KIRINYAGA UNIVERSITY

SCHOOL OF ENGINEERING AND TECHNOLOGY

BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMPUTER ENGINEERING.

FINAL YEAR PROJECT REPORT

1. **ALLAN KIPTANUI NDEGE EE100/G/14638/21**
2. **MICHAEL KAMAU EE100/G/14613/21**
3. **MARTIN KAMAU EE100/G/14612/21**

IoT HOME AUTOMATION WITH ESP32 AND PZEM-004T

A Final Year Project Report Submitted to The School of Engineering and Technology as Part of the Requirements for The Award of Bachelor of technology in Electronics and Computer Engineering.

APRIL 2025

DECLARATION

1. I declare that this final year project report is my original work and has not been submitted elsewhere for examination, award of a degree, diploma or publication. Where other people's work, or my own work has been used, this has properly been acknowledged and referenced in accordance with the University's requirements.
2. I have not sought or used the services of any professional agencies to produce this proposal.
3. I have not allowed, and shall not allow anyone to copy my work with the intention of passing it off as his/her own work.
4. I understand that any false claim in respect of this work shall result in disciplinary action.

1. ALLAN KIPTANUI NDEGE EE100/G/14638/21 25/03/2025
2. MICHAEL KAMAU EE100/G/14613/21 25/03/2025
3. MARTIN KAMAU EE100/G/14612/21 25/03/2025

TITLE OF PROPOSAL:

IoT AUTOMATED HOME USING ESP32 AND PZEM-004T CONTROLS

SUPERVISOR CONFIRMATION:

This project report has been submitted to the School of Engineering and Technology, Kirinyaga University with my blessing as the supervisor:

NAME: ……………………………………………………….

SIGNATURE: ………………………………………………..

DATE: ………………………………………………………….

ABSTRACT

The idea of automating homes with IoT technology has quickly become a reality, moving away from being only a future concept. This evolution represents more than just a convenient upgrade, but rather a fundamental transformation in how we engage with areas that are occupied by people [1]. The central idea of this change is the "smart home," where technology smoothly integrates with daily activities to promote enhanced efficiency, control, and adaptability. The Internet of Things (IoT) has transformed technology by allowing for remote control of physical objects through the Internet. This report explores the creation of an advanced smart home automation system that utilizes IoT switches connected to the ESP32 module to enable remote control capabilities. Our project proposes developing a home automation system with IoT switches using an ESP32 microcontroller to control and monitor electrical devices remotely through a web interface. With functionalities integrated through Telegram, users can control switches, sockets, and doors, and monitor real-time power usage using a PZEM-004T meter to measure the amount of power the system consumes. The system include servo motors for door control in opening and closing, an LCD for status updates on the amount of power consumed and current that flows into the system, and automation based on sensor data such as light intensity. Our project design enhances user convenience through automation with IoT, improves energy efficiency, and ensures the safe operation of devices.

Contents

List of figures

1. CHAPTER ONE

1.1. Introduction

IOT home automation, once something out of the future, but soon a real-life reality, driven principally by using Internet of Things (IoT). The transformation increases in comfort: is a radical shift in the worlds we inhabit. This change in IOT home automation can create a scenario where technology and daily life come together to enhance efficiency, control, and adaptability [1].

– Home automation is going to be the luxury of its time, limited to remote-control systems built to regulate lights, doors, power, and heating.

The rise of (IoT) has completely altered the way we think about technology. Think about a scenario in which normal ordinary objects, usually "things" transcend their normal stationary state and are saturated with sensory, software, and perhaps a bit of technological wizardry [1]. Its objects will be no longer standalone units but network elements, sharing information over the Internet. It will be injecting life into dead things, a seamless blending of the real and digital worlds, that will create astonishing possibilities for progress and comfort.

The IOT home automation market will include various types of functions such as automated lighting, doors, security system etc. [2]. But also, for more advanced parts like smart refrigerators and ovens. The machine can be controlled from far away and are getting smarter. They will learn about user's preferences and then automatically tune to maximize energy usage, safety and comfort. The effects of this technology is more than more convenience system [2]. It change how energy is consumed and directed at home. Sustainable IOT home systems enable better control of heating, lighting, and other power-intensive devices. The device gives more accurate energy management with lower wasted energy consumption monitoring by PZEM-004T module [2]. It is a question of relevance to rising energy costs and rising environmental concerns. Also, if we integrate IoT in the actual home automation then new opportunities for elder care and accessibility is possible and it allow greater autonomy to people who have limited mobility or are physically challenged with diseases [2]. Using IOT home automation patterns and remote monitoring solutions helps in making living easier and safer for older people or the disabled.

The trend towards a more interconnected and automated future of IoT more significant to change our homes [2]. IOT home automation shows us the sweeping potential of the Internet of Things (IoT) to transform our home experience, and help create smarter, more efficient and greener homes.

1.2. Problem Statement

With the Internet of Things (IoT), Automation in the dynamic and dynamic tech world is here. Even if this field has seen significant expansion, the gap between technological development and seamless incorporation into our life continues to be large. A lot of today's home automation solutions aren't as advanced and intuitive to really transform convenience and utility within homes. Furthermore, the existing approach suffers from remote access limitations, power usage and UI.

The problem is to design an IOT home automation system that exceeds the capabilities of existing technology and seamlessly combines IoT functionality with easy-to-understand interfaces and reliable remote-control capabilities. In a perfect system, you want it to not only be advanced with features such as remote access and low energy consumption but should also value user-friendliness and integration with current household conditions. Also, data security, interoperability and cost related problems need to be addressed as the priority to ensure wide adoption and long-term viability of home automation products.

The purpose of this project is to design and implement a smart IOT home automation system based on IoT, PZEM-004T module, servo motors, photoresistor and ESP32 module that overcomes the current limitations and re-engineer how people lives in homes. our project is aimed at the user who is looking for an all-in-one solution that offers remote access to your home appliances and power consumption monitoring in real time using a easy to use Telegram client and LCD screen. The purpose of this project is to integrate technological advancement with commercialization in domestic automation with an emphasis on convenience, efficiency and ease of use.

1.3. Problem Justification

Our application is more user friendly, more energy efficient, and a part of the expanding space of IoT smart home solutions. The effort also provides the platform for future home automation softwares which might include data analytics and AI **based automation system.**

1.4. Objectives

1.4.1 Main Objective

1. To create and install home automation system using IOT switches and doors controlled from a remote by ESP32 module via Telegram link.
2. To measure Power consumption data and display PZEM-004T meter.

1.4.2. Specific Objectives

1. To use sensor-based automation for energy savings.

2. To make them work safely with MCB and circuit breakers.

3. Develop and implement remote IOT switches to control home electrical devices.

4. To leverage ESP32 module wireless features to create a secure connection for remote control.

5. To have an easy interface so that anyone can manage their appliances from anywhere easily.

CHAPTER TWO: LITERATURE REVIEW

2.0. Introduction

This literature on home automation, IoT, energy management and smart home technologies provides a basic outline for what lies behind the development of IoT-enabled control systems in home appliances and the advantages it has. The article reviews the latest developments in the technology market and reveals the significance of ESP32-based automation, power monitoring system PZEM-004T, servo motors for doors and user interfaces such as Telegram for remote device management.

2.1.IoT-Enabled Home Automation Systems

Home automation solutions powered by IoT have accelerated due to the availability of cheap, interconnected, and web-based devices capable of exchanging data. We have investigated several home automation models for ease of use, security, and energy savings. As shown by Piyare and Tazil [1], IoT enables real-time control and monitoring of home appliances, simplifying usage and improving energy efficiency. Such systems heavily rely on microcontrollers, such as the ESP32, for cost-effective and scalable implementation. Several studies report that these microcontrollers offer excellent connectivity via Wi-Fi for seamless interaction with cloud-based applications and mobile devices.

Relevance to the Project: The ESP32-based system is leverage these IoT principles, allowing users to monitor and control home appliances remotely through Wi-Fi for real-time data exchange and control.

2.1.1. ESP32 in IoT Applications

Due to its dual-core processor, built-in Wi-Fi, and minimal power consumption, the ESP32 microcontroller has become the most popular choice for IoT devices. Researchers have stated that it is modular and capable of supporting multiple peripherals, making it an ideal candidate for complex systems that require sensor data processing and integration with IoT infrastructure. [1], the ESP32's ability to integrate with various sensors and relay modules makes home automation systems both manageable and powerful, even in large environments.

The controller consists of the ESP32, which will serve as the central hub and connect to components such as the PZEM-004T power meter, relay boards, and LCD monitor. This setup will enable the user, through Telegram, to control the appliances directly in real-time, minimizing control and feedback loops.

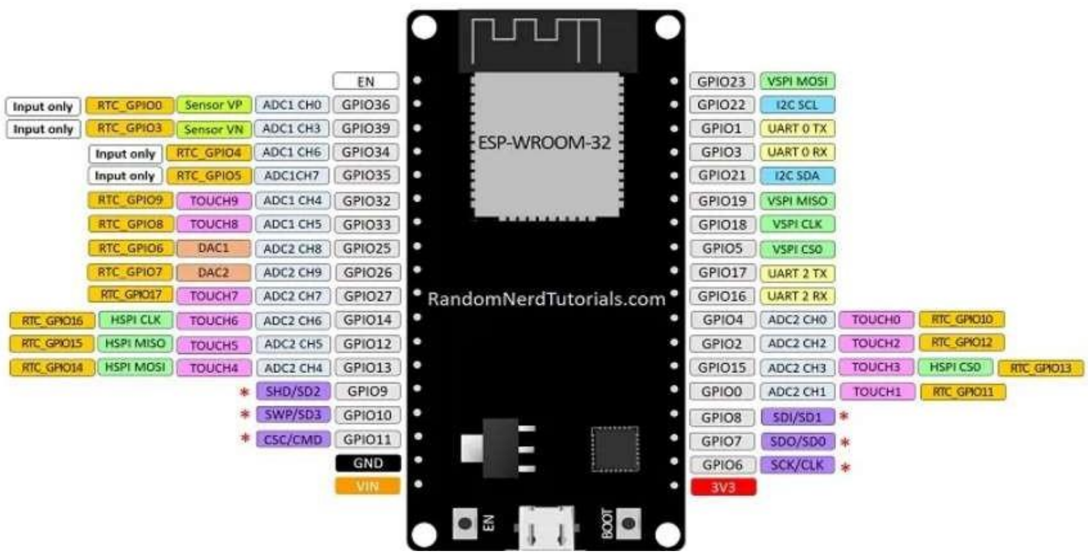

Figure 1:ESP32 Module

The ESP32 demonstrates exceptional efficiency and reliability in executing input and output commands. Its computational capabilities allow it to function as a central hub for managing various components and operations within a smart home environment [3]. The system effectively coordinates multiple aspects of domestic automation, including basic lighting control and complex management of systems such as home ventilation and air conditioning, which are crucial for maintaining residential comfort.

The ESP32 is equipped with a general-purpose I/O pin that interfaces with sensors, actuators, and 48-pin data display devices. Of these 48 pins, 15 are analog-to-digital converters (ADC), 2 are universal asynchronous receiver/transmitter (UART) serial ports, 25 are pulse-width modulation (PWM) pins, 2 are digital-to-analog converters (DAC), 3 are serial peripheral interface (SPI) pins, and 1 is an inter-integrated circuit (I2C) pin.

2.1.2. Relay Module

The 5V four-relay module is a useful smart home device that enhances the safety and efficiency of your home's technological setup. It functions as an electrical bridge between low-voltage control circuits and the high-power circuits that drive home appliances. This is a significant advancement in electrical engineering, combining technical success with a leap forward in safety and intelligence.

The relay module serves as a protective guardian of your home's electrical system, isolating the low-voltage control circuits from high-voltage circuits, thereby reducing the risk of electrical accidents. The key benefit is ensuring that your smart home remains secure, stable, and running harmoniously, making the working environment safer. One of the greatest achievements in electrical engineering is the separation of low-voltage control circuits from high-voltage circuits (typically found in household appliances) [4]. This separation enables the 5V relay module to ensure that the home electrical system remains safe.

Relay modules act as electromagnetic (EM) switches, where low-power circuits control high-power equipment. In IoT home automation, relay modules bridge the gap between low-voltage microcontrollers and high-voltage household appliances. Relays allow multiple devices to be controlled using a minimal power supply from the microcontroller

(ESP32)[5]. They isolate control circuits from powerful appliances, ensuring safe operation.

The 8-channel relay module enables the ESP32 to power multiple high-power devices (such as bulbs and sockets) with low-voltage signals, providing central control and maintaining separation between the microcontroller and high-voltage appliances.

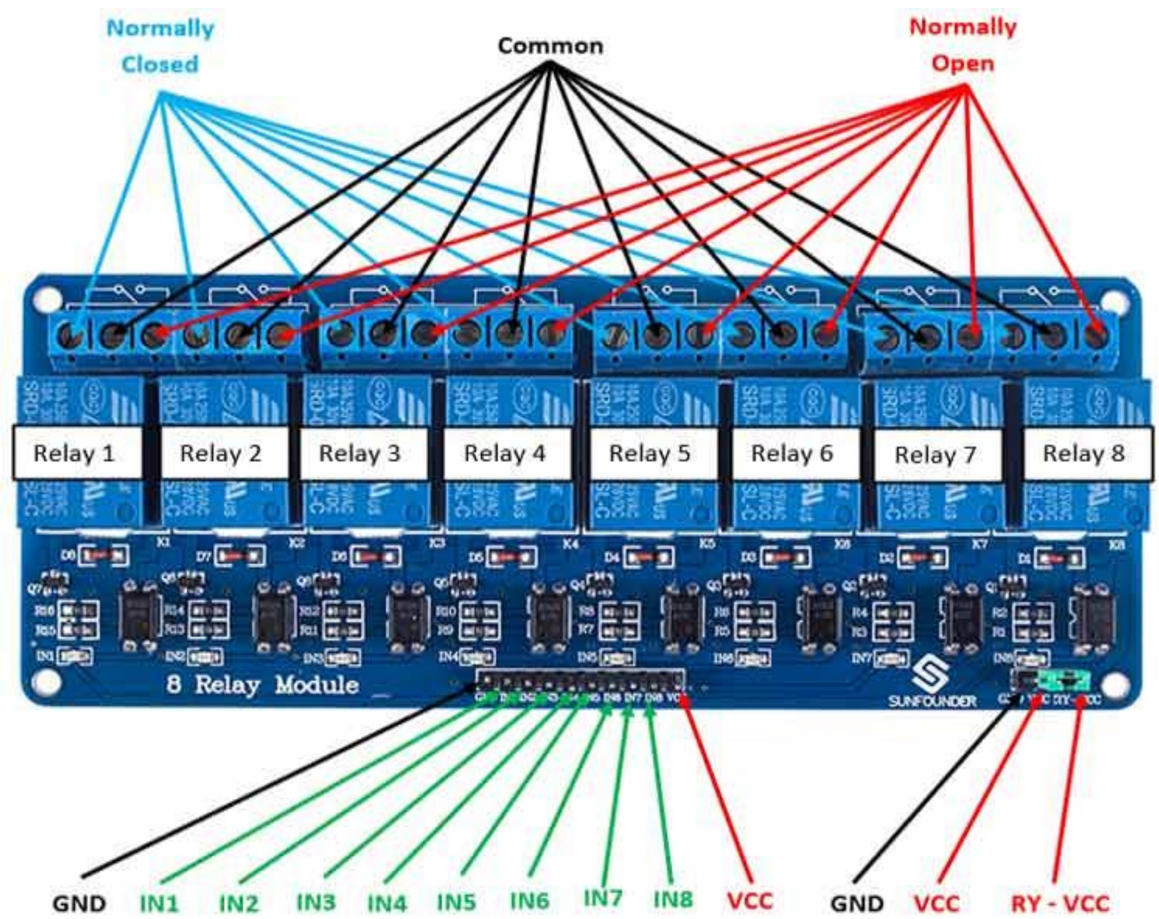

Figure 2: Relay module

2.2.Communication System

The foundation of any intelligent home automation system, particularly in installations based on the Internet of Things (IoT), is efficient communication. This section explores the several communication systems and protocols crucial in facilitating smooth interaction among different components within a smart home environment.

Role of Wireless Communication Technologies

Wi-Fi has revolutionized the communication industry. Wi-Fi is the essential component that facilitates the seamless transmission of data and commands between a microcontroller, such as an ESP32, and the various intelligent devices in your household [5]. The selection of a communication protocol frequently relies on various considerations, including but not limited to the range of communication, data need, power use and the particular requirements of the home automation system.

2.2.1. Wi-Fi Technology

Wireless networking is one of the most popular technologies nowadays. "Wi- Fi" is an acronym from "wireless fidelity". We can claim the Wi-Fi in its origin with the help of NCR Corporation and AT&T, Netherlands in the year 1991 [5]. This protocol is used to pass information between devices. This is what Wi-Fi has been all about, providing connectivity for the mobile devices (especially laptops). It is wireless LAN or local area network.

Wi-Fi can run local area networks without cables or wires. It is the popular choice for both residential and commercial networks [5]. Wi-Fi is generally known in the field of research as having the ability to send and receive data at lightning speeds and offer broad coverage. As such, it’s the best pick for smart home solutions with robust, high-bandwidth connections [5]. This technology’s ubiquity in the domestic space only further makes it ideal for incorporating different smart devices into a single connected network.

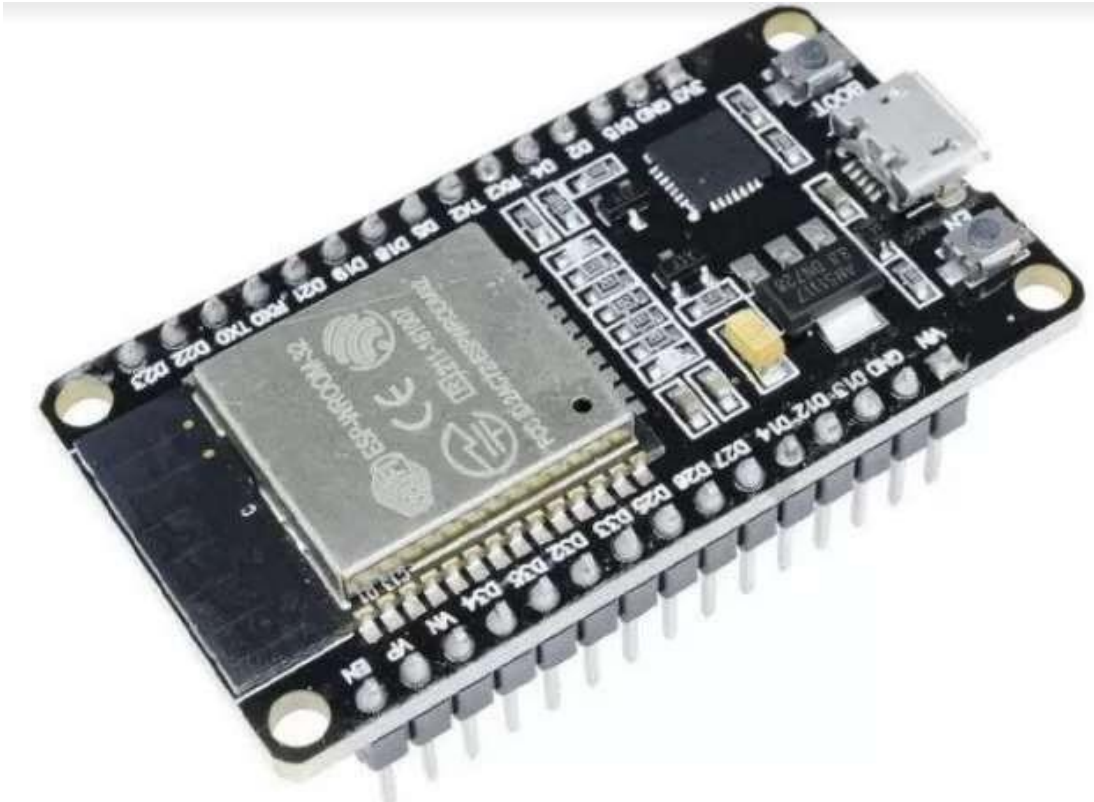

Figure 3: ES32 Wi-Fi Module

2.2.2. User interface in home automation

User interfaces are crucial for IoT systems as they simplify user interactions with complex automation tasks. The use of instant messaging platforms, such as Telegram, to create interactive and user-friendly interfaces for home automation systems. Telegram's Bot API enables secure and convenient communication with IoT devices, offering commands like turning appliances on/off and receiving status updates, making it highly effective for home automation applications. This approach is further enhanced by the ability to access the system from anywhere, ensuring users maintain control even when away from home. By using Telegram as the user interface,in the project allow users to control appliances, check power usage, and view the system's status in real-time. This not only simplifies interaction but also provides a secure platform for remote control.

The efficiency of smart home systems largely depends on the smooth synchronization between mobile applications and Internet of Things (IoT) devices [7]. It's similar to having a handheld device that provides unmatched ease and control for the entire home, acting as a universal remote control. These apps enable users to monitor home security and control lighting, among other tasks, using their smartphones. They employ wireless technologies like Wi-Fi and Bluetooth to communicate with IoT devices.

Rafi's research highlights the importance of connecting applications and gadgets to create a seamless mix that turns living spaces into dynamic and intelligent environments [7]. The project involved integrating the Telegram program as a user interface, together with Wi-Fi technology, to offer users a user-friendly means of controlling their smart home automation system. This interface selection provides a recognizable and easy-to-use experience, enabling homeowners to effortlessly control their domestic appliances and remotely monitor their living environment.

Additionally, Rafi's discoveries align with the main objective of the project, which is to improve user experience and accessibility in smart home automation [7]. The project seeks to enhance the control process of smart home systems by utilizing the features of Telegram and Wi-Fi technology. This result is a more user-friendly and efficient interaction for consumers, enabling smooth communication between gadgets and creating a unified and interconnected environment within the home.

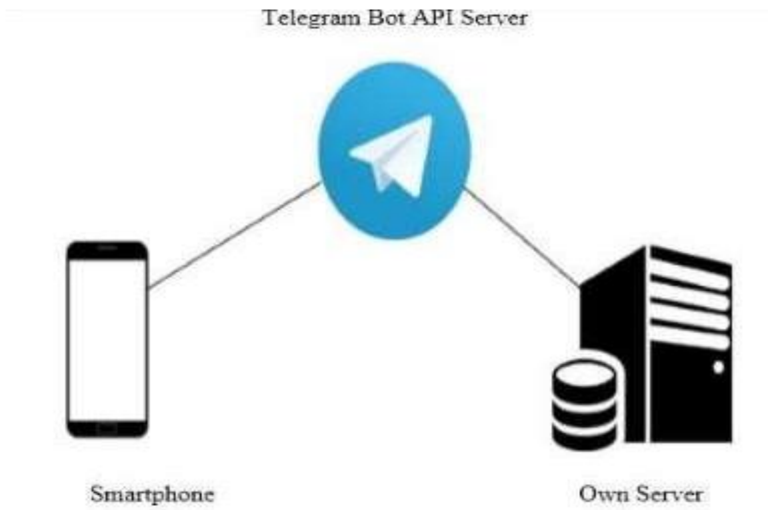

Figure 4: User interface connectivity

Based on the literature, we can expect the upward trajectory of smart home automation mobile applications [7]. Artificial intelligence and machine learning are allowing the building of smarter, user-predictable control systems.

2.3. Sensor-Based Automation in Smart Homes

Sensors that monitor the environment, such as photoresistors, enable smart home automation systems to gather information about their surroundings.[1] indicates that light sensors in home automation systems can help save energy, for instance, by dimming lighting based on the amount of natural light. Photoresistors can also program lighting systems to automatically adjust to user-defined thresholds, thereby eliminating unnecessary power consumption.

By incorporating photoresistors, this project automate lighting based on ambient light levels, making it more energy-efficient and user-friendly by eliminating the need for manual adjustments.

Figure 5: Photoresistor

Safety is a critical consideration in high-voltage home automation systems. Research emphasizes the need for circuit breakers and Miniature Circuit Breakers (MCBs) to protect IoT devices from electrical surges and short circuits, ensuring the safety of both devices and end users. [1] on circuit protection guidelines for IoT systems highlights that circuit breakers automatically disconnect faulty connections, minimizing equipment failure and fire risks.

The project will utilize MCBs and circuit breakers to safely operate high-voltage devices, such as lights and sockets, ensuring safety precautions for both users and equipment.

2.4.PZEM-004T Power Meter

The PZEM-004T power meter is a widely used device for tracking voltage, current, power, and energy consumption of electrical installations in real-time. In the context of IoT-enabled home automation, it provides essential data on energy use patterns, energy savings, and the safe operation of appliances. This review highlights the research and use cases of the PZEM-004T in IoT and automation solutions, focusing on its role in energy management for home automation systems. Power monitoring in home automation has become increasingly important as IoT technologies evolve, addressing the growing need for efficient energy management. [1] note that when connected to smart homes, power monitoring enables consumers to make informed decisions about energy consumption, saving electricity and promoting sustainability. By monitoring real-time power usage, home automation systems can even reprogram to prevent energy wastage by turning off unnecessary devices or notifying users when power usage is excessive.

The PZEM-004T is an AC voltage, current, active power, and energy consumption metering module specifically designed to provide accurate real-time measurements. Studies by Al-Mutairi et al. [2] test the module’s ability to monitor power use in high-voltage household scenarios. According to some studies, the PZEM-004T, when connected with a suitable current transformer (CT), can handle currents up to 100A, making it suitable for high-power appliances. Its ability to measure power factor and frequency provides a comprehensive view of power quality, which is crucial for safeguarding connected devices. The PZEM-004T supports UART communication, making it easy to integrate with microcontrollers like the ESP32. Highlight the PZEM-004T’s compatibility with most common IoT microcontrollers, as the UART-based communication simplifies data transmission and reduces computation time. The use of objects like PZEM004T.h makes it easy for developers to retrieve data in a user-friendly format, adaptable to various IoT projects. These studies emphasize that the simplicity of the communication protocol makes the PZEM-004T an invaluable component for IoT-enabled energy monitoring systems.

Figure 6: PZEM-004T Power Meter

2.5.Servo Motors

Servo motors play an essential role in home automation systems, particularly for applications requiring precise control of movement, such as door locks, window blinds, and automated mechanisms. In IoT-based systems, servo motors, controlled by microcontrollers like the ESP32, enable reliable and precise actuation in response to user commands or automated triggers. This literature review explores the applications, technical capabilities, and integration challenges of servo motors in home automation, specifically within IoT systems. Servo motors are widely used in home automation due to their precision, speed, and ability to handle mechanical loads, making them ideal for tasks such as opening/closing doors. Servo motors are crucial for applications that require specific angular movement and positioning. Unlike traditional motors, servos offer feedback control, allowing the microcontroller to continuously monitor the motor shaft's position. This feature enables highly controlled, accurate positioning in IoT applications.

The project uses servo motors to control doors or other actuated components, ensuring accurate and reliable control for security-related functions, such as locking mechanisms, through user commands via Telegram.

Figure 7: Servo motor

CHAPTER THREE

Methodology

The research and development (R&D) method's guidelines were adhered to when conducting this study. Prototype development studies have made extensive use of this strategy; some of these studies are shown in earlier ones. Steps one through five of the research process include requirements analysis, design, coding, testing, and maintenance.

The research commences with a survey aimed at comprehending the problem and the imperative need for a resolution. Subsequently, a solution is formulated and transformed into a prototype. The prototype undergoes additional development through programming, enabling it to operate by the stated specifications. Furthermore, the prototype that has been programmed is tested to ensure performance conformity with the design.

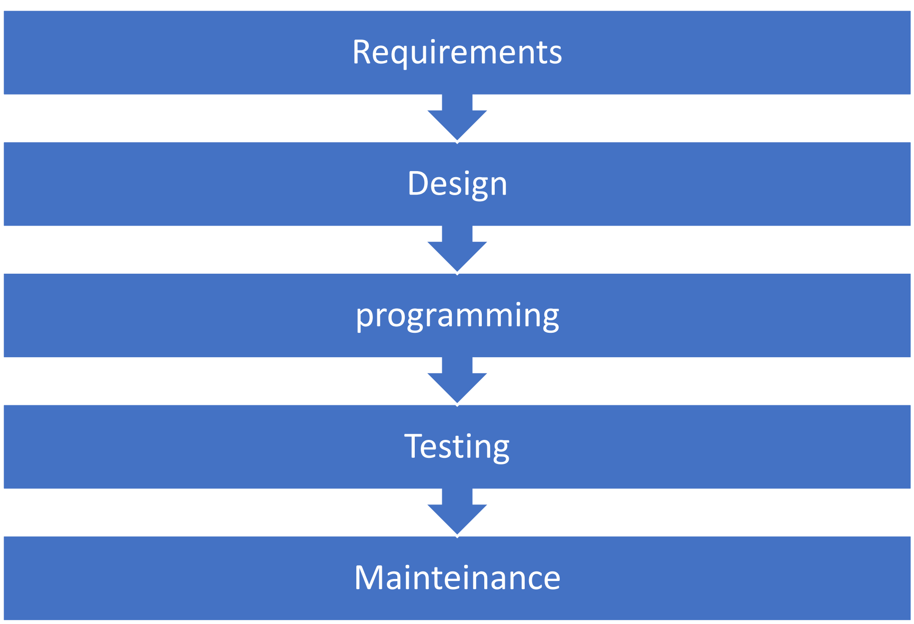

Figure 8:work flow

3.1. Smart Home System Design

This project suggests a setup for a smart home system utilizing the ESP32 microcontroller, built around Internet of Things (IoT) architecture. The current system configuration is tailored to fulfill user requirements by offering fundamental smart home functionalities while cost-efficient due to its minimal intricacy. The design had several process which are shown below from planning to the integration part of the smart home prototype.

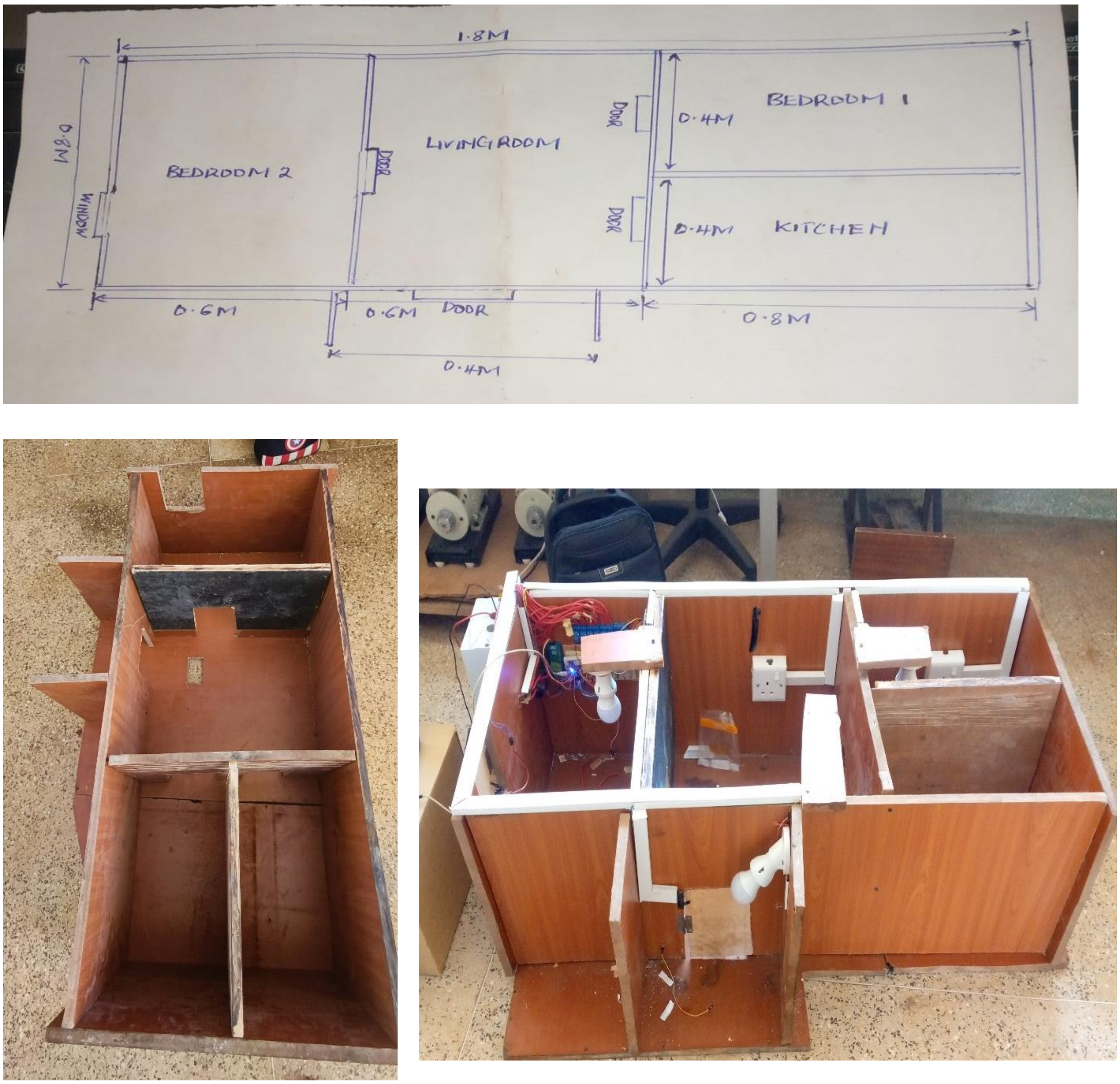

Figure 9: Smart Home System Design

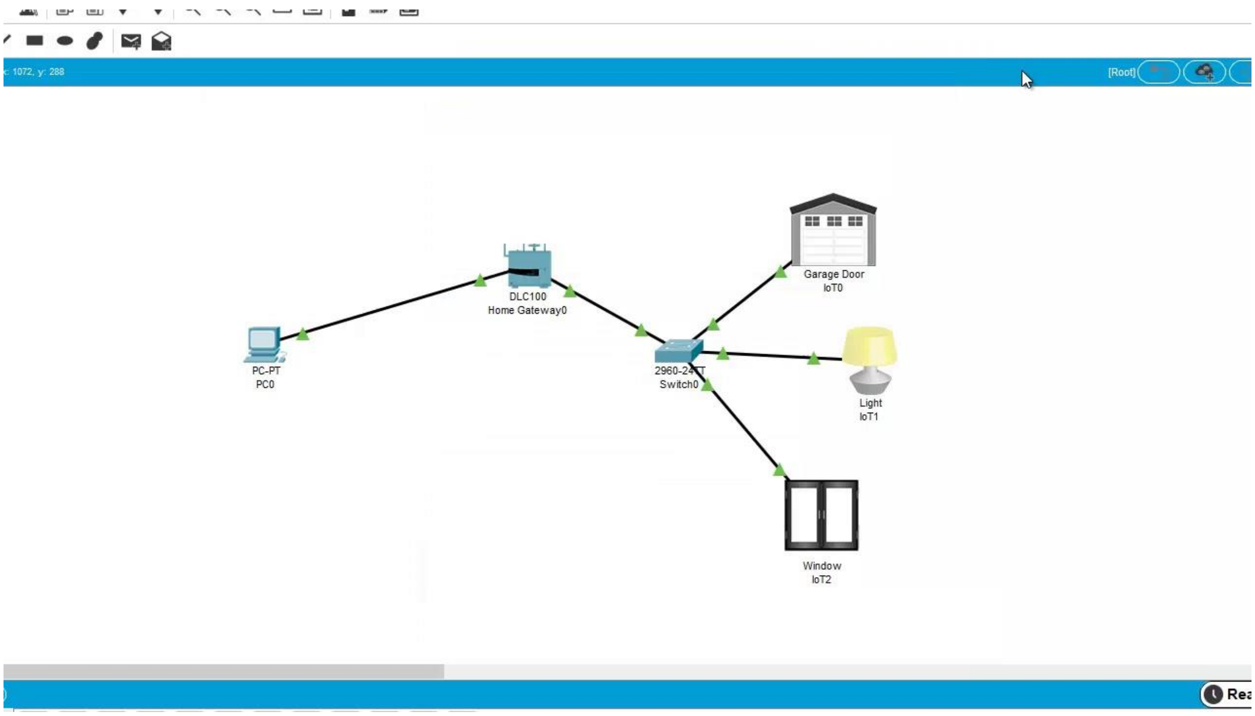

Figure 10: Smart Home System Design

The project section is divided into six working units;

1. The power source section comprises of electrical energy. The safety- equipped power source provides the necessary electrical energy for the operational devices.
2. The relay module: The appliances are connected to the relay module using wiring. The power line commonly used is linked to the standard (COM) port of the relay, while the power line of the appliance is attached to the normally open (NO) port. Upon activation of the relay, the circuit is fully connected, enabling the passage of electric current to the appliance.
3. ESP32 microcontroller: The ESP32 is programmed to regulate the relay module. This entails configuring the general purpose input/output (GPIO) pins as output channels and developing functions to manage the activation and deactivation signals for the relays.

4. Mobile Application: The ESP32 microcontroller will be connected to the Telegram messaging app to create a user-friendly control system for this smart home automation project. Telegram provides a safe platform that allows users to transmit commands via a chatbot interface, which are subsequently accepted by the ESP32 system. When the ESP32 receives an instruction, it triggers the corresponding relay on the four relay modules thereby managing the linked household appliances. By using Telegram's extensive availability and user- friendly interface, this integration enables users to control their smart home equipment remotely using straightforward text based instructions. This functionality simplifies smart home management to the point where it is as effortless as sending a message.

5. Automation: The door and window automation system using an ESP32 and Telegram bot works by allowing users to send commands via Telegram to open or close doors and windows remotely. The ESP32 connects to WiFi and listens for messages from the Telegram bot. When a user sends a command (e.g., /door1_open), the ESP32 processes it and activates the corresponding servo motor to move the door or window to the desired position. It then sends a confirmation message back to the user, indicating the new status. Additionally, users can check the current state of all doors and windows by sending the /status command. This system enhances convenience and security, enabling remote home automation via the internet.

6. PZEM-004T is a digital energy meter used to measure voltage, current, power, energy, and frequency in AC electrical systems. It communicates with a microcontroller (ESP32) via UART (RX/TX pins) at a baud rate of 9600. The ESP32 sends a request to the PZEM-004T, which responds with real-time electrical readings. The meter is connected to the live and neutral AC lines for voltage measurement, while a current transformer (CT) sensor clamps around the live wire to measure current. The ESP32 processes the received data and can display it on an

LCD or send it via Telegram. This setup allows remote monitoring of power consumption, making it useful for smart energy management systems.

3.2. Electrical and Connectivity Design

The development of the smart home system in this project aims to provide benefits for users to monitor and control their electrical equipment remotely using an internet network by the command of the Telegram bot.

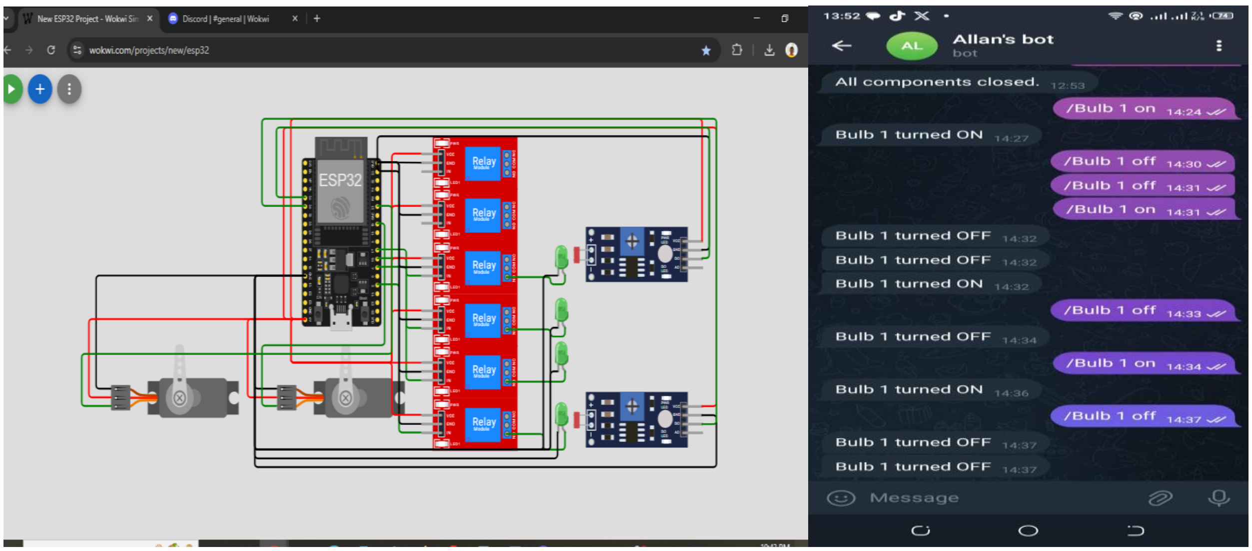

Figure 11: Electrical Connectivity of Smart Home

The above figure shows that the electrical installation of a smart home system integrated with ESP32 is only a light load and a socket. Monitoring the socket to see electrical equipment that functions based on the current flowing through it is necessary. The data received is transmitted to the user's Telegram account to display the current status of the home equipment.

Alternatively, the user can issue commands through a telegram to remotely operate the ESP32 via the Internet and manage the operation or disabling of electrical devices.

3.4. Automation connectivity

Door and window automation system using an ESP32 and Telegram bot works by allowing users to send commands via Telegram to open or close doors and windows remotely. The ESP32 connects to WiFi and listens for messages from the Telegram bot. When a user sends a command (e.g., /door1_open), the ESP32 processes it and activates the corresponding servo motor to move the door or window to the desired position. It then sends a confirmation message back to the user, indicating the new status. Additionally, users can check the current state of all doors and windows by sending the /status command. This system enhances convenience and security, enabling remote home automation via the internet.

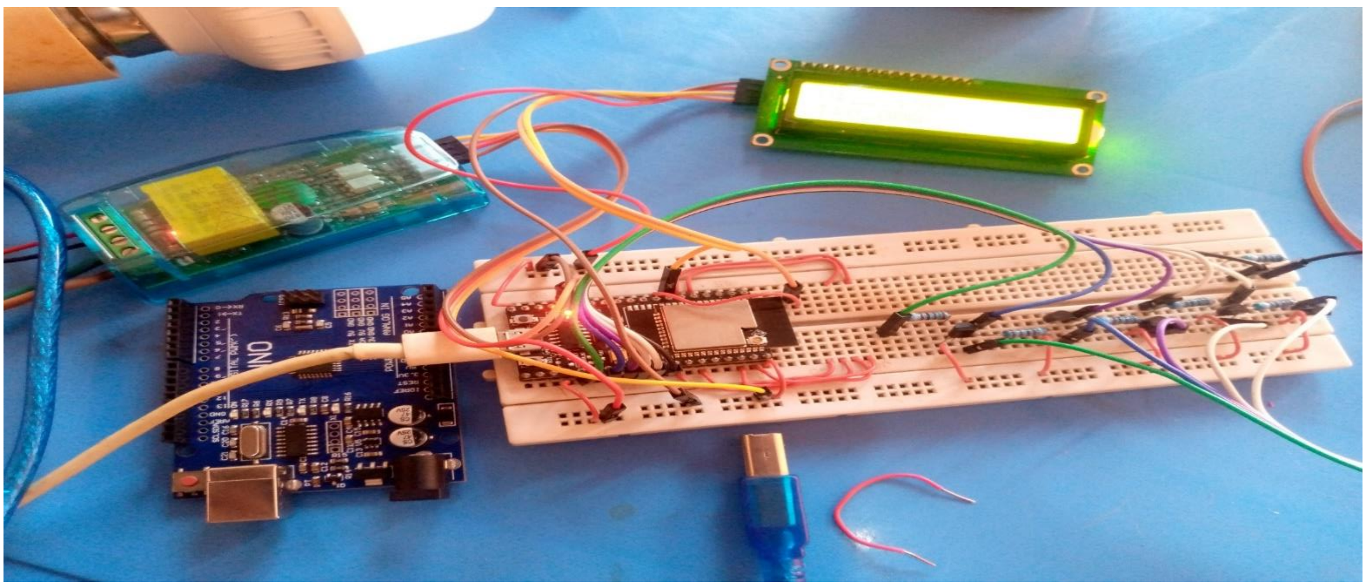

Figure 12: Automation connectivity

3.5. Smart Home System Module Workflow

In our Smart Home System (SHS) prototype, we have utilized the widely recognized Arduino Integrated Development Environment (IDE) to program the ESP32 microcontroller. The Arduino IDE is widely recognized for its simple interface and compatibility with Arduino-compatible boards. It is an ideal choice for our project because it combines ease of use and functionality. We have chosen to use C/C++ as our programming language due to its exceptional efficiency and precise management of hardware resources. These qualities are crucial in the field of embedded systems development.

Our Smart Home System's (SHS) primary characteristic is the ability to monitor and standardize electrical appliances and automation remotely through the Internet. This fundamental characteristic not only determines the ease of use of our system but also its significance in the present context of intelligent home solutions. In order to connect the user's intentions with the system's actions, we have incorporated the Telegram chat platform. This integration enables a smooth and protected transfer of user instructions to the ESP32 module. The ESP32 then interprets these instructions and performs corresponding physical actions using the eight relay modules, servos for automation effectively managing the domestic appliances.

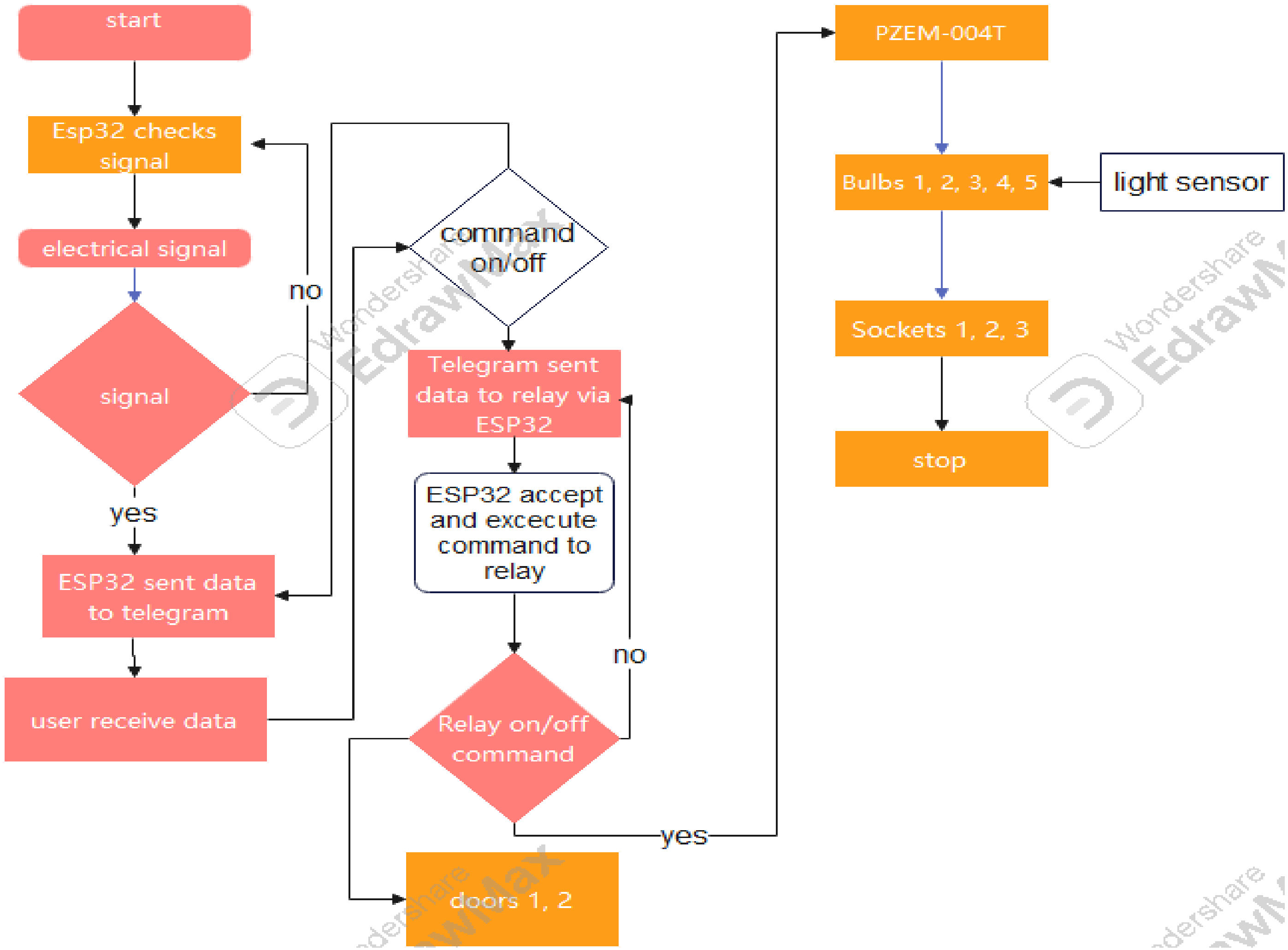

Figure 13: flowchart.

4.0. CHAPTER FOUR

4.1. Result and Discussion

4.1.1. System Performance and Testing

The main objective is To create and install home automation system using IOT switches and doors/windows controlled from a remote by ESP32 module via Telegram link. Home automation system using the ESP32 module and Telegram bot successfully enables remote control of lights, sockets, doors, and windows via IoT-based commands. Users can send specific Telegram commands to operate relays for lights and sockets or servo motors for doors and windows. The system provides real-time feedback, confirming the status of each device. The integration of photoresistors for bulb status detection ensures accurate feedback, enhancing system reliability. The use of WiFi connectivity allows seamless remote operation, making home automation more accessible and efficient. This implementation improves convenience, security, and energy management by allowing users to control and monitor their home devices from anywhere.

Another objective was to measure power consumption using PZEM004T meter. PZEM-004T energy meter successfully measures and displays real-time voltage, current, power, energy, and frequency data on an LCD screen and via Telegram notifications. The ESP32 communicates with the PZEM-004T via UART, fetching power consumption data and providing instant updates. This feature allows users to monitor their energy usage remotely, helping them make informed decisions to optimize power consumption and reduce electricity costs. By integrating IoT-based monitoring with home automation, the system not only enhances control over electrical appliances but also promotes efficient energy management, making it an ideal solution for smart homes.

The performance of the smart home system prototype was assessed by evaluating its response time across different internet speeds, in addition to its functionality. The test sought to measure the latency encountered by the system in carrying out commands, taking into account the internet speed of the provider. The findings suggest that the smart home system prototype consistently maintains a satisfactory reaction time, while there are fluctuations that correspond to the speed of the internet connection. This analysis is crucial for comprehending the system's ability to adapt to various network conditions and guaranteeing a suitable user experience.

Wi-Fi operator	**Delay** **Time(morning)**	**Delay** **Time(afternoon)**	**Delay** **Time(evening)**
Michael provider	5	4	6
Marto net	6	6	5
Electrical lab	7	6	7
kilele	6	5	6
IOT	3	1	2

Figure 14: various network connectivity

Based on the above, it is known that various internet providers available at the test location found different delays in the morning, afternoon, and evening. The average delay time from all operators is 5.3 seconds in the morning, 4.2 seconds in the afternoon, and 5.1 seconds in the afternoon. The highest delay occurs for 7s, and the fastest is 1s. These three times are times when the user is likely away from home.

5.0 CHAPTER FIVE

Project Time Plan

In this project plan, we present a strategic blueprint for the development of an innovative smart home automation system for a period of six months. Our objective is to revolutionize residential living by seamlessly integrating IoT technology, automation and user-friendly interfaces. The overarching vision is to establish a dynamic and intelligent environment where homeowners can effortlessly control household appliances and monitor living spaces remotely.

ACTIVITY	NOV	DEC	JAN	FEB	MAR	APR
DOCUMENTATION						
PROPOSAL WRITING						
RESEARCH						
MINI PRESENTATION						
HARDWARE TESTING						
SOFTWARE DESIGN						
TESTING						
FINAL PRESENTATION						

Figure 15: project plan

6.0 CHAPTER SIX

Budget

The product produced in this study is an IOT smart home system prototype developed with the ESP32 module. The number of loads being monitored consists of 4 lights and 2 sockets. Details of the costs required to develop this system are shown in the table below. The costs referred to are only for making the system and do not include plywood for the house and the tools used. The total price is estimated at Ksh. 9,630.

Material	Unit price	Amount	Total
ESP32	2000	1pc	2000
8Channel Relay	600	1pc	600
Transistor	50	3pc	150
Jumper wires	150	4pc	600
Bulbs	100	5pc	500
ConnectingWires	40	20 meters	800
Sockets	200	3pc	600
Circuit breaker	150	2pc	300
Bulb Holders	50	5	250
MCB	300	1pc	300
PZEM-004T	2000	1pc	2000
LCD display	350	1pc	350
Photoresistor	50	5pc	250
Servo motors	550	3pc	1650
Resistor	10	8pc	80
			9,630

Figure 16: budget

Conclusion

The integration of IoT technology into homes has become a reality, transforming how we engage with our living spaces. The smart home concept involves technology seamlessly integrating with daily activities, promoting efficiency, control, and adaptability. An advanced smart home automation system uses IoT switches connected to the ESP32 module for remote control capabilities. This system overcomes constraints associated with conventional methods by enabling remote management of household equipment. The ESP32 module's wireless functionalities ensure a dependable connection, and a user-friendly interface such as the use of Telegram app which allows homeowners to control appliances from anywhere with an internet connection.

REFERENCES

[1] R. Piyare and M. Tazil, "IoT based home automation systems: Design and development," *Int. J. Comput. Appl.*, vol. 10, no. 1, pp. 20-24, 2015.

[2] J. Lee et al., "ESP32: The ultimate IoT platform for smart homes," *Int. Conf. Ind. Electron. Appl.*, pp. 23-28, 2017.

[3] A. Singh and M. Kapoor, "Relay modules for home automation," *Proc. Int. Conf. Electr. Comput. Eng.*, pp. 67-70, 2018.

[4] Wi-Fi Alliance, "Wi-Fi technology in smart homes," *Wi-Fi Alliance Reports*, vol. 7, no. 5, pp. 50-55, 2020.

[5] S. Shrestha and S. Silpakar, "Telegram-based smart home automation system," *J. Electr. Eng. Technol.*, vol. 12, no. 6, pp. 220-225, 2017.

[6] A. Al-Ali et al., "Light sensors for home automation systems," *J. Smart Homes Autom.*, vol. 9, no. 2, pp. 31-38, 2017.

[7] S. Kumar and P. Sharma, "Energy management in smart homes using IoT," *Int. J. Renew. Energy Eng.*, vol. 14, no. 3, pp. 100-107, 2019.

[8] K. Patel and R. Shah, "Servo motors for automation in IoT-based smart homes," *J. Robotics Autom.*, vol. 5, no. 1, pp. 45-50, 2019.

Appendix

Arduino IDE code

IOT_AUTOMATION_ALLAN_MARTIN_MICHAEL | Arduino IDE 2.3.4

File Edit Sketch Tools Help

Select Board

IOT_AUTOMATION_ALLAN_MARTIN_MICHAEL.ino

```cpp
#include <Arduino.h>
#include <ESP32Servo.h>
#include <WiFi.h>
#include <UniversalTelegramBot.h>
#include <WiFiClientSecure.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <PZEM004Tv30.h>

// WiFi Credentials
const char* ssid = "IOT";
const char* password = "12345678";

// Telegram Bot Token
const char* botToken = "7626283325:AAFPXefMPJPHrVlu2mh2FTvobKhXKYf4Xp8";

WiFiClientSecure client;
UniversalTelegramBot bot(botToken, client);

// Relay Pins & States
const int relayPins[] = {2, 4, 16, 17, 5, 18};
const char* relayNames[] = {"Bulb 1", "Bulb 2", "Bulb 3", "Bulb 4", "Socket 1", "Socket 2"};
bool relayStates[] = {false, false, false, false, false, false};

// Servo Definitions
Servo servo1, servo2, servo3, servo4;
#define SERVO_1_PIN 19  // Door 1
#define SERVO_2_PIN 21  // Lock 1
#define SERVO_3_PIN 22  // Window 1
#define SERVO_4_PIN 23  // Lock 2

```

Ln 1, Col 1 × No board selected

IOT_AUTOMATION_ALLAN_MARTIN_MICHAEL | Arduino IDE 2.3.4

File Edit Sketch Tools Help

Select Board

IOT_AUTOMATION_ALLAN_MARTIN_MICHAEL.ino

```cpp
// Photoresistor Pins
#define PHOTO_1_PIN 34  // Bulb 3 Sensor
#define PHOTO_2_PIN 35  // Bulb 4 Sensor

// LCD Display
#define I2C_ADDR 0x3F
#define LCD_COLUMNS 16
#define LCD_ROWS 2
#define SDA_PIN 32
#define SCL_PIN 33
LiquidCrystal_I2C lcd(I2C_ADDR, LCD_COLUMNS, LCD_ROWS);

// PZEM-004T Energy Meter
PZEM004Tv30 pzem(Serial2, 13, 15);

void gradualMove(Servo &servo, int startPos, int endPos, int durationMs) {
    int stepCount = abs(endPos - startPos);
    int stepDelay = durationMs / stepCount;

    if (startPos < endPos) {
        for (int pos = startPos; pos <= endPos; pos++) {
            servo.write(pos);
            delay(stepDelay);
        }
    } else {
        for (int pos = startPos; pos >= endPos; pos--) {
            servo.write(pos);
            delay(stepDelay);
        }
    }
}
```

Ln 1, Col 1 × No board selected

IOT_AUTOMATION_ALLAN_MARTIN_MICHAEL | Arduino IDE 2.3.4

File Edit Sketch Tools Help

Select Board

IOT_AUTOMATION_ALLAN_MARTIN_MICHAEL.ino

```cpp

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  client.setInsecure();
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi");

  // Initialize relays
  for (int i = 0; i < 6; i++) {
    pinMode(relayPins[i], OUTPUT);
    digitalWrite(relayPins[i], HIGH);
  }

  // Attach servos & set to default positions
  servo1.attach(SERVO_1_PIN);
  servo2.attach(SERVO_2_PIN);
  servo3.attach(SERVO_3_PIN);
  servo4.attach(SERVO_4_PIN);
  servo1.write(0);
  servo2.write(0);
  servo3.write(0);
  servo4.write(0);

  // Initialize LCD
  Wire.begin(SDA_PIN, SCL_PIN);
  lcd.init();
  lcd.backlight();
```

Ln 1, Col 1 × No board selected

IOT_AUTOMATION_ALLAN_MARTIN_MICHAEL | Arduino IDE 2.3.4

File Edit Sketch Tools Help

Select Board

IOT_AUTOMATION_ALLAN_MARTIN_MICHAEL.ino

```cpp
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("PZEM Meter Init");
  Serial.println("PZEM-004T Energy Meter Test on ESP32");
}

void handleTelegramMessages() {
  int messageCount = bot.getUpdates(bot.last_message_received + 1);
  while (messageCount) {
    for (int i = 0; i < messageCount; i++) {
      String chat_id = bot.messages[i].chat_id;
      String text = bot.messages[i].text;
      bool commandFound = false;

      Serial.println("Received command: " + text);

      for (int j = 0; j < 6; j++) {
        if (text.equalsIgnoreCase("/" + String(relayNames[j]) + " on")) {
          digitalWrite(relayPins[j], LOW);
          relayStates[j] = true;
          bot.sendMessage(chat_id, String(relayNames[j]) + " turned ON", "");
          commandFound = true;
        } else if (text.equalsIgnoreCase("/" + String(relayNames[j]) + " off")) {
          digitalWrite(relayPins[j], HIGH);
          relayStates[j] = false;
          bot.sendMessage(chat_id, String(relayNames[j]) + " turned OFF", "");
          commandFound = true;
        }
      }

      if (text.equalsIgnoreCase("/status")) {
```

Ln 1, Col 1 × No board selected


```
                    relayStates[j] = true;
                    bot.sendMessage(chat_id, String(relayNames[j]) + " turned ON", "");
                    commandFound = true;
                } else if (text.equalsIgnoreCase("/" + String(relayNames[j]) + " off")) {
                    digitalWrite(relayPins[j], HIGH);
                    relayStates[j] = false;
                    bot.sendMessage(chat_id, String(relayNames[j]) + " turned OFF", "");
                    commandFound = true;
                }
            }

            if (text.equalsIgnoreCase("/status")) {
                String statusMsg = "Current Status:\n";
                for (int k = 0; k < 6; k++) {
                    statusMsg += String(relayNames[k]) + ": " + (relayStates[k] ? "ON" : "OFF") + "\n";
                }
                statusMsg += "Photoresistor 1: " + String(analogRead(PHOTO_1_PIN)) + "\n";
                statusMsg += "Photoresistor 2: " + String(analogRead(PHOTO_2_PIN)) + "\n";
                bot.sendMessage(chat_id, statusMsg, "");
                commandFound = true;
            }
        }
        messageCount = bot.getUpdates(bot.last_message_received + 1);
    }
}

void loop() {
    handleTelegramMessages();
    delay(1000);
}
```

Datasheet for ESP32 Module

2.1 Pin Layout

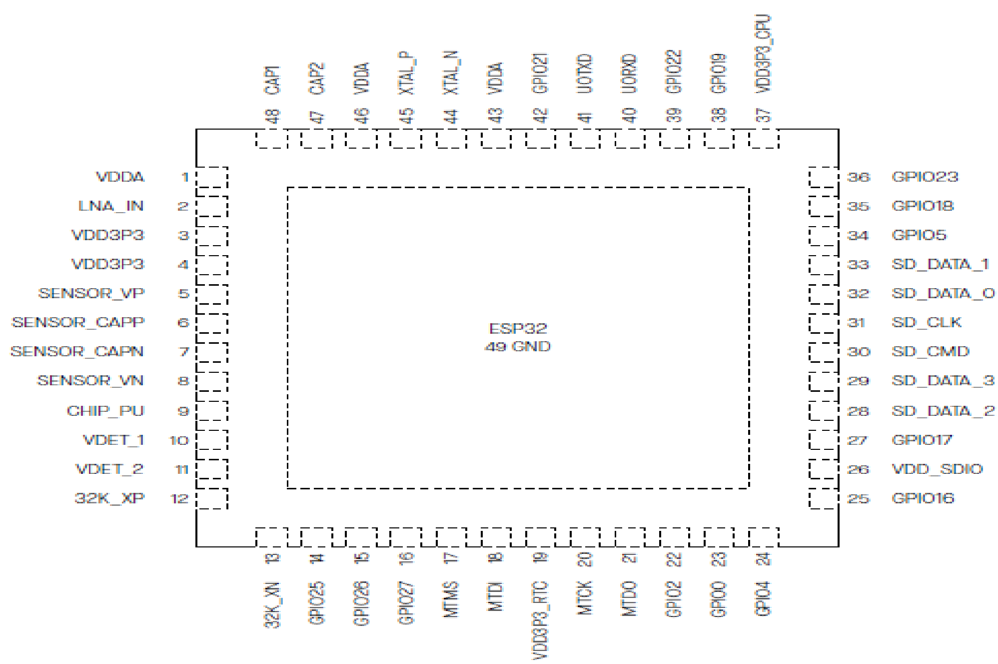

Figure 2-1. ESP32 Pin Layout (QFN 6*6, Top View)

https://www.espressif.com/sites/default/files/documentation/esp32_datashe et_en.pdf

Table 2-1. Pin Overview

Name	No.	Type	Function								
Analog											
VDDA	1	P	Analog power supply (2.3 V – 3.6 V)								
LNA_IN	2	I/O	RF input and output								
VDD3P3	3	P	Analog power supply (2.3 V – 3.6 V)								
VDD3P3	4	P	Analog power supply (2.3 V – 3.6 V)								
VDD3P3_RTC											
SENSOR_VP	5	I	GPIO36,	ADC1_CH0,	RTC_GPIO0						
SENSOR_CAPP	6	I	GPIO37,	ADC1_CH1,	RTC_GPIO1						
SENSOR_CAPN	7	I	GPIO38,	ADC1_CH2,	RTC_GPIO2						
SENSOR_VN	8	I	GPIO39,	ADC1_CH3,	RTC_GPIO3						
CHIP_PU	9	I	High: On; enables the chip Low: Off; the chip shuts down Note: Do not leave the CHIP_PU pin floating.								
VDET_1	10	I	GPIO34,	ADC1_CH6,	RTC_GPIO4						
VDET_2	11	I	GPIO35,	ADC1_CH7,	RTC_GPIO5						
32K_XP	12	I/O	GPIO32,	ADC1_CH4,	RTC_GPIO9,	TOUCH9,	32K_XP (32.768 kHz crystal oscillator input)				
32K_XN	13	I/O	GPIO33,	ADC1_CH5,	RTC_GPIO8,	TOUCH8,	32K_XN (32.768 kHz crystal oscillator output)				
GPIO25	14	I/O	GPIO25,	ADC2_CH8,	RTC_GPIO6,	DAC_1,	EMAC_RXD0				
GPIO26	15	I/O	GPIO26,	ADC2_CH9,	RTC_GPIO7,	DAC_2,	EMAC_RXD1				
GPIO27	16	I/O	GPIO27,	ADC2_CH7,	RTC_GPIO17,	TOUCH7,	EMAC_RX_DV				
MTMS	17	I/O	GPIO14,	ADC2_CH6,	RTC_GPIO16,	TOUCH6,	EMAC_TXD2,	HSPICLK,	HS2_CLK,	SD_CLK,	MTMS
MTDI	18	I/O	GPIO12,	ADC2_CH5,	RTC_GPIO15,	TOUCH5,	EMAC_TXD3,	HSPIQ,	HS2_DATA2,	SD_DATA2,	MTDI
VDD3P3_RTC	19	P	Input power supply for RTC IO (2.3 V – 3.6 V)								
MTCK	20	I/O	GPIO13,	ADC2_CH4,	RTC_GPIO14,	TOUCH4,	EMAC_RX_ER,	HSPID,	HS2_DATA3,	SD_DATA3,	MTCK
MTDO	21	I/O	GPIO15,	ADC2_CH3,	RTC_GPIO13,	TOUCH3,	EMAC_RXD3,	HSPICS0,	HS2_CMD,	SD_CMD,	MTDO
GPIO2	22	I/O	GPIO2,	ADC2_CH2,	RTC_GPIO12,	TOUCH2,		HSPIWP,	HS2_DATA0,	SD_DATA0	
GPIO0	23	I/O	GPIO0,	ADC2_CH1,	RTC_GPIO11,	TOUCH1,	EMAC_TX_CLK,	CLK_OUT1,			
GPIO4	24	I/O	GPIO4,	ADC2_CH0,	RTC_GPIO10,	TOUCH0,	EMAC_TX_ER,	HSPIHD,	HS2_DATA1,	SD_DATA1	
VDD_SDIO											
GPIO16	25	I/O	GPIO16,	HS1_DATA4,	U2RXD,	EMAC_CLK_OUT					
VDD_SDIO	26	P	Output power supply: 1.8 V or the same voltage as VDD3P3_RTC								
GPIO17	27	I/O	GPIO17,	HS1_DATA5,	U2TXD,	EMAC_CLK_OUT_180					
SD_DATA_2	28	I/O	GPIO9,	HS1_DATA2,	U1RXD,	SD_DATA2,	SPIHD				
SD_DATA_3	29	I/O	GPIO10,	HS1_DATA3,	U1TXD,	SD_DATA3,	SPIWP				
SD_CMD	30	I/O	GPIO11,	HS1_CMD,	U1RTS,	SD_CMD,	SPICS0				
SD_CLK	31	I/O	GPIO6,	HS1_CLK,	U1CTS,	SD_CLK,	SPICLK				
SD_DATA_0	32	I/O	GPIO7,	HS1_DATA0,	U2RTS,	SD_DATA0,	SPIQ				
SD_DATA_1	33	I/O	GPIO8,	HS1_DATA1,	U2CTS,	SD_DATA1,	SPID				
VDD3P3_CPU											
GPIO5	34	I/O	GPIO5,	HS1_DATA6,	VSPICS0,	EMAC_RX_CLK					
GPIO18	35	I/O	GPIO18,	HS1_DATA7,	VSPICLK						
GPIO23	36	I/O	GPIO23,	HS1_STROBE,	VSPID						
VDD3P3_CPU	37	P	Input power supply for CPU IO (1.8 V – 3.6 V)								
GPIO19	38	I/O	GPIO19,	U0CTS,	VSPIQ,	EMAC_TXD0					
GPIO22	39	I/O	GPIO22,	U0RTS,	VSPIWP,	EMAC_TXD1					
U0RXD	40	I/O	GPIO3,	U0RXD,	CLK_OUT2						
U0TXD	41	I/O	GPIO1,	U0TXD,	CLK_OUT3,	EMAC_RXD2					
GPIO21	42	I/O	GPIO21,		VSPIHD,	EMAC_TX_EN					
Analog											
VDDA	43	P	Analog power supply (2.3 V – 3.6 V)								
XTAL_N	44	O	External crystal output								
XTAL_P	45	I	External crystal input								
VDDA	46	P	Analog power supply (2.3 V – 3.6 V)								
CAP2	47	I	Connects to a 3.3 nF (10%) capacitor and 20 kΩ resistor in parallel to CAP1								
CAP1	48	I	Connects to a 10 nF series capacitor to ground								
GND	49	P	Ground								