(Relay2,OUTPUT);
  pinMode(Relay3,OUTPUT);
  pinMode(Relay4,OUTPUT);





  WiFi.begin(ssid,password);
  while(WiFi.status() !=WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the Wifi network");
  Serial.println(WiFi.localIP());
```

Figure 15:Code for the project (part 2)

```
 server.on("/", [](AsyncWebServerRequest * request)
{

request->send_P(200, "text/html", webpage);
});

 server.on("/s1/on", HTTP_GET, [](AsyncWebServerRequest * request)
{
  digitalWrite(Relay1,HIGH);
request->send_P(200, "text/html", webpage);
});
   server.on("/s1/off", HTTP_GET, [](AsyncWebServerRequest * request)
{
  digitalWrite(Relay1,LOW);
request->send_P(200, "text/html", webpage);
});
 server.on("/s2/on", HTTP_GET, [](AsyncWebServerRequest * request)
{
  digitalWrite(Relay2,HIGH);
request->send_P(200, "text/html", webpage);
});
   server.on("/s2/off", HTTP_GET, [](AsyncWebServerRequest * request)
{


  digitalWrite(Relay2,LOW);
request->send_P(200, "text/html", webpage);
});
   server.on("/s3/on", HTTP_GET, [](AsyncWebServerRequest * request)
{
  digitalWrite(Relay3,HIGH);
request->send_P(200, "text/html", webpage);
});
   server.on("/s3/off", HTTP_GET, [](AsyncWebServerRequest * request)
{
  digitalWrite(Relay3,LOW);
request->send_P(200, "text/html", webpage);
});
 server.on("/s4/on", HTTP_GET, [](AsyncWebServerRequest * request)
{
  digitalWrite(Relay4,HIGH);
request->send_P(200, "text/html", webpage);
});
   server.on("/s4/off", HTTP_GET, [](AsyncWebServerRequest * request)
{
  digitalWrite(Relay4,LOW);
request->send_P(200, "text/html", webpage);
});
server.onNotFound(notFound);

server.begin();
}
```

Figure 16:Code for the project (part 3)

The figures above collectively show the codes required to run the grid control system remotely. The HTML codes for the user interface is embedded into this code. The HTML code yields in the website shown in the following figure:

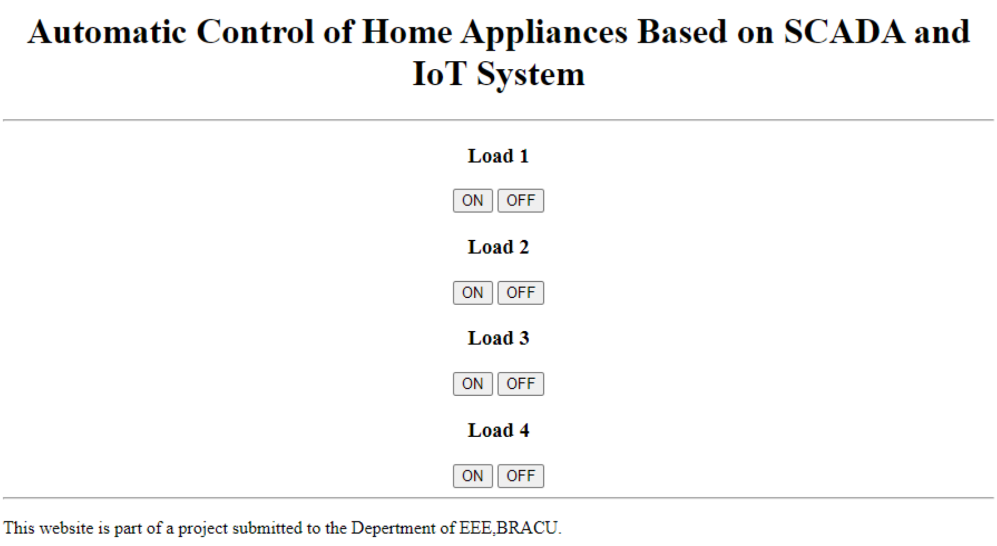

Figure 17:Screen shot of the user interface website

The ON and OFF buttons are used to send commands to the ESP32 board which controls bulbs through the relays. Pressing the ON button turns the respective bulb on and vice versa.

To control the system centrally only, mild alterations are required in the codes. The following part of the codes are for remote operations and this part of the code is responsible for connecting the ESP32 board to the internet:

```
WiFi.begin(ssid,password);
while(WiFi.status() !=WL_CONNECTED) {
  delay(500);
  Serial.println("Connecting to WiFi..");
}
Serial.println("Connected to the Wifi network");
Serial.println(WiFi.localIP());
```

Figure 18:Part of the codes establish connection between ESP32 and Wi-Fi router.

The above part of the codes is needed to be replaced with the codes shown below to operate the system only centrally:

```
WiFi.softAP("Default", "");
 Serial.println("softap");
 Serial.println("");
 Serial.println(WiFi.softAPIP());


 if (MDNS.begin("GCS")) { //gcs.local/
   Serial.println("MDNS responder started");
 }
```

Figure 19:Part of the code that create local connection

The part of the codes, shown above, is responsible for creating a local network, to which other devices can be connected.

3.4 Outcome and Results

After writing and uploading the code, the circuit is implemented. To connect the ESP32 board to the internet, the serial monitor of Arduino IDE is launched. Now the EN button of ESP32 is pressed and the microprocessor is connected to the internet. After connection, the local IP address of the board can be seen on the serial monitor.

Once the ESP32 board is connected to the internet, Ngrok is launched. Authtoken is taken from the Ngrok website and entered in Ngrok, and after that, the IP address of ESP32, received from the serial monitor earlier, is inputted in Ngrok. Then Ngrok provides a link, which can now be used to access any device in any network. The user interface can be found in figure 17 in the previous page.

At first the user interface is accessed form the same network that the ESP32 is connected to. Then the “ON” buttons for ‘Load 1’ is pressed and it was observed that the bulb for ‘Load 1’ is turned on. After that the “OFF” button for the same load is pressed and it was observed that the light is now turned off. This process was repeated for all other loads and each load is tested ten times, and every time the bulbs turned on or off accordingly. This means we can now successfully turn these loads on or off – wirelessly.

The interface is accessed from a different device in a different network and the process is repeated again and same result was obtained. After that, the user interface is accessed from two different network and a light is turned on from one network and switched off from another and vice versa. This essentially means that the bulbs can now be controlled from anywhere in the world.

Chapter 4

Application and Safety

4.1 Practical Application of the system

This project aimed to develop a system that could remotely monitor and control household appliances. Aided by proper technical components, this system can have limitless possibilities – from managing household appliances to gardening to security and many more. This system is mainly designed so that a user can manage their home appliance while being far from home. Some of the ways this system can be used are discussed below.

A) Controlling Home appliance: This system can be used to control any home appliance remotely. Most common home automation systems nowadays use IR based or use intranet which only allows the user to control their devices locally and on the other hand, our system can allow the user to control their devices from anywhere as long as they have internet access. Users can be relieved from wastage of money, or the hassle of coming back home, just because they forgot to turn off their appliance before leaving home – they can now turn their appliance from anywhere using this system. Users can also turn on their appliance in advance for example turning on the water heater before they even reach home – no more waiting before they can take a warm bath. A user can even use this system to remotely cut-off power to their apartment if needed. Many other functions can be executed if the system is incorporated with sensors, motors and other devices. However, appropriate types of relays or relay modules must be used with corresponding devices, which means, to control light appliances like fan, bulbs etc., a relay with a maximum voltage of 250 VAC will suffice, but to control a device like an air conditioner or water heater relays with higher maximum voltage and current must be used.

B) Gardening and farming: This system can be used with small water pumps to the water plant. Gardeners or farmers cannot afford to go away for vacation or other purposes, as they worry that their plants or crops may dry up or even die since they cannot water their plants. In case of emergency, and they must leave their home, they have to go through the trouble of managing other people to water their plants or risk damage to their plant or crops. Farmers or people who have an interest in gardening will find this system extremely helpful as they can stay away from their homes and still water their plants. Normally, for small scale gardens, a small pump is sufficient, however for large farms, larger pumps with heavier relays must be used.

Figure 20:A 12-volt small water pump.[34]

C) Automatic Doors: This system can also be used, with help from motors and other mechanical components, to remotely and automatically control sliding doors and windows, for instance, in case of storms and rains.

Home automation is becoming increasingly popular and is expected to gain more popularity in the years to come. According to a report from Fortune Business Insight, it is expected that by 2025, home automation will have a share of 144 billion US dollars globally.

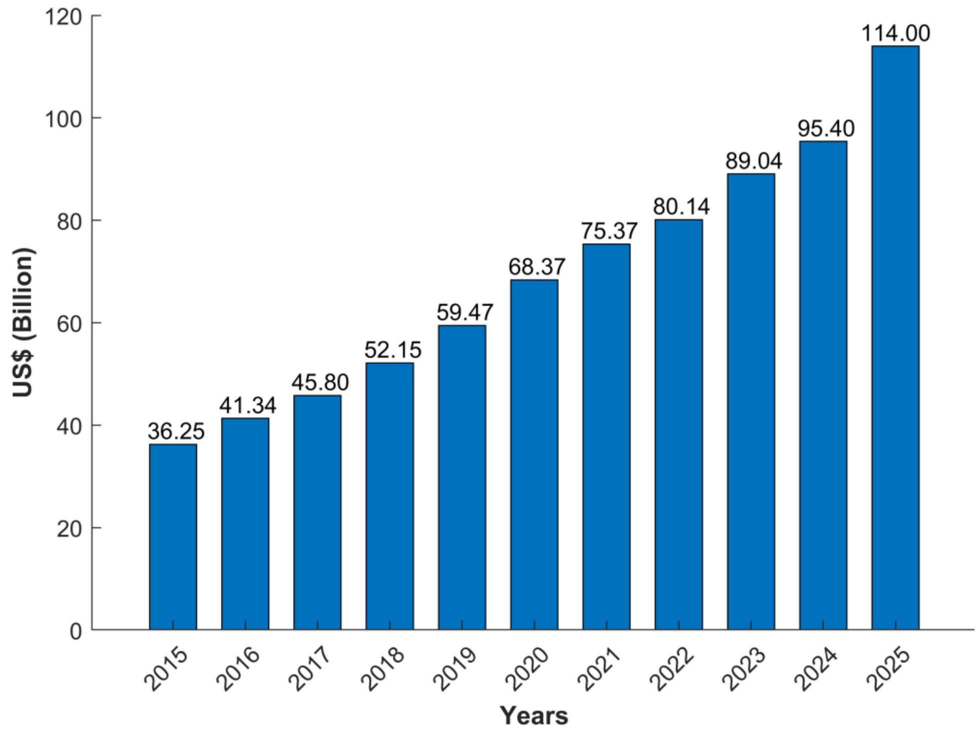

Figure 21:Global Home Automation Market size from 2015 to 2025

Apart from household appliances, this system can be used for other industrial purposes, for example:

i. Oil pipeline: Oil companies can use this type of system to remotely control the pipeline. An operator can monitor and open/close valve the valves of the pipeline

ii. Power distribution: Most power distributors rely on manual labor to carry out distribution related tasks such as connecting or interrupting power to loads, repairing faults, restoring service, and taking voltage and current readings by every hour. This system can help automate this process, lowering the use of manual labor and increasing efficiency.

iii. Renewable energy generation: This type of system can continuously monitor energy generation from various renewable sources. Which can help in better integration of renewable sources to our electrical grid.

iv. Smart grid: This system can help our grid become smarter. The system can control relays and various switchgear devices remotely and can monitor their condition. This can help in detecting and clearing faults faster and safely.

4.2 Safety

As the system is connected to the internet it is susceptible to cyber-attack. A malevolent individual or group might bring a SCADA system to its knees. Hackers might wreak havoc by getting access to vital SCADA components, causing everything from service disruption to cyber warfare. SCADA systems are also vulnerable to malware, such as viruses, spyware, and ransomware. While malware may not be able to target the network directly, it can nevertheless represent a threat to the critical infrastructure that aids in the management of the SCADA network. Thus, protecting the system also takes priority.

As the system is designed to share data via the internet, isolating the network is not an option. An appropriate firewall is needed between the hardware and software components. A verification system can also be created where only the authorized users will get excess to the system.

As for physical safety, appropriate switchgear and protection must be employed. To prevent any electricity-related accident, equipment such as circuit breaker, fuse etc. must be used. This will not only make it safer for users but also prevents hardware from being damaged.

Chapter 5

Challenges

5.1 Challenges faced

The following challenges were faced by us during the research and implementation of this project.

i. The libraries for ESP32 programming in the Arduino IDE are quite hard to find. There aren't many resources available and working with it felt restrictive. Additionally, there were various types of Arduino boards available. However, the Arduino not having a built-in Wi-Fi module made us look into the ESP32

ii. As students of the EEE background, our knowledge regarding HTML was also very limited which produced the result of a very simple user interface.

iii. An internet connection is required for the system to keep running. Furthermore, a PC is also required to connect the ESP32 to the router. This makes the system less practical since it's not compact.

iv. We do not have a visual indicator inside the website for the loads. Hence, we do not have any way of signaling whether it is on or off.

Additionally, the recent Covid-19 global pandemic posed a great challenge. Initially, we spent a great deal of time being confused and not knowing how to continue the project. Most of the meetings and conversations between the teammate and with our Honorable supervisors had to be done online as meeting physically was difficult. Getting the hardware parts was another challenge owing to the lockdown due to Covid-19.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In conclusion, the project creates a small-scale SCADA system for use in households, which helps achieve remote connectivity. The system can continuously monitor and execute given commands when needed. It can also help in the various sectors where remote connectivity is required. SCADA systems generally build with an isolated network, where there's additional equipment for communication purposes. The system's novelty is that it integrates the communication within the system to reduce cost and make the system less complex and easier to operate. In addition, it reduces the components needed to work.

6.2 Future Work

Here are some improvements that can be employed to make the system more effective:

i. To solve the issue of not being compact, we can utilize the Raspberry Pi instead of a PC. This will make the project more practical and portable.

ii. The issue of the Internet can be solved if we used a mobile network-based router. Additionally, a separate power supply can be added to supply constant internet connectivity.

iii. Upon increasing the knowledge of the ESP libraries and HTML, the user interface website can be developed further to possess Internet security options where the user will be required to type in their login information. This will prevent hackers from stealing sensitive data.

References

1. *What is SCADA?* Inductive Automation. (n.d.). Retrieved October 21, 2021, from https://inductiveautomation.com/resources/article/what-is-scada

2. *RF Wireless World*. Advantages of SCADA System | disadvantages of SCADA System. (n.d.). Retrieved September 27, 2021, from https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-SCADA.html.

3. *RF Wireless World*. Advantages of IoT | Disadvantages of IoT | Internet of Things. (n.d.). Retrieved September 27, 2021, from https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-IoT-Internet-Of-Things.html.

4. Daneels, A., & Salter, W. (1999). What is SCADA.

https://accelconf.web.cern.ch/ica99/papers/mc1i01.pdf

5. Raghvendra, N. (2019, February 12). *SCADA system - components, Hardware & Software Architecture, types*. electricalfundablog.com. Retrieved September 27, 2021, from https://electricalfundablog.com/scada-system-components-architecture/.

6. Raghvendra, N. (2019, February 12). *SCADA system - components, Hardware & Software Architecture, types*. electricalfundablog.com. Retrieved September 27, 2021, from https://electricalfundablog.com/scada-system-components-architecture/.

7. Anton, S. D. D., Fraunholz, D., Lipps, C., Pohl, F., Zimmermann, M., & Schotten, H. D. (2017). Two decades of SCADA exploitation: A brief history. *2017 IEEE Conference on Application, Information and Network Security (AINS)*, 98–104. https://doi.org/10.1109/ains.2017.8270432

8. Process Solutions, Inc. (2020, July 10). A Brief History of the SCADA System [web log]. Retrieved September 27, 2021, from https://processsolutions.com/a-brief-history-of-the-scada-system/

9. Ujvarosi, A. (2016). EVOLUTION OF SCADA SYSTEMS.

http://webbut.unitbv.ro/BU2015/Series%20I/2016/BULETIN%20I%20PDF/Ujvarosi_Al.pdf

10. Anton, S. D. D., Fraunholz, D., Lipps, C., Pohl, F., Zimmermann, M., & Schotten, H. D. (2017). Two decades of SCADA exploitation: A brief history. *2017 IEEE Conference on Application, Information and Network Security (AINS)*, 98–104. https://doi.org/10.1109/ains.2017.8270432

11. Yadav, G., & Paul, K. (2020). Architecture and security of SCADA systems: A Review. *International Journal of Critical Infrastructure Protection*, *33*. https://doi.org/10.1016/j.ijcip.2021.100433

12. Parashar, D., & Harrelson, E. (n.d.). *Functions of SCADA - tutorials of SCADA - InstrumentationTools*. Inst Tools. Retrieved October 21, 2021, from https://instrumentationtools.com/functions-of-scada/

13. *How do scada systems work?* How SCADA Systems Work - data, sensors, networks, and RTUs. (n.d.). Retrieved September 27, 2021, from https://www.dpstele.com/scada/how-systems-work.php.

14. Yadav, G., & Paul, K. (2020). Architecture and security of SCADA systems: A Review. *International Journal of Critical Infrastructure Protection*, *33*. https://doi.org/10.1016/j.ijcip.2021.100433

15. *What is the internet of things (IOT)?* What Is the Internet of Things (IoT)? (n.d.). Retrieved September 27, 2021, from https://www.oracle.com/internet-of-things/what-is-iot/.

16. *IOT architecture layers*. HIOTRON. (2019, July 12). Retrieved September 27, 2021, from https://www.hiotron.com/iot-architecture-layers/

17. Ashraf, I., Umer, M., Majeed, R., Mehmood, A., Aslam, W., Yasir, M. N., & Choi, G. S. (2020). Home Automation using general purpose household electric appliances with Raspberry Pi and commercial smartphone. *PLOS ONE*, *15*(9). https://doi.org/10.1371/journal.pone.0238480

18. Jones, C. (2019, May 14). *The birth and development of SCADA*. The Earth Awards. Retrieved September 27, 2021, from https://www.theearthawards.org/a-brief-history-of-the-scada-system/

19. Aghenta, L. O., & Iqbal, M. T. (2019). Low-cost, open source IOT-based SCADA system design using Thinger.IO and ESP32 thing. *Electronics*, *8*(8), 822. https://doi.org/10.3390/electronics8080822

20. Sagues, P. (2013, August). Small Scale SCADA. *Treatment Plant Operator*. Retrieved September 27, 2021, from https://www.tpomag.com/editorial/2013/08/small_scale_scada_wso.

21. Sivaranjith. (2018, December 8). *How a SCADA works? and functions of SCADA*. Instrumentation and Control Engineering. Retrieved October 21, 2021, from https://automationforum.co/how-a-scada-works-and-functions-of-scada/

22. Pillai, S. (2016, March 30). *SCADA functions*. SlideShare. Retrieved October 21, 2021, from https://www.slideshare.net/sundarapandy/scada-functions \

23. *What is SCADA?* Inductive Automation. (n.d.). Retrieved October 21, 2021, from https://inductiveautomation.com/resources/article/what-is-scada

24. Aghenta, L. O., & Iqbal, M. T. (2019). Low-cost, open source IOT-based SCADA system design using Thinger.IO and ESP32 thing. *Electronics*, *8*(8), 822. https://doi.org/10.3390/electronics8080822

25. Chestnut, D. (2019, December 18). *What is a data processing system?* It Still Works. Retrieved October 21, 2021, from https://itstillworks.com/data-processing-system-6701338.html

26. RF Wireless World. Advantages of SCADA System | disadvantages of SCADA System. (n.d.). Retrieved September 27, 2021, from https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-SCADA.html.

27. RF Wireless World. Advantages of IoT | Disadvantages of IoT | Internet of Things. (n.d.). Retrieved September 27, 2021, from https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-IoT-Internet-Of-Things.html.

28. Espressif Systems.(2021). *ESP32 Series*. Retrieved October 21, 2021, from

esp32_datasheet_en.pdf (espressif.com)

29. *5V four-channel Relay Module*. Components101. (2020, January 11). Retrieved September 27, 2021, from https://components101.com/switches/5v-four-channel-relay-module-pinout-features-applications-working-datasheet

30. *10 watt LED bulbs*. 10W LED Bulbs - Manufacturer, Supplier, Exporter. (n.d.). Retrieved September 27, 2021, from http://www.eslightbulbs.com/products/10-watt-led-bulbs

31. 300Mbps wireless N router. TP. (n.d.). Retrieved September 27, 2021, from https://www.tp-link.com/us/home-networking/wifi-router/tl-wr841n/#specifications.

32. Ashraf, I., Umer, M., Majeed, R., Mehmood, A., Aslam, W., Yasir, M. N., & Choi, G. S. (2020). Home Automation using general purpose household electric appliances with Raspberry Pi and commercial smartphone. *PLOS ONE*, *15*(9). https://doi.org/10.1371/journal.pone.0238480

33. Documentation. ngrok. (n.d.). Retrieved September 27, 2021, from https://ngrok.com/docs.

34. *Mavel Star 12 volt small mini submersible water pump for Camping DIY swamp cooler PC CPU water cooling fountain water fall 63 GPH*. Desertcart. (n.d.). Retrieved September 27, 2021, from https://bangladesh.desertcart.com/products/42115736-mavel-star-12-volt-small-mini-submersible-water-pump-for-camping-diy-swamp-cooler-pc-cpu-water-cooling-fountain-water-fall-63-gph-upgraded-version