BLACKFOX
GROUP

9/1, Rathinam Kannusamy Valagam, 3rd
Floor - 3rd Room, Agilmedu, 6th St, Erode,
Tamil Nadu 638001

INTERNSHIP REPORT

NAME: SRIDHAR M

REGISTER NUMBER: 513121106093

In partial fulfilment for the award of the degree
of
BACHELOR OF ENGINEERING
in
ELECTRONICS AND COMMUNICATION ENGINEERING
THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY,
BAGAYAM,VELLORE-632002.

THANTHAI PERIYAR GOVERNMENT INSTITUTE OF TECHNOLOGY

BAGAYAM, VELLORE-632 002.

DEGREE IN ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the internship report titled **“FIRWARE TRAINEE”** is the bonafide record of work done by **SRIDHAR M (Reg no: 513121106093)** in partial fulfillment of the requirement of the degree in Electronics and Communication Engineering during the year 2024 -2025.

Internship completed for 100 days from **20.05.2024 to 30.08.2024.**

Internship Guide
Principal

RGMTTC, Chennai-16.

RGMTTC, Chennai-16.

Submitted for the Industrial Training Report and Evaluation held on

____

ACKNOWLEDGEMENT

We feel glad to take this opportunity to cordially acknowledge a number of people who provided us a great support during our internship.

We would like to express our deep sense of gratitude to our respected Principal **Dr.P.K. PALANI, B.E (Hons), M.E., Ph.D**., who has bestowed his kind grace and affection of us in accomplishing this internship.

Sincere thanks to our **prof**. **Dr.S.LETITIA, M.E., Ph.D.,** Head of the Department of Electronics and Communication Engineering, Vellore on providing facilities to attend the internship in Enterprise Data Networking, Chennai and utilize all the facilities in this college.

We would like to thanks our faculty adviser, **Dr..Sundaravanan J**, **M.E,Ph.D.,** Assistant Professor, Department of Electronics and Communication Engineering, who guided us throughout the entire phase of this internship. It is his motivation and guidance which made us explore this internship.

We would also like to express our profound gratitude to **Shri.K.S.SILAMPOLI, CEO, BLACKFOX**. It is our opportunity to express gratitude and sincere thanks to our guide **Mr. YOGANATHAN, CTO,BLACKFOX,** Internship coordinator in BLACKFOX who gave us the motivation and guided us to do this internship in **"FIRMWARE TRAINEE"**. We are also thankful to all the faculty members and laboratory assistants in BLACKFOX for their constructive guidance and encouragement.

Finally, we thank our parents, teaching, non-teaching staff and friends who helped us in completing this internship successfully.

TABLE OF CONTENTS

CHAPTER – 1

OVERVIEW OF FIRMWARE

1.1 WHAT IS FIRMWARE?

- Firmware is specialized software programmed into hardware devices. Here are three key points:
- **Permanent Software:** Firmware is software that provides low-level control for a device's specific hardware. It is stored in non-volatile memory (like ROM, EEPROM, or flash memory) and remains even when the device is powered off.
- **Essential for Hardware Functionality:** Firmware enables the hardware to function by providing necessary instructions and control processes. Without firmware, most devices, such as microcontrollers, routers, and even simple appliances, wouldn't operate.
- **Updateable:** Though firmware is embedded, it can often be updated to improve functionality, fix bugs, or provide new features, usually through special firmware update processes.

1.2 EMBEDDED SYSTEM AND IOT

Firmware is fundamental to both embedded systems and IoT devices because it bridges the gap between hardware and higher-level software, enabling devices to operate seamlessly in their environments. In embedded systems, firmware not only controls the hardware but also manages timing, sensor readings, and communication with other parts of the system. Since embedded systems are often resource-constrained, firmware must be optimized for power efficiency and real-time performance.

For IoT devices, firmware goes beyond basic hardware control by handling the complexities of network communication protocols (like Wi-Fi, Bluetooth, or Zigbee) and ensuring secure data transfer. It also allows devices to communicate with cloud platforms or local servers for tasks such as remote monitoring, control, or analytics. As IoT devices are often deployed in large numbers, firmware must be robust and secure, with features such as Over-the-Air (OTA) updates to ensure that devices can be patched or upgraded remotely.

In both areas, firmware is also critical for ensuring device security, which is a growing concern as more devices become connected to networks. Firmware vulnerabilities can leave embedded and IoT devices exposed to cyber threats, making secure firmware design and regular updates essential for maintaining the integrity of systems.

1.3 **CAREER IN FIRMWARE**

A career in firmware development is a dynamic and specialized field, offering a range of opportunities to work with cutting-edge technology across various industries. Firmware engineers create the low-level code that directly controls hardware in devices like computers, IoT systems, medical equipment, automobiles, and more. It requires a deep understanding of hardware architecture, embedded systems, and real-time operating systems.

Key Aspects of a Firmware Career:

- **Diverse Industry Applications:** Firmware developers can work in industries like consumer electronics, automotive, aerospace, telecommunications, and healthcare. For instance, they might design firmware for microcontrollers in wearables, smart home devices, or industrial machines.

- **Skills and Tools:** A strong background in computer science or electrical engineering is usually essential. Key skills include proficiency in programming languages like C/C++, assembly language, understanding of hardware components, and familiarity with development tools like JTAG debuggers and oscilloscopes.
- **Firmware Specializations:** You can specialize in different areas, such as low-power firmware for battery-operated devices, real-time systems, IoT firmware for connected devices, or security-focused firmware for encrypted communications.
- **Growing Demand:** As IoT and smart devices continue to proliferate, the demand for skilled firmware developers is growing rapidly. From developing firmware for microcontrollers in consumer gadgets to building critical systems in autonomous vehicles, the opportunities are vast.
- **Constant Learning:** The field is continually evolving with advancements in hardware and communication technologies. Firmware engineers must stay updated with the latest trends in embedded systems, communication protocols, security practices, and power management techniques.

A career in firmware development is not only technically challenging but also offers the chance to work on transformative technologies that shape the future of everyday devices.

CHAPTER -2

COMPANY PROFILE

2.1 BLACKFOX– EMBEDDED AND SOLUTION

- Blackfox Embedded Solutions, founded in 2020, is a growing company based in Erode, Tamil Nadu. Specializing in embedded systems and electronics development, they offer a variety of services, including PCB design, assembly, soldering, and testing. They work as manufacturers, service providers, and suppliers for electronic product development. Blackfox plays a crucial role in embedded system prototyping, focusing on sectors like IoT and industrial automation, helping clients turn their concepts into functional electronic products.
- The company operates from a relatively small base, with a CEO named Mr. Silampoli KS, and has established itself as an emerging player in the embedded solutions market. Their expertise lies in developing customized hardware that meets specific client needs, particularly in areas where innovation and electronic design are critical. This could range from IoT devices, smart products, to industrial machinery requiring embedded controls.
- Blackfox also provides PCB-related services, including designing complex circuits, assembling components, and performing rigorous soldering and testing to ensure the final product meets industry standards. As an embedded solutions company, they assist in the development of electronic devices by working closely with clients to refine their prototypes and bring their designs to life.
- Their service is particularly valuable for businesses looking to develop custom embedded systems without having to handle the

intricate and technical aspects of electronics production. By providing a full suite of services, from PCB design to final testing, Blackfox Embedded Solutions acts as a one-stop-shop for companies looking to outsource their hardware development.

- ➢ In addition to their primary focus on PCBs and embedded systems, Blackfox helps companies prepare their products for mass production by testing the functionality and durability of the devices they develop. Their offerings also allow for ongoing development and adjustments, such as modifications to improve product performance or add new features.

- ➢ By specializing in the growing fields of IoT and industrial automation, Blackfox Embedded Solutions is positioning itself as a significant player in the electronics industry, serving clients who are increasingly looking for smart, connected solutions. Their ability to handle everything from the initial design to final testing makes them a valuable partner for businesses that need embedded systems to operate their products effectively.

- ➢ Operating from their location in Erode, the company continues to expand its reach and offerings, taking on more complex and diverse projects across various sectors. Their commitment to quality, innovation, and customer satisfaction has allowed them to maintain steady growth in a competitive industry.

- ➢ With a modest annual turnover of Rs 12 Lakhs and a small but capable team, Blackfox has laid the foundation for continued success in embedded systems development. Their expertise in areas like IoT, coupled with their specialized knowledge in electronics, ensures that they remain relevant in a rapidly evolving technological landscape.

2.2 VISION OF THE INSTITUTE

To be the foremost innovator and trusted partner in embedded systems and electronics development, empowering businesses worldwide with cutting-edge solutions in IoT, industrial automation, and beyond. We aim to drive technological advancement through exceptional design, meticulous quality, and unwavering customer commitment, transforming visionary concepts into impactful, functional products."

2.3 MISSION OF THE INSTITUTE

- ➢ Deliver exceptional embedded systems and electronics solutions.
- ➢ Leverage expertise in PCB design, assembly, and testing.
- ➢ Meet unique client needs through personalized service and innovative design.
- ➢ Ensure rigorous quality assurance and continuous improvement.
- ➢ Support clients in achieving their goals and success in the digital landscape.

2.4 QUALITY OBJECTIVES

Training Quality

- ✧ **MQTT:** Comprehensive training on Message Queuing Telemetry Transport, including protocol basics and application in IoT.
- ✧ · **IoT:** In-depth instruction on Internet of Things concepts, devices, and integration strategies.
- ✧ · **Arduino:** Hands-on training in Arduino development, including coding, interfacing, and project implementation.
- ✧ · **PIC:** Detailed training on PIC microcontrollers, covering programming, configuration, and application.

- ✧ · **ESP32:** Advanced training on ESP32 development, focusing on features, connectivity, and practical applications.
- ✧ · **NodeMCU:** Practical training in using NodeMCU for IoT projects, including setup, programming, and deployment.
- ✧ · **GSM:** Training on GSM module integration and communication, including SMS, voice calls, and data handling.
- ✧ · **Sinric Pro:** Instruction on using Sinric Pro for IoT device control and integration with smart home solutions.

2.5 RESEARCH AND DEVELOPMENT

At Blackfox Embedded Solutions, our Research and Development efforts are driven by a commitment to innovation and excellence in embedded systems and electronics. We focus on exploring and developing cutting-edge technologies to create advanced solutions that meet the evolving needs of our clients. Our R&D activities encompass prototyping and rigorous testing to ensure that our products meet the highest industry standards. We are dedicated to advancing the fields of IoT and industrial automation, working on custom hardware solutions that are tailored to specific client requirements. Through continuous improvement and collaboration with industry experts and technology partners, we strive to stay at the forefront of technological advancements and deliver exceptional products that enhance performance, reliability, and functionality.

CHAPTER – 3

DEPARTMENTS IN BLACKFOX

3.1 RESEARCH AND DEVELOPMENT (R&D):

The R&D department at Blackfox Embedded Solutions is dedicated to pushing the boundaries of technology in embedded systems and electronics. This team is responsible for exploring emerging technologies and developing innovative solutions to meet the evolving needs of clients. Their work involves designing and creating prototypes, conducting rigorous testing to ensure reliability and performance, and refining products based on feedback and industry trends. The R&D department collaborates closely with clients to understand their specific requirements and transform their concepts into functional, cutting-edge solutions. This department plays a crucial role in maintaining Blackfox's position at the forefront of technological advancements.

3.2 PRODUCTION AND PACKAGING:

The Production and Packaging department oversees the entire lifecycle of manufacturing, from assembly to final packaging. This team is responsible for the efficient and high-quality production of electronic products, ensuring that every component is assembled correctly and meets stringent quality standards. They manage the workflow to optimize production efficiency, handle quality control processes, and implement best practices in manufacturing. The packaging process is meticulously managed to ensure that products are securely packaged and protected during transit, maintaining their integrity and quality upon delivery. This department's focus on precision and reliability ensures that clients receive well-crafted products that meet their specifications.

3.3 SALES:

The Sales department is instrumental in driving the company’s business growth and expanding its market presence. This team is responsible for identifying new business opportunities, building and nurturing relationships with clients, and managing the sales process from initial contact through to closing deals. They work to understand clients’ needs and offer tailored solutions that align with their objectives. The Sales department also collaborates with other departments to ensure that client expectations are met and exceeded. By leveraging market insights and sales strategies, this team plays a key role in achieving revenue targets and positioning Blackfox Embedded Solutions as a leader in the industry.

3.4 MARKETING:

The Marketing department is dedicated to enhancing Blackfox’s brand visibility and market reach. This team develops and executes comprehensive marketing strategies to promote the company’s products and services. They create compelling promotional materials, manage digital and traditional marketing campaigns, and engage with the target audience through various channels. The Marketing department also conducts market research to understand trends and customer preferences, allowing them to tailor messaging and outreach efforts effectively. By focusing on brand development, lead generation, and market positioning, this department helps to attract new clients and retain existing ones, driving overall business success.

CHAPTER -4

Starting and Running Your Electronics Business: Key Insights

4.1. ENTREPRENEURSHIP IDEAS AND STARTING YOUR OWN COMPANY:

Identify Your Niche:

Focus on a specific area where you have expertise or see a market need. For example, Blackfox specializes in embedded systems, IoT, and electronics development.

Create a Business Plan:

Outline your business goals, target market, competitive analysis, and financial projections. A well-thought-out business plan is essential for guiding your startup and attracting investors.

Legal and Financial Setup:

Register your business, choose a suitable legal structure (e.g., sole proprietorship, partnership, LLC), and set up financial systems for managing expenses, revenue, and taxes.

Develop Your Product or Service:

Start by developing a prototype or a minimum viable product (MVP). Test it thoroughly and gather feedback to refine your offering.

Build a Strong Network:

Connect with industry professionals, mentors, and potential clients through networking events, industry conferences, and online platforms.

4.2. STRUGGLES IN PRODUCTION:

Quality Control:

Maintaining high standards in production is crucial. Implement strict quality control measures to ensure your products meet industry standards and client expectations.

Supply Chain Management:

Managing suppliers and inventory can be challenging. Build reliable relationships with suppliers and develop contingency plans for potential disruptions.

Cost Management:

Keep a close eye on production costs and look for ways to optimize processes and reduce waste. Efficient production practices can help improve profitability.

Scalability:

Plan for scaling your production as your business grows. This might involve investing in new equipment, expanding your facility, or automating processes.

4.3. SELECTING WORKERS:

Define Roles and Responsibilities:

Clearly outline the roles and responsibilities for each position. Ensure that job descriptions are accurate and align with your company's needs.

Recruitment and Hiring:

Use various recruitment channels to find suitable candidates, such as job boards, industry networks, and recruitment agencies. Consider candidates with relevant experience and a passion for your industry.

Training and Development:

Provide comprehensive training to ensure that new hires understand your company's processes and standards. Invest in ongoing development to help employees grow and contribute effectively.

Company Culture:

Foster a positive company culture that aligns with your values and vision. A strong culture can help attract and retain top talent.

4.4. PACKAGING:

Design and Functionality:

Ensure that your packaging design is functional, protects the product, and reflects your brand identity. Consider factors like durability, ease of use, and visual appeal.

Cost-Efficiency:

Find cost-effective packaging solutions that don’t compromise quality. Bulk purchasing and standardizing packaging materials can help reduce costs.

Compliance and Regulations:

Ensure that your packaging complies with industry regulations and standards, including labeling requirements and environmental considerations.

Sustainability:

Consider sustainable packaging options that minimize environmental impact and appeal to eco-conscious consumers.

4.5. FINDING CLIENTS:

Market Research:

Conduct market research to identify potential clients and understand their needs. Tailor your marketing strategies to target these segments effectively.

Online Presence:

Establish a professional online presence through a well-designed website and active social media profiles. Use digital marketing techniques such as SEO, content marketing, and online advertising to attract clients.

CHAPTER-5

DAILY REPORT

FIELSD OF EMBEDDED SYSTEM 20/05/2024

Today, I learned about the various fields where embedded systems are crucial. These systems are at the core of automotive engineering, controlling functions like engine management and safety features. In consumer electronics, they power devices like smartphones and smart home appliances. In healthcare, embedded systems are used in medical devices such as patient monitors and pacemakers. They are also essential in industrial automation for controlling machinery and processes. The Internet of Things (IoT) further expands their role in connecting smart devices across industries like agriculture and energy. Overall, embedded systems are key to modern innovation, impacting diverse fields globally.

AUTOMOBILE,HEALTHCARE SECTOR 21/05/2024

Today, I focused on how embedded systems are used in the automobile and healthcare sectors. In automotive engineering, embedded systems control critical functions like engine management, braking, airbags, and infotainment systems, improving both safety and driving experience. In healthcare, these systems are vital in medical devices such as pacemakers, patient monitoring systems, and diagnostic equipment. Their precision and reliability are essential for patient care, making them key components in life-saving technology. These two fields showcase the importance of embedded systems in enhancing safety and functionality in our daily lives.

TELECOMMUNICATION AND AEROSPACE 22/05/2024

Today, I explored how embedded systems impact the telecommunication and aerospace sectors. In telecommunications, embedded systems are used in network equipment, base stations, and mobile devices, ensuring efficient data transmission and connectivity. These systems enable reliable communication across global networks, playing a vital role in modern communication infrastructure. In aerospace, embedded systems are critical in controlling navigation, avionics, and flight control systems. Their high precision and reliability are essential for the safety and operation of aircraft and spacecraft. These fields demonstrate how embedded systems contribute to complex and high-stakes industries.

ENERGY MANAGEMENT/AGRICULTURE/HOMEAUTOMATION 23/05/2024

Today, I focused on how embedded systems are utilized in energy, management, agriculture, and home automation. In the energy sector, embedded systems help monitor and control smart grids, optimize power distribution, and manage renewable energy sources like solar and wind. In agriculture, they are used in smart irrigation systems, automated machinery, and livestock monitoring, increasing efficiency and productivity. In home automation, embedded systems enable the control of lighting, security, HVAC systems, and smart appliances, making homes more energy-efficient and convenient. These fields highlight the versatility of embedded systems in improving efficiency and sustainability.

TRANSPORTATION/RETAIL POINT F SALE 24/05/2024

Today, I learned about the use of embedded systems in transportation and retail. In transportation, embedded systems are essential for traffic management, vehicle control, and navigation systems, ensuring safer and more efficient travel. They also play a role in public transportation systems by managing schedules, ticketing, and route optimization. In retail, embedded systems are commonly used in Point of Sale (POS) systems for transactions, inventory management, and data tracking, streamlining operations and enhancing customer service. These applications demonstrate how embedded systems improve functionality and efficiency in both industries.

MILTARY/ENTERTAINMENT MEADIA/WEARABLE 25/05/2024
DEVICES

Today, I studied the role of embedded systems in the military, entertainment media, and wearable devices. In the military, embedded systems are crucial for communication, surveillance, navigation, and weapons control, ensuring operational efficiency and safety. In the entertainment industry, they are used in gaming consoles, digital media players, and streaming devices, providing seamless user experiences. Wearable devices, such as fitness trackers and smartwatches, rely on embedded systems to monitor health metrics, track activity, and provide real-time data. These fields showcase the wide-reaching impact of embedded systems on both security and daily life.

BLYNK 27/05/2024

Today, I explored the use of Blynk in embedded systems. Blynk is a powerful IoT platform that allows easy integration of hardware with mobile apps for monitoring and controlling devices remotely. It enables users to create custom interfaces on their smartphones to interact with embedded systems like Arduino, ESP8266, and ESP32. Blynk's cloud service facilitates real-time data transmission and device control, making it ideal for home automation, smart agriculture, and energy management projects. Its simplicity and versatility help transform embedded systems into fully connected IoT solutions, enhancing remote functionality and user interaction.

BLYNK WITH SERVO 28/05/2024

Today, I worked on controlling a servo motor using Blynk in an embedded system project. Blynk provides an easy way to create a mobile app interface to control hardware remotely. By connecting a servo motor to an ESP8266 or Arduino, I was able to use the Blynk app to adjust the servo’s position in real-time. The app sends commands to the microcontroller through the Blynk cloud, which then adjusts the PWM signal to the servo, allowing precise control over its movement. This setup is ideal for applications like home automation, robotics, and remote control systems, showcasing Blynk’s powerful IoT integration capabilities

Additionally, Blynk's intuitive interface makes it simple to add sliders, buttons, or gauges for controlling multiple servos simultaneously. The platform's flexibility allows for easy expansion of the project, incorporating other sensors or actuators, making it a versatile tool for any IoT-based servo control system.

BLYNK WITH ULTRASONIC **29/05/2024**

Today, I integrated an ultrasonic sensor with Blynk for remote monitoring. The ultrasonic sensor measures the distance to an object by sending out sound waves and detecting their reflection. Using an ESP8266 or Arduino, I connected the sensor and linked it to the Blynk app. The sensor's readings are sent to the Blynk cloud, allowing real-time distance measurements to be displayed on my smartphone. This setup is useful for applications such as obstacle detection, level monitoring, and security systems.

Moreover, Blynk allows for easy customization of the app interface, so I could add gauges or charts to visualize the distance data more effectively. This project demonstrates how Blynk enhances the functionality of sensors, providing real-time data remotely, and making the system more interactive and user-friendly.

ARDUINO IOT CLOUD **30/05/2024**

Today, I explored the use of Arduino IoT Cloud for managing and controlling IoT devices. Arduino IoT Cloud allows seamless connection between hardware, such as Arduino boards, and cloud-based services for remote monitoring and control. By connecting sensors and actuators to an Arduino, I was able to send data to the cloud and access it via a web dashboard or mobile app. This platform supports real-time monitoring, alerts, and automation, making it ideal for applications in home automation, energy management, and smart agriculture.

Arduino IoT Cloud's integration with Alexa further enhances its versatility, enabling voice control of connected devices. The platform also provides built-in security features, such as secure authentication and encryption, ensuring data integrity and privacy.

ARDUINOIOT CLOUD WITH BUZZER 31/05/2024

In the lab, we configured static routing using either two or three routers to estaboday, I worked on a project using Arduino IoT Cloud to control a buzzer remotely. By connecting a buzzer to an Arduino board and linking it to the Arduino IoT Cloud, I was able to create a system that can trigger the buzzer through a web dashboard or mobile app. This allows real-time activation of the buzzer from anywhere, making it useful for applications like alarms, notifications, and security alerts.

With Arduino IoT Cloud, I could easily set up automation rules, such as triggering the buzzer when certain sensor thresholds are met. The platform also enables integration with Alexa for voice control, providing more convenience. This project demonstrates how Arduino IoT Cloud can be used to enhance simple components like a buzzer, turning them into part of a connected, interactive IoT system.

ARDUINOIOT CLOUD WITH ULTASONIC SENSOR 01/06/2024

Today, I explored using Arduino IoT Cloud with an ultrasonic sensor to remotely monitor distance measurements. By connecting the ultrasonic sensor to an Arduino board and linking it to the Arduino IoT Cloud, I was able to send real-time distance data to a cloud-based dashboard. This allowed me to track distance readings from anywhere via a web interface or mobile app, making the setup ideal for applications such as tank level monitoring, obstacle detection, or security systems.

THINGSPEAK **03/06/2024**

Today, I explored the use of ThingSpeak for data collection and analysis in IoT projects. ThingSpeak is a cloud platform that allows the aggregation, visualization, and analysis of sensor data from IoT devices such as Arduino, ESP8266, or Raspberry Pi. By connecting sensors to an Arduino and sending the data to ThingSpeak, I was able to monitor real-time readings via the platform's web interface.

ThingSpeak also provides powerful data visualization tools like graphs and charts, which make it easy to interpret the sensor data over time. Additionally, ThingSpeak supports MATLAB for advanced analytics and processing, enabling the creation of custom algorithms for applications like predictive maintenance or environmental monitoring. Overall, ThingSpeak offers a comprehensive solution for IoT data management, making it ideal for projects that require real-time monitoring and analysis.

HTML BASICS **04/06/2024**

Today, I reviewed the basics of HTML (HyperText Markup Language), which is the foundational language used for creating webpages. HTML structures a webpage using a series of tags and elements that define content such as headings, paragraphs, images, and links. The basic structure of an HTML document includes the <!DOCTYPE html> declaration, followed by the <html>, <head>, and <body> tags, where the content and metadata of the page are placed.

I learned about key HTML elements like <h1> to <h6> for headings, <p> for paragraphs, <a> for hyperlinks, and <img> for embedding images. HTML also allows for the use of attributes, which provide additional information about elements, such as src for image URLs or href for links.

Attributes and Elements in HTML 05/06/2024

Today, I focused on understanding the difference between attributes and elements in HTML. An **HTML element** is the basic building block of a webpage. Elements are typically defined by opening and closing tags, like <h1>Heading</h1>, where "h1" is the element used to define a heading. Common HTML elements include <p> for paragraphs, <a> for links, and <img> for images.

An **attribute** provides additional information about an HTML element. Attributes are included inside the opening tag and typically consist of a name and value pair, such as src="image.jpg" or href="https://example.com". For example, in the <img> element, the src attribute defines the image source, while in the <a> element, the href attribute specifies the link's destination. Attributes allow customization and further control over how elements are displayed and behave.

CREATING A WEBPAGE USING HTML 06/06/2024

Today, I created a basic webpage using HTML, learning how to structure and design a simple web page. I started by setting up the fundamental HTML document, which includes a basic structure with essential tags. The webpage began with the <!DOCTYPE html> declaration to define the document type, followed by the <html> tag, which contains all other elements. Within the <head> section, I added a title for the webpage, which appears in the browser tab. The main content of the page was placed in the <body> section, where I included a heading, a paragraph of text, a hyperlink, and an image. The hyperlink was used to direct users to another website, while the image tag was used to display a picture on the page.

READING SENSOR VALUES AND DISPLAYING IN A WEBPAGE 07/06/2024

Today, I worked on a project that involved reading sensor values and displaying them on a webpage. Using an embedded device such as an ESP8266 or Arduino, I connected a sensor (like a temperature or humidity sensor) to capture real-time data. The microcontroller reads the sensor values and, using a simple web server code, sends this data to a webpage. The webpage is set up to display the live sensor data in a readable format. Every few seconds, the sensor values are updated and refreshed on the webpage, allowing for real-time monitoring. This method is particularly useful for applications like home automation, weather stations, or any IoT-based system where remote monitoring of sensor data is crucial. The process provided insight into how web technologies can interact with physical sensors to display dynamic data in real-time

CONTROLLING A BUZZER USING A WEBPAGE 08/06/2024

Today, I worked on controlling a buzzer using a webpage. By setting up a simple web interface hosted on a microcontroller such as an ESP8266 or Arduino, I created a button on the webpage that allows the user to turn the buzzer on or off remotely. When the button on the webpage is clicked, it sends a command to the microcontroller, which then controls the buzzer through one of its digital output pins. This approach is particularly useful for remote control applications, such as home security alarms, notification systems, or automated alerts. The integration of a webpage with a physical device like a buzzer demonstrates how web technologies can effectively control hardware in real-time, enhancing interactivity and automation

SOLDERING(HANDS-ONPRACTICE) 10/06/2024

Today, I engaged in hands-on practice with soldering, an essential skill in electronics assembly and repair. The session involved soldering electronic components onto a printed circuit board (PCB) using a soldering iron and solder wire. I learned how to handle the soldering iron properly, maintain the correct temperature, and ensure that the solder joints were clean and well-formed to establish reliable electrical connections. Through this practice, I developed better control and precision in soldering techniques, improving my ability to connect components securely without causing damage. Soldering is a vital skill in prototyping and building electronics, and today's experience enhanced my confidence and dexterity in performing this task.

MQTT 24/06/2024

Today, I explored MQTT (Message Queuing Telemetry Transport), a lightweight messaging protocol used in the Internet of Things (IoT) for communication between devices. MQTT is based on a publish/subscribe model, where devices (clients) can publish messages to topics or subscribe to receive messages on specific topics. I learned how MQTT is ideal for scenarios requiring minimal bandwidth and power consumption, making it a popular choice for IoT devices and remote sensors. Using an MQTT broker (such as Mosquitto or a cloud-based service), I tested sending and receiving messages between devices like sensors and microcontrollers, enabling real-time data exchange. This protocol is widely used in applications such as home automation, environmental monitoring, and industrial IoT, and today's experience helped me understand its practical implementation and advantages.

CONTROLL LED USING MQTT(MOSQUITTO) 25/06/2024

Today, I implemented a project that involved controlling an LED remotely using MQTT with the Mosquitto broker. The setup included connecting an LED to a microcontroller like an ESP8266 or Arduino, which acted as an MQTT client. The microcontroller subscribed to a specific topic on the MQTT broker (Mosquitto), and whenever a message was published to that topic (e.g., "LED ON" or "LED OFF"), the microcontroller would execute the corresponding action to turn the LED on or off. This project demonstrated how MQTT enables efficient, real-time control of devices over a network, making it suitable for smart home applications or other IoT systems where remote device management is essential. It showcased the simplicity and power of MQTT for wireless communication and device control.

DISPLAY THE SENSOR VALUE USING MQTT 26/06/2024

(MOSQUITTO)

Today, I worked on a project where I displayed sensor values using MQTT with the Mosquitto broker. The setup involved a sensor connected to a microcontroller, such as an ESP8266 or Arduino, which read real-time data (e.g., temperature, humidity) from the sensor. The microcontroller acted as an MQTT client and published the sensor data to a specific topic on the Mosquitto broker. Another MQTT client, such as a mobile app or desktop application, subscribed to this topic and received the sensor data in real-time. The sensor values were then displayed on the subscriber's interface, enabling remote monitoring of environmental conditions. This method is highly effective for IoT applications like smart agriculture, weather monitoring, or home automation, where sensor data needs to be monitored remotely through efficient communication.

USING SHIFTR.IO FOR MQTT COMMUNICATION 27/06/2024

Today, I explored Shiftr.io, a cloud-based MQTT broker service that simplifies the process of setting up and managing MQTT communication between devices. I utilized Shiftr.io to publish and subscribe to topics for a project involving sensor data transmission. By connecting my microcontroller, such as an ESP8266, to the Shiftr.io broker, I was able to send real-time sensor data (like temperature or humidity readings) to the broker, which was then visualized on the Shiftr.io dashboard. I also tested remote control capabilities by sending commands to the microcontroller through MQTT topics, allowing me to control devices such as LEDs or buzzers remotely. Shiftr.io provides an intuitive and efficient way to manage and monitor MQTT communication in real-time, making it a valuable tool for IoT projects and smart applications.

CONTROL BUZZER USING SHIFTR 28/06/2024

Today, I implemented a project where I controlled a buzzer remotely using Shiftr.io, a cloud-based MQTT broker. I connected a buzzer to a microcontroller, such as an ESP8266, which subscribed to a specific MQTT topic on Shiftr.io. Through the Shiftr.io platform, I sent MQTT messages, such as "BUZZER ON" or "BUZZER OFF," to the subscribed topic. The microcontroller received these messages and activated or deactivated the buzzer accordingly. This allowed for remote control of the buzzer via the internet, demonstrating how MQTT communication can be effectively utilized for real-time device control. Shiftr.io provided a simple and interactive platform to monitor and manage the messages, making it ideal for IoT applications like alarms or notification systems.sense of accomplishment.

DISPLAY THE SENSOR VALUE USING SHIFTR 29/06/2024

Today, I worked on displaying sensor values using Shiftr.io for MQTT communication. I connected a sensor, such as a temperature or humidity sensor, to a microcontroller like an ESP8266. The microcontroller read the sensor data and published it to a specific topic on the Shiftr.io MQTT broker. By subscribing to that topic, I could visualize the real-time sensor values directly on the Shiftr.io dashboard or on any MQTT client. This setup allowed remote monitoring of the sensor data over the internet, making it easy to track environmental changes from anywhere. Shiftr.io's user-friendly interface provided a clear way to observe the flow of messages and sensor data, making it an effective tool for real-time IoT applications such as smart home systems and environmental monitoring.

SINRIC PRO 01/07/2024

Today, I explored Sinric Pro, a cloud-based platform designed for controlling IoT devices using Alexa, Google Home, or custom apps. Sinric Pro simplifies the integration of smart devices into home automation systems by providing APIs that allow microcontrollers, like ESP8266 or ESP32, to communicate with the platform. I tested controlling devices such as lights and buzzers through voice commands by integrating Sinric Pro with an ESP8266. The platform allows real-time interaction with connected devices, offering seamless control from anywhere using voice assistants or a mobile app. Sinric Pro's versatility makes it an ideal choice for creating smart home solutions, enabling IoT projects to be more responsive and user-friendly.

CONTROLLING BUZZER WITH RELAY USING SINRIC PRO 02/07/2024

Today, I worked on controlling a buzzer using a relay connected to Sinric Pro, a cloud-based IoT platform. I connected the buzzer to a relay module, which was controlled by an ESP8266 microcontroller integrated with Sinric Pro. By configuring the relay as a switch in the Sinric Pro app, I was able to remotely turn the buzzer on or off via the app or through voice commands using Alexa or Google Home. The relay acted as an intermediary, controlling the high-power buzzer safely, while the commands were sent through Sinric Pro's API. This setup demonstrated how Sinric Pro can be effectively used to control household devices in real-time, adding more functionality and convenience to IoT automation projects.

INTEGRATING VOICE ASSISTANCE IN IOT PROJECTS 03/07/2024

Today, I explored integrating voice assistance into IoT projects using platforms like Alexa and Google Home. By connecting my microcontroller, such as an ESP8266 or ESP32, to a cloud-based service like Sinric Pro, I was able to control devices through voice commands. Voice assistants, such as Alexa or Google Home, allowed me to issue commands like "Turn on the light" or "Activate the buzzer," which were then processed by the cloud platform and sent to the microcontroller to execute the desired action. This integration enhances the interactivity and convenience of IoT systems, allowing users to control various home automation devices hands-free. Voice assistance in IoT not only improves user experience but also adds an element of smart control to everyday tasks.

CONNECTING SINRIC PRO WITH MQTT 04/07/2024

Today, I focused on connecting Sinric Pro with MQTT to enhance device control and monitoring. Sinric Pro offers a straightforward way to control IoT devices via voice assistants like Alexa, and by integrating MQTT, it extends its capabilities for real-time data communication. I linked my microcontroller, such as an ESP8266, to both Sinric Pro and an MQTT broker. This setup allowed me to receive MQTT messages for monitoring sensor data while simultaneously controlling devices, such as turning a buzzer on or off, through Sinric Pro. By combining the two, I was able to create a more versatile system where voice commands could trigger MQTT events, enabling both voice control and real-time communication for IoT applications. This integration is ideal for complex smart home automation systems.

CONTROLLING SWITCHES USING SINRIC PRO AND MQTT 05/07/2024

Today, I successfully implemented a system where switches could be controlled via both Sinric Pro and MQTT. By integrating my microcontroller, such as an ESP8266, with Sinric Pro, I could manage the switches using voice commands through Alexa or Google Home. Simultaneously, I connected the microcontroller to an MQTT broker to enable real-time control and monitoring via MQTT messages. This dual-control setup allowed me to switch devices on or off either through voice commands in Sinric Pro or by publishing messages to the relevant MQTT topics. The integration of both systems provides flexibility, ensuring that devices can be controlled remotely, either through voice or MQTT, enhancing automation in smart homes and IoT environments.

CONNECTING SINRIC PRO THROUGH MQTT 06/06/2024 VOICE ASSISTANT

Today, I worked on integrating Sinric Pro with MQTT to enable control of IoT devices through Alexa voice assistant. By setting up an ESP8266 microcontroller with Sinric Pro, I linked it to both an MQTT broker and Alexa. This allowed me to control devices like lights or buzzers using Alexa voice commands, while also receiving real-time feedback and status updates via MQTT. The setup was seamless, as Sinric Pro provides native support for voice control, and by incorporating MQTT, I added enhanced data communication capabilities. This combination allows for greater flexibility in smart home automation, where voice commands from Alexa trigger device actions, which can then be monitored and controlled through MQTT messages.

INTEGRATING GSM WITH ARDUINO 16/07/2024

Today, I worked on integrating a GSM module with Arduino to enable cellular communication in my IoT project. The GSM module, such as SIM800 or SIM900, was connected to the Arduino via UART, allowing the microcontroller to send and receive SMS messages, make voice calls, and access mobile data networks. By using AT commands, I programmed the Arduino to interact with the GSM module, enabling features like sending alerts, controlling devices remotely via SMS, or receiving notifications. This integration is particularly useful for applications in remote areas where Wi-Fi or standard network connections are unavailable, making GSM a key component in projects like remote monitoring, emergency systems, and smart device control via cellular networks.

USING GSM WITH ARDUINO FOR CALL MANAGEMENT 17/07/2024

Today, I programmed the Arduino to manage calls using a GSM module, such as the SIM800A. By sending specific AT commands, the Arduino was able to initiate calls, answer incoming calls, and hang up when needed. For making a call, the command ATD followed by the phone number was used, while incoming calls were handled with the ATA command to answer and ATH to hang up. This setup enables the Arduino to act as a communication hub, ideal for projects like security systems or emergency response units, where automatic calls can be triggered by sensors or manual inputs, and incoming calls can be answered or managed remotely.

USING GSM WITH ARDUINO FOR SMS MANAGEMENT 18/07/2024

Today, I focused on implementing SMS management using a GSM module, such as the SIM800A, connected to an Arduino. I used AT commands to send and receive SMS messages, allowing the Arduino to communicate with mobile devices. The AT+CMGS command enabled the Arduino to send SMS messages by specifying the recipient's number and the message content. For reading incoming SMS messages, the AT+CMGR command was used to retrieve and process them. This functionality is essential for IoT applications that require remote control, monitoring, or alert systems, as SMS offers a reliable communication method, especially in areas with limited internet access.

CONNECTING GSM WITH MQTT 19/07/2024

Today, I integrated GSM with MQTT to enable wireless data communication over cellular networks. By using a GSM module like SIM800A with an Arduino, I established an internet connection through the mobile network by configuring the correct APN settings. Once connected, I used the PubSubClient library to send and receive MQTT messages over the cellular network. The Arduino acted as an MQTT client, publishing sensor data to an MQTT broker and subscribing to topics for real-time control of devices. This combination of GSM and MQTT is highly beneficial for IoT projects in remote areas where Wi-Fi is unavailable, allowing seamless data exchange and device control using mobile networks.

CONNECTING GSM WITH MQTT TO CONTROL BUZZER 20/07/2024

Today, I successfully connected a GSM module with MQTT to remotely control a buzzer using cellular networks. By integrating the SIM800A GSM module with an Arduino, I established an internet connection via the mobile network and configured the Arduino as an MQTT client using the PubSubClient library. The Arduino subscribed to an MQTT topic dedicated to controlling the buzzer. Whenever a message was published to the topic, the Arduino received it via the GSM module and executed the command to turn the buzzer on or off. This setup provides reliable remote control of devices, like a buzzer, over mobile networks, making it ideal for applications such as security systems, alarm systems, and emergency notifications, especially in areas lacking Wi-Fi connectivity.

RECEIVING SENSOR DATA THROUGH GSM AND MQTT 22/07/2024

Today, I implemented a system to receive sensor data using GSM and MQTT. I connected a sensor to an Arduino and integrated the GSM module (such as SIM800A) to enable data transmission over cellular networks. The Arduino was configured as an MQTT client, publishing sensor data to a specific topic on the MQTT broker. The GSM module facilitated the internet connection, allowing the Arduino to send sensor readings to the broker despite the lack of Wi-Fi. I set up another MQTT client to subscribe to the same topic, enabling real-time reception and display of the sensor data. This setup is particularly useful for remote monitoring applications, providing a reliable method to collect and view data from sensors in areas with limited or no internet access.

INTEGRATING CALL AND SMS PARALLELY WITH GSM AND ARDUINO 23/07/2024

Today, I worked on integrating call and SMS functionalities simultaneously using a GSM module with Arduino. The project involved managing both incoming and outgoing calls and SMS messages without interference. I configured the Arduino to handle GSM communication by setting up separate routines for SMS and call operations. Using AT commands, the Arduino was programmed to respond to incoming calls and SMS messages while also being able to initiate calls or send SMS messages. To manage parallel operations effectively, I implemented an interrupt-based approach where the GSM module's ring indicator (RI) pin triggered an interrupt for incoming calls or SMS, allowing the Arduino to handle each event appropriately. This integration ensures that the GSM module can manage multiple communication tasks concurrently.

INTEGRATING GSM TO RESPONSE CALLS AND SEND SMS 24/07/2024

Today, I focused on integrating a GSM module with Arduino to handle both responding to incoming calls and sending SMS messages. I used the GSM module (e.g., SIM800A) to enable the Arduino to manage call and SMS functions effectively.

For call management, I configured the Arduino to use AT commands to respond to incoming calls with the ATA command and hang up with the ATH command. The Arduino was programmed to monitor the GSM module’s ring indicator (RI) pin to detect incoming calls and trigger the appropriate response.

For SMS management, I set up the Arduino to send SMS messages using the AT+CMGS command and read incoming messages with the AT+CMGR command. This integration allows the Arduino to send text messages in response to specific events or commands and process incoming messages.

By implementing these functionalities, I enabled the Arduino to handle call responses and SMS operations simultaneously, making the system suitable for applications such as automated alert systems and remote communication solutions.

CONNECTING GSM WITH MQTT, VOICE, AND SMS 25/07/2024- 26/07/2024

Today, I worked on a complex integration involving GSM, MQTT, voice calls, and SMS with Arduino. The goal was to enable comprehensive communication capabilities, where the GSM module would handle voice calls, SMS, and MQTT data transmission simultaneously. This integration required managing multiple tasks concurrently: receiving and sending SMS, handling incoming and outgoing voice calls, and processing MQTT messages for remote control and monitoring.

The GSM module, such as the SIM800A, was connected to the Arduino and configured to handle voice and SMS functionalities using AT commands. Simultaneously, MQTT communication was established to send and receive messages through a cellular network. I implemented a system where the Arduino could process MQTT messages to control devices while managing voice calls and SMS notifications. This involved careful coding and task management to ensure smooth operation across all communication channels.

Despite the complexity, this integration demonstrates the capability of combining multiple communication methods in a single system, enhancing its versatility and functionality for advanced IoT applications.

DEBUGGING ERRORS IN GSM AND ARDUINO CODE 07/8/2024

Today, I tackled debugging issues in the GSM and Arduino code, focusing on call responses and SMS functionality. I identified errors related to AT command execution, interrupt handling for incoming calls, and SMS management. I corrected the AT commands and improved interrupt handling to ensure the Arduino could respond correctly to calls and messages. Extensive testing was conducted to verify that the system could manage multiple incoming calls and SMS messages simultaneously without conflicts. These adjustments enhanced the stability and performance of the GSM module integration, ensuring reliable operation of both call and SMS functionalities.

TRYING TO CONNECT USING SIM800C 08/08/2024

Today, I worked on integrating the SIM800C GSM module with Arduino for communication tasks. I focused on setting up the SIM800C to handle both SMS and call functionalities. This involved configuring the module with the correct AT commands and establishing a stable connection for sending and receiving SMS messages and managing calls. I tested various scenarios to ensure the SIM800C could effectively handle call responses and SMS operations. This new setup aims to enhance the reliability and performance of the GSM communication in my project.

DEBUGGING MQTT IDENTIFICATION USING SERIAL INTERRUPT WITH SIM800C 09/08/2024

Today, I addressed the issue where MQTT messages were not being correctly identified during incoming calls or SMS with the SIM800C module. To resolve this, I implemented a serial interrupt system to prioritize and manage MQTT communication alongside GSM tasks. By setting up interrupts on the serial communication line, I aimed to ensure that MQTT messages are processed even when there are concurrent GSM activities. This involved configuring the Arduino to handle incoming GSM interrupts while maintaining reliable MQTT communication. The goal was to achieve seamless operation where both GSM and MQTT functions could be managed effectively without interference.

DEBUGGING MQTT IDENTIFICATION THROUGH POLLING METHOD WITH SIM800C 10/08/2024

Today, I worked on resolving the issue where MQTT messages were not being identified correctly during incoming calls or SMS with the SIM800C module. To address this, I implemented a polling method to manage MQTT communication alongside GSM tasks. Instead of relying on interrupts, I periodically checked the status of the GSM module and MQTT messages using polling. This approach involved writing code to regularly query the GSM module for any new calls or SMS and simultaneously check for MQTT messages. By adjusting the polling intervals and ensuring that both GSM and MQTT functions were monitored effectively, I aimed to achieve reliable communication and data handling even when multiple tasks are occurring concurrently.

DEBUGGING MQTT IDENTIFICATION USING TIMER INTERRUPT WITH SIM800C 12/08/2024

Today, I focused on resolving the issue where MQTT messages were not being correctly identified during incoming calls or SMS with the SIM800C module. I implemented a timer interrupt method to handle both GSM and MQTT tasks efficiently. By configuring a timer interrupt, I ensured that the Arduino periodically checks the status of the GSM module and processes MQTT messages at regular intervals. This approach allows the system to manage GSM communications and MQTT operations concurrently, without missing critical messages or calls. Testing confirmed that the timer interrupt method provided a reliable way to handle both GSM and MQTT tasks, improving the overall performance and responsiveness of the communication system.

ANALYSIS OF GSM MODULE DATASHEET 13/08/2024

Today, I analyzed the datasheet of the GSM module (SIM800C) to understand its specifications and capabilities better. The datasheet provided detailed information on the module’s operational parameters, including voltage requirements, current consumption, and supported frequency bands. Key features such as AT command sets for SMS, call management, and data communication were reviewed to ensure correct implementation in my project. I also examined the pin configurations, antenna requirements, and recommended operating conditions to optimize the module's performance. This analysis is crucial for ensuring that the GSM module is integrated correctly and operates reliably within the specified parameters for my application.

USING RI PIN INTERRUPT FOR VOICE AND SMS WITH MQTT 14/08/2024

Today, I integrated the Ring Indicator (RI) pin of the GSM module (SIM800C) with the Arduino to handle external interrupts for detecting incoming voice calls and SMS messages. The RI pin was configured to trigger an interrupt on the Arduino whenever there was an incoming call or SMS, allowing the system to promptly respond to these events. By setting up the interrupt, I ensured that the GSM module's status changes were monitored effectively without continuously polling, which improved the responsiveness of the system.

In parallel, I worked on connecting this setup with MQTT communication. The goal was to ensure that while handling GSM events via interrupts, the Arduino could also manage MQTT messages effectively. This integration aimed to provide a seamless operation where incoming voice and SMS notifications were processed in real-time, and MQTT messages were handled concurrently. The implementation of the RI pin interrupt improved the system's ability to manage both GSM and MQTT tasks efficiently, enhancing overall performance and reliability.

SYNCHRONIZING MQTT WITH VOICE AND SMS 15/08/2024

Today, I focused on synchronizing MQTT communication with voice and SMS functionalities. I integrated the GSM module with MQTT to manage incoming and outgoing calls, SMS, and MQTT messages simultaneously. The system was designed to handle MQTT messages for remote control and monitoring while processing voice calls and SMS notifications. Synchronization was achieved by implementing a structured approach to ensure that all communication channels functioned efficiently and without conflicts, providing a reliable system for managing multiple communication tasks concurrently.

SYNCHRONIZING MQTT WITH VOICE AND SMS ON ESP32 AND NODEMCU 16/08/2024

Today, I focused on integrating and synchronizing MQTT communication with voice and SMS functionalities using ESP32 and NodeMCU microcontrollers. The project involved configuring a GSM module to handle both SMS and voice calls while also managing MQTT communication for remote control and monitoring purposes.

I began by setting up the GSM module to manage incoming and outgoing calls and SMS messages. For the ESP32 or NodeMCU, I configured it as an MQTT client to send and receive messages through a broker. The challenge was to ensure that the GSM module and MQTT communication could operate concurrently without interfering with each other.

To achieve this, I implemented efficient task management and scheduling techniques. The GSM module was programmed to handle voice and SMS events in real-time, utilizing interrupts for prompt detection of incoming calls and messages. Meanwhile, the ESP32 or NodeMCU was set up to handle MQTT messages, ensuring that data could be sent and received seamlessly.

By integrating these components, I ensured that the system could effectively manage multiple communication channels. The MQTT communication provided remote control and monitoring capabilities, while the GSM module handled voice and SMS functionalities. This setup enhanced the overall functionality and reliability of the system.

INTRODUCTION TO PIC MICROCONTROLLERS 19/08/2024

PIC microcontrollers, developed by Microchip Technology, are versatile and widely used in embedded systems. "PIC" stands for "Peripheral Interface Controller," indicating their design to handle various peripheral tasks. They come in several families, such as PIC10, PIC12, PIC16, PIC18, and PIC32, each with distinct features. PIC microcontrollers feature a CPU, memory, and peripherals like timers, ADCs, and communication interfaces (UART, SPI, I2C). They can be programmed in assembly language or C, using tools like MPLAB X IDE and XC compiler. PIC microcontrollers are employed in applications ranging from consumer electronics to industrial automation, thanks to their flexibility and comprehensive support.

BLINKING LED USING PIC MICROCONTROLLER 20/08/2024

During my internship, I developed a simple application to blink an LED using a PIC microcontroller. I connected the LED to an I/O pin and set the pin as an output. Using MPLAB X IDE, I wrote a C program that toggled the pin's state, making the LED blink with a 500 ms on/off cycle. This project provided practical experience in configuring I/O pins, using delay functions, and understanding basic microcontroller operations. It was an effective introduction to embedded systems programming.This project enhanced my understanding of microcontroller programming and I/O handling.

PULL-UP AND PULL-DOWN RESISTORS 21/08/2024

During my internship, I explored the concepts of pull-up and pull-down resistors in digital circuits. Pull-up resistors are used to ensure that a pin is at a high logic level when it is not actively being driven low, whereas pull-down resistors ensure the pin is at a low logic level when not driven high. I implemented both types of resistors in my projects to stabilize signal levels and prevent floating states. By configuring pull-up and pull-down resistors, I learned how they influence digital input stability and improve circuit reliability. This experience enhanced my understanding of proper signal handling in embedded systems.

LCD DISPLAY INTERFACE WITH 22/08/2024 PIC MICROCONTROLLER

During my internship, I interfaced an LCD display with a PIC microcontroller to display text and information. I connected the LCD to the microcontroller, configuring the data and control pins correctly. Using MPLAB X IDE, I wrote a C program to initialize the LCD, set the display mode, and send commands and data for text output. This involved configuring the LCD for 8-bit or 4-bit operation, handling data and control signals precisely. I also implemented functions to manage cursor positioning and clear the display, which improved the user interface. This project enhanced my skills in managing peripheral devices and developing effective display solutions in embedded systems.

7-SEGMENT DISPLAY INTERFACE WITH PIC MICROCONTROLLER 23/08/2024

During my internship, I worked on interfacing a 7-segment display with a PIC microcontroller to show numerical values. I connected the 7-segment display to the microcontroller through appropriate I/O pins, setting up the segments to represent digits from 0 to 9. I wrote a C program using MPLAB X IDE to control the display, sending binary data to light up the correct segments for each digit. The project involved creating functions to display numbers, manage digit transitions, and update the display in real-time. I also implemented multiplexing techniques to drive multiple 7-segment displays with fewer I/O pins. This hands-on experience enhanced my understanding of digital display technologies and microcontroller interfacing, crucial for developing user-friendly embedded systems.

KEYBOARD INTERFACE WITH PIC 24/08/2024

During my internship, I worked on interfacing a keyboard with a PIC microcontroller to capture and process user inputs. I connected the keyboard to the microcontroller's I/O pins, configuring it to detect key presses and generate corresponding signals. Using MPLAB X IDE, I developed a C program to read and decode the keyboard inputs, translating them into actions or displayed characters. I implemented debouncing techniques to ensure accurate key detection and prevent false triggers. This project provided valuable experience in handling input devices, managing data acquisition, and integrating user input into embedded systems.

LOGICAL THINKING WITH PIC 26/08/2024

During my internship, I focused on applying logical thinking to solve problems using a PIC microcontroller. I developed and implemented various projects, such as controlling LEDs, reading sensor data, and interfacing with peripherals. Each project required careful planning and problem-solving to translate functional requirements into effective microcontroller code. By designing algorithms and writing efficient programs, I learned to address issues like signal processing, input handling, and communication protocols. This hands-on experience honed my logical thinking and troubleshooting skills, essential for developing reliable embedded systems and understanding microcontroller operations in depth.

PWM WITH PIC MICROCONTROLLER 27/08/2024

During my internship, I implemented Pulse Width Modulation (PWM) using a PIC microcontroller to control device speed and brightness. I configured the PWM module of the PIC microcontroller to generate variable duty cycles, adjusting the output signal's width. By programming the duty cycle through the microcontroller’s registers, I controlled the intensity of an LED and the speed of a motor. This project involved setting the PWM frequency, adjusting the duty cycle values, and fine-tuning the output to achieve desired performance. This hands-on experience with PWM enhanced my understanding of signal modulation and its applications in embedded systems.

PWM WITH LED IN PIC MICROCONTROLLER 28/08/2024

During my internship, I worked on implementing Pulse Width Modulation (PWM) with an LED using a PIC microcontroller. I set up the microcontroller to generate PWM signals, controlling the LED's brightness by varying the duty cycle of the signal. By adjusting the duty cycle values through the PIC’s PWM registers, I was able to achieve different brightness levels, demonstrating how PWM can be used for fine-grained control of LED intensity. This project involved configuring the PWM frequency and ensuring smooth transitions between brightness levels. This hands-on experience provided a practical understanding of PWM’s role in managing output signals and its applications in embedded systems.

ADC WITH PIC MICROCONTROLLER 29/08/2024

During my internship, I implemented Analog-to-Digital Conversion (ADC) using a PIC microcontroller to read and process analog signals. I connected an analog sensor to one of the ADC input pins of the PIC and configured the ADC module through the microcontroller's registers. Using MPLAB X IDE, I wrote a C program to initiate ADC conversions, read the digital values, and interpret them to reflect the sensor’s output. The project involved setting up the ADC's reference voltage, configuring sampling rates, and processing the resulting data. This experience enhanced my understanding of analog signal processing and the integration of ADC functionality in embedded systems.

GETTING SENSOR DATA WITH PIC 30/08/2024

During my internship, I focused on acquiring sensor data using a PIC microcontroller. I interfaced an analog sensor with the PIC, connecting it to the ADC input pin. I configured the ADC module to convert the analog signal from the sensor into a digital value. Using MPLAB X IDE, I developed a C program to read the digital data from the ADC, process it, and display or use it as needed. This project involved setting up the ADC configuration, handling sensor calibration, and ensuring accurate data acquisition. This experience improved my skills in interfacing sensors with microcontrollers and processing analog data in embedded systems.

CHAPTER - 6

PROJECT DEVELOPMENT
GSM MQTT WITH BUZZER

ABSTRACT:

This project aims to develop a remote buzzer control system using GSM (Global System for Mobile Communications) and MQTT (Message Queuing Telemetry Transport) protocols, integrated with the Shiftr.io MQTT broker. The system employs the SIM800 GSM module for cellular connectivity and an Arduino microcontroller to interface with the GSM module and handle MQTT communication. The main objective is to enable the control of a buzzer through remote commands sent over the internet, achieving real-time status updates of the buzzer state using MQTT while utilizing readily available and cost-effective hardware components for implementation.

The system components include the SIM800 GSM module for cellular network connectivity, an Arduino microcontroller as the central control unit, a buzzer as the output device, and the Shiftr.io MQTT broker to facilitate MQTT messaging between the client and the server. The methodology involves setting up the hardware by connecting the SIM800 module to the Arduino microcontroller and the buzzer to a designated GPIO pin. The software implementation includes initializing the GSM module, configuring the Arduino to connect to the Shiftr.io MQTT broker, handling received MQTT messages to control the buzzer, and publishing the buzzer's state to inform remote users.

The system successfully demonstrates the ability to remotely control the buzzer through MQTT messages. The buzzer can be turned or off based on commands received via the MQTT broker,

COMPONENTS:

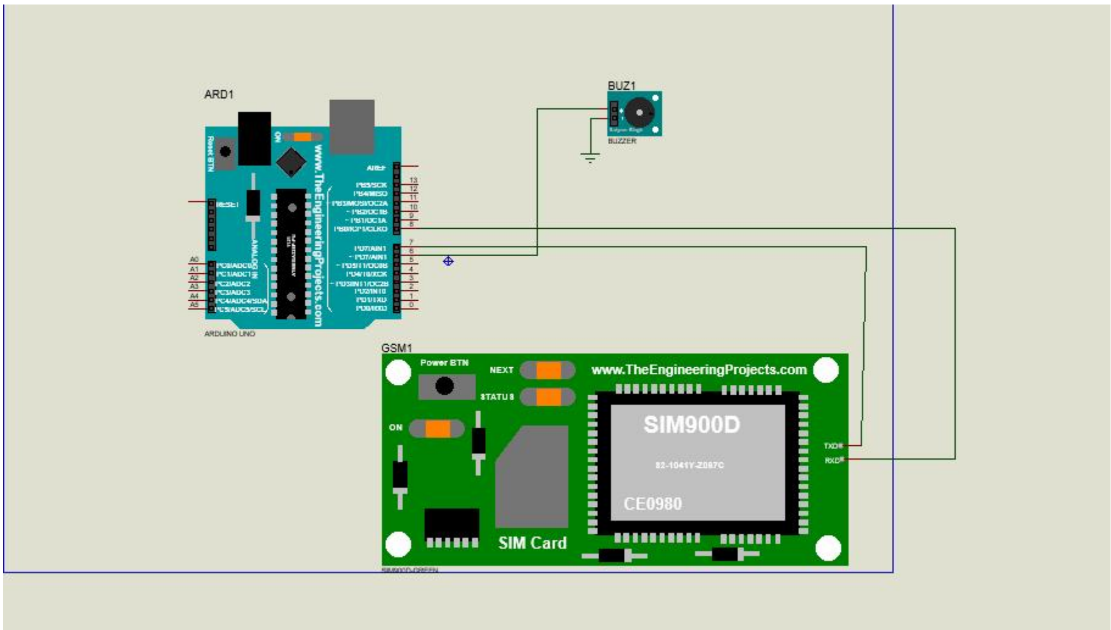
ARDUINO: Arduino is an open-source electronics platform based on easy-to-use hardware and software, designed for creating interactive projects. It allows users to write code and upload it to a microcontroller to control various sensors, motors, lights, and other electronic components.

GSM SIM800C: The GSM SIM800C is a quad-band GSM/GPRS module that provides a cost-effective and reliable solution for embedding cellular connectivity in projects. It supports voice, SMS, and data transmission, making it ideal for IoT applications requiring remote communication.

Jumper Wires: Jumper wires are insulated wires with connectors (typically male or female pins) at each end, used to create temporary or semi-permanent connections between different components on a breadboard or between a breadboard and other devices.

BUZZER: The buzzer system integrated with the Arduino IoT Cloud enables remote toggling of its state, providing a seamless user experience for activating or deactivating auditory feedback. This functionality is achieved through the synchronization of the NodeMCU microcontroller with the cloud platform, facilitating real-time control and monitoring of the buzzer's operations from anywhere with an internet connection.

CIRCUIT DIAGRAM:

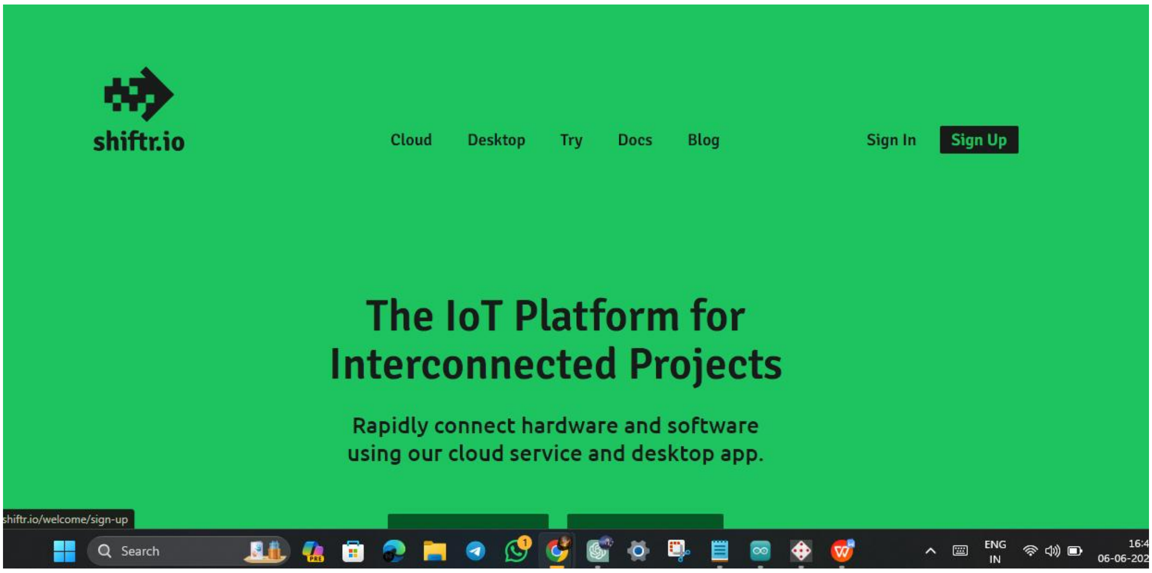


STEPS TO CREATE A PROJECT

Step 1: Set Up Shiftr.io

Create an Account:

- Visit Shiftr.io and create an account.

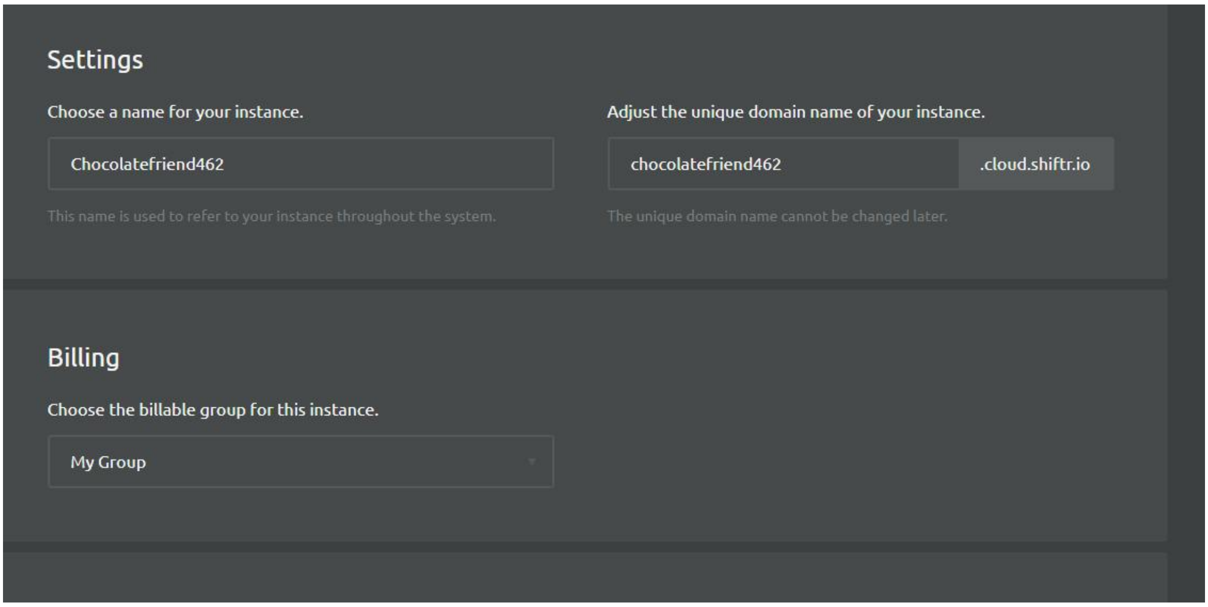


The IoT Platform for Interconnected Projects

Rapidly connect hardware and software using our cloud service and desktop app.

Create a New Space:

- ➢ After logging in, create a new space. This space will be your MQTT broker.



Settings

Choose a name for your instance.

Chocolatefriend462

This name is used to refer to your instance throughout the system.

Adjust the unique domain name of your instance.

chocolatefriend462 .cloud.shiftr.io

The unique domain name cannot be changed later.

Billing

Choose the billable group for this instance.

My Group

Get Connection Details:

- ➢ Note the broker URL, username, and password provided by Shiftr.io. You will need these for your ESP32 and MQTT Box setup.

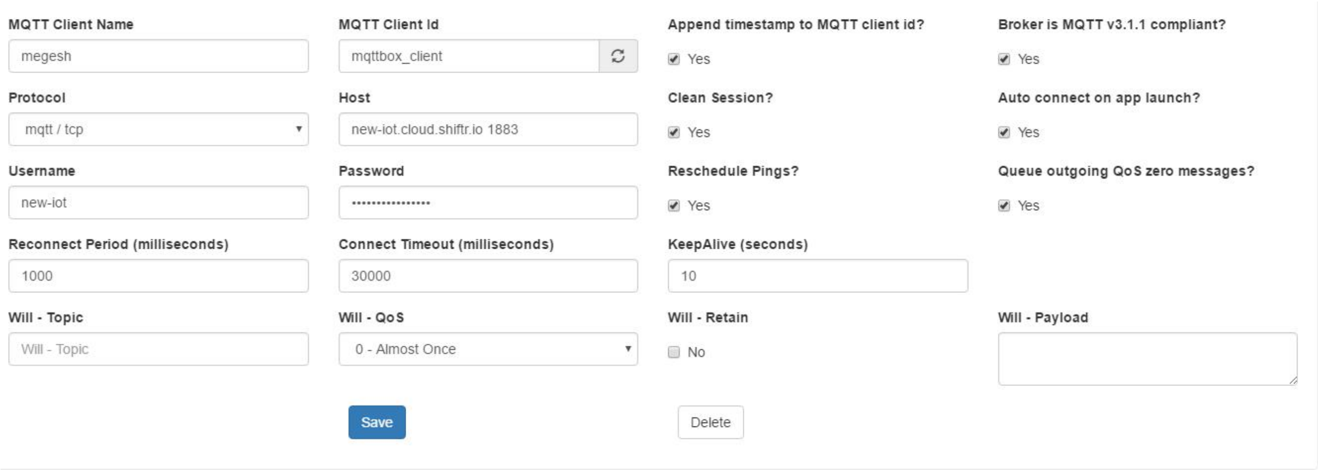
Step 2: Set Up MQTT Box

Install MQTT Box:

- ➢ Download and install MQTT Box from MQTT Box.

Create a New Connection:

- ➢ Open MQTT Box and create a new connection.
- ➢ Enter the broker URL, username, and password you got from Shiftr.io.



MQTT Client Name: megesh

MQTT Client Id: mqttbox_client

Append timestamp to MQTT client id? ☑ Yes

Broker is MQTT v3.1.1 compliant? ☑ Yes

Protocol: mqtt / tcp

Host: new-iot.cloud.shiftr.io 1883

Clean Session? ☑ Yes

Auto connect on app launch? ☑ Yes

Username: new-iot

Password: ••••••••••••••••

Reschedule Pings? ☑ Yes

Queue outgoing QoS zero messages? ☑ Yes

Reconnect Period (milliseconds): 1000

Connect Timeout (milliseconds): 30000

KeepAlive (seconds): 10

Will - Topic: Will - Topic

Will - QoS: 0 - Almost Once

Will - Retain: □ No

Will - Payload:

Subscribe to Topics:

- ➢ Subscribe to the following topics to monitor and control the buzzer:
- ➢ /ultra for controlling the buzzer

Step 3: Connect Hardware Components

Buzzer:

- ➢ Connect the positive terminal to a digital pin (e.g., D26).
- ➢ Connect the negative terminal to the ground.
- ➢ Step 4: Write the Arduino Code

Libraries:

- ➢ Ensure you have the necessary libraries installed:
- ➢ WiFi
- ➢ MQTT

Step 5: Upload Code to arduino

Arduino IDE Setup:

- Ensure you have thearduini uno board support installed in the Arduino IDE.
- Select the correct board and port from the Tools menu.

Upload the Code:

- Connect your arduino uno to the computer via USB.
- Upload the code.

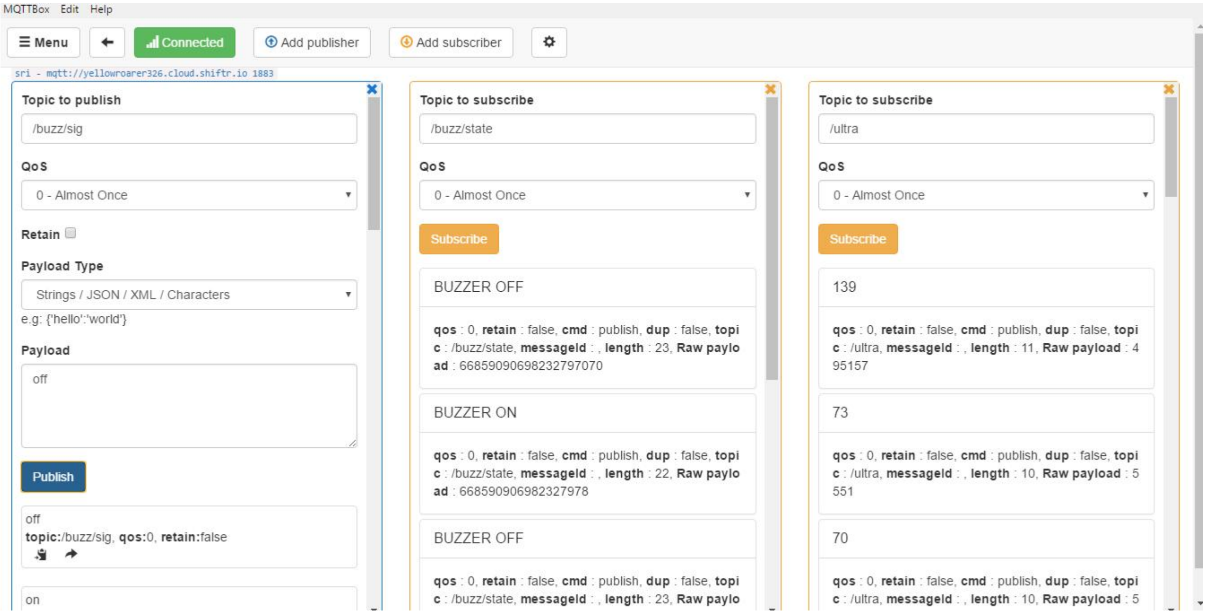
Step 6: Monitor and Control via MQTT Box

Control the Buzzer:

- Publish "on" to the /buzzer topic to turn the buzzer on.
- Publish "off" to the /buzzer topic to turn the buzzer off.

Monitor Buzzer Status:

- Subscribe to the /buzzerStatus topic to receive updates on the buzzer's status (ON or OFF).



sri - mqtt://yellowroarer326.cloud.shiftr.io 1883

Topic to publish: /buzz/sig

QoS: 0 - Almost Once

Retain □

Payload Type: Strings / JSON / XML / Characters

e.g: {'hello':'world'}

Payload: off

```
off
topic:/buzz/sig, qos:0, retain:false

on
```

Topic to subscribe: /buzz/state

QoS: 0 - Almost Once

```
BUZZER OFF
qos : 0, retain : false, cmd : publish, dup : false, topic : /buzz/state, messageId : , length : 23, Raw payload : 66859090698232797070

BUZZER ON
qos : 0, retain : false, cmd : publish, dup : false, topic : /buzz/state, messageId : , length : 22, Raw payload : 668590906982327978

BUZZER OFF
qos : 0, retain : false, cmd : publish, dup : false, topic : /buzz/state, messageId : , length : 23, Raw paylo
```

Topic to subscribe: /ultra

QoS: 0 - Almost Once

```
139
qos : 0, retain : false, cmd : publish, dup : false, topic : /ultra, messageId : , length : 11, Raw payload : 495157

73
qos : 0, retain : false, cmd : publish, dup : false, topic : /ultra, messageId : , length : 10, Raw payload : 5551

70
qos : 0, retain : false, cmd : publish, dup : false, topic : /ultra, messageId : , length : 10, Raw payload : 5
```

CHAPTER – 7

CONCLUSION

My internship offered an in-depth exploration of cutting-edge technologies in embedded systems and IoT, significantly broadening my expertise and practical skills. I engaged in several projects that involved integrating GSM modules for effective communication tasks, including managing calls and SMS, and interfacing these with MQTT protocols for robust data transmission and remote monitoring. This experience deepened my understanding of how to use GSM for real-time communication and how MQTT facilitates efficient data exchange in IoT applications.

Additionally, I worked with Sinric Pro and Blynk platforms, implementing voice control and mobile app interfaces to manage and control devices remotely. These projects showcased the potential of integrating voice assistants and mobile applications into embedded systems, enhancing user interaction and control. By using Arduino IoT Cloud, I developed solutions for cloud-based data monitoring and device management, demonstrating the capability of cloud services to provide scalable and flexible system solutions.

Throughout the internship, I tackled challenges such as system integration, protocol management, and user interface development, gaining valuable experience in configuring and deploying complex systems. This hands-on exposure to diverse technologies and platforms has not only improved my technical proficiency but also enhanced my problem-solving abilities and understanding of embedded systems and IoT applications.