Al-Quds University

Najjad Zeenni Faculty of Engineering

Computer Engineering Department

Graduation Project

(0702592)

IoT-Driven Smart Baby Care System

Prepared By:

Remana Hawamdeh 22020031

Eman Abu Sarhan 22020052

Ferdaws Qandeel 22010371

Supervised by:

Prof. Dr. Salaheddin Odeh,

Dr. Yacoub Sabatin

2024 - 2025

Dedication

Alhamdulillah, for His blessings, guidance, and strength throughout our journey.

We dedicate this project, "*IoT-Driven Smart Baby Care System,"* to the essence of care, love, and the shared responsibility of nurturing life.

To our beloved mothers, the first source of love, warmth, and care—whose sacrifices, sleepless nights, and boundless compassion laid the very foundation of our understanding of care.

To our fathers, whose strength, support, and quiet endurance have guided us through every challenge.

To our sisters and brothers, who stood by us with patience, encouragement, and love, even in moments when we were too busy or tired to notice.

To our teachers and mentors, especially our supervisors, *Prof. Dr. Salaheddin Odeh* and *Dr. Yacoub Sabatin,* whose expert guidance and dedication have been invaluable in shaping this project.

To our friends and colleagues, whose unwavering support and understanding kept us motivated through every obstacle.

And finally, to all who work toward a future where care and technology come together to support and protect those who need it most—we dedicate this effort to you.

Abstract

In today's dynamic world, parents face the dual challenge of balancing professional responsibilities with ensuring their child's well-being. This project presents an innovative engineering solution to enhance childcare using integrated IoT technologies. The system addresses critical concerns such as environmental monitoring, cry detection, and diaper wetness notifications, offering parents peace of mind and improved caregiving efficiency. By combining smart sensors, automated responses, and real-time alerts, the system bridges the gap between work and childcare, creating a safer, more comfortable, and connected environment for infants. This initiative demonstrates how engineering advancements can transform modern parenting challenges into effective solutions.

الملخص:

في عالمنا المتغير باستمرار، يواجه الآباء تحديًا مزدوجًا يتمثل في الموازنة بين مسؤولياتهم المهنية وضمان سلامة أطفالهم. يقدم هذا المشروع حلاً هندسيًا مبتكرًا لتحسين رعاية الأطفال باستخدام تقنيات إنترنت الأشياء المتكاملة. يعالج النظام مشكلات بالغة الأهمية، مثل مراقبة البيئة، وكشف البكاء، وإشعارات بلل الحفاضات، مما يوفر للآباء راحة البال ويعزز كفاءة الرعاية. من خلال الجمع بين أجهزة الاستشعار الذكية والاستجابات الآلية والتنبيهات الفورية، يسد النظام الفجوة بين العمل ورعاية الأطفال، مما يخلق بيئة أكثر أمانًا وراحة وتواصلًا للأطفال الرضع. توضح هذه المبادرة كيف يمكن للتطورات الهندسية أن تحول تحديات الأبوة والأمومة الحديثة إلى حلول فعالة.

Preface

In an age where artificial intelligence and the Internet of Things (IoT) are seamlessly integrated into everyday life, the IoT-Driven Smart Baby Care System represents a revolutionary advancement in modern parenting. This system goes beyond traditional monitoring by analyzing vital signs, tracking sleep patterns, and proactively assessing a baby's overall well-being. Imagine a crib equipped with sensors that measure heart rate and breathing, or smart cameras that detect facial expressions to interpret a baby's needs responding with adaptive lighting and soothing sounds .tailored to comfort

More than just a data-gathering tool, this system creates a responsive and interactive environment, alleviating parental concerns and enhancing their ability to provide care, even amidst busy schedules. By leveraging advanced algorithms and intelligent automation, it functions as a digital companion an extension of parental intuition that ensures a child’s safety and comfort at all times. As we embrace the future of smart parenting, this innovation marks a significant leap toward a world where technology and nurturing instincts work hand in hand to shape a more secure and supportive childhood experience.

Table of Contents:

Chapter One: Introduction

Chapter Two: System Analysis

Chapter Three: System Functional Requirements

<u>List of Figures:</u>

List of Tables:

Chapter One : Introduction

1.1 Overview

The IoT-Driven Smart Baby Care System is an intelligent solution designed to automate and enhance infant care by integrating modern IoT technologies into the daily routines of parents and caregivers. The system continuously monitors key parameters related to the baby's environment and behavior, such as temperature, humidity, crying, and diaper wetness, to ensure real-time awareness and proactive responses.

At its core, the system utilizes a network of interconnected sensors and microcontrollers that collect and analyze data, triggering automated actions like cradle rocking or sending immediate SMS alerts to caregivers. This enables continuous monitoring and rapid responses to potential issues, ensuring a seamless and efficient support system for infant care.

A major feature of the system is its wireless communication capability, particularly through the GSM module and ESP32-CAM, which enables remote access, live video streaming, and SMS-based notifications. This ensures that parents remain connected to their child's status even when physically distant, enhancing both safety and peace of mind.

The system also includes mobile accessibility, allowing caregivers to receive alerts, view live data, and access historical logs anytime and anywhere. This mobile integration ensures ease of use and promotes timely decision-making.

By combining IoT, automation, and remote connectivity, the project delivers a smart, reliable, and user-friendly baby care assistant that supports modern families and advances the future of personalized, technology-driven childcare.

1.2 Project Idea Description

The IoT-Driven Smart Baby Care System is envisioned as a smart, real-time solution designed to enhance modern infant care through the integration of sensor-based monitoring, automation, and remote communication. The main concept is to utilize Internet of Things (IoT) technology to continuously track the baby's environment and condition—

such as temperature, humidity, crying and diaper wetness, automatically responding or notifying caregivers when attention is needed.

This project aims to address the limitations of traditional baby monitoring systems, which are often reactive and lack advanced automation. By enabling real-time monitoring and intelligent response mechanisms, the system helps prevent discomfort, reduces risks, and improves caregiving efficiency. Collected data is processed by a microcontroller (Arduino Uno), which controls the system’s actions, including activating a servo motor to rock the crib or sending SMS alerts via the GSM module when necessary.

A key aspect of this project is its ability to operate remotely. Through GSM communication and the ESP32-CAM module, the system allows parents to receive real-time alerts and live video feeds on their mobile devices. This ensures that they are always aware of their baby's condition, even when not physically present.

By integrating smart sensing technologies, automated comfort features, and remote access capabilities, the project presents a practical and scalable solution that reflects how IoT can transform traditional caregiving into a proactive, responsive, and intelligent system tailored.

1.3 Problem Statement

A very common concern parents have stems from the anxiety of not always being in view of the baby and in control of their surroundings or behavior. Specific anxieties include never knowing whether the baby is crying, needing a change, the room temperature is too hot or too cold.

Moreover, parents often have to trust someone else with the care of their baby. The ability to ensure that caregivers respond to the needs of the infant is an anxiety problem all its own. These various concerns stem from the failure of a holistic, real-time monitoring system that keeps them up-to-date on the ongoing status of the baby.

Traditional solutions include audio monitors, periodic physical checks, or basic video monitors: these usually cannot meet the above needs. Audio monitors can hardly provide sufficient information about the state of the baby, physical checks are disturbing and impractical for busy parents, and basic video monitors lack advanced features such as environmental monitoring and automated response. In the absence of a reliable system to fill these gaps, parents are left with increased uncertainty and stress. This situation underlines the need for an effective solution that will ensure the baby's safety, comfort, and well-being while offering peace of mind to the parents, knowing their baby is continuously and effectively monitored.

1.4 Project Objectives

The main focus of this project is to develop an IoT-Driven Smart Baby Care System that incorporates various technologies to care for babies by automatically observing their well-being and soothing them. It's a system that will take some of the load off parenting, especially for working parents of babies. The system keeps mothers and fathers who need to be away from their children free from anxiety by decreasing the risk of possible dangers and creating a safer environment. Moreover, it would offer smart solutions to make babies more comfortable.

1.5 Project Benefits

1. **Enhanced Infant Safety**
   The system provides continuous real-time monitoring of vital and environmental parameters such as temperature and humidity, enabling early detection of signs of distress or health abnormalities. This reduces the risk of Sudden Infant Death Syndrome (SIDS) and other infant health issues.

2. **Reduction of Parental Anxiety**
   Immediate alerts and notifications allow parents to remotely monitor their infant's status, significantly alleviating the psychological burden associated with constant supervision.

3. **Support for Modern Working Parents**
   the automated response to infant crying through a motorized cradle mechanism mimics traditional soothing methods, reducing the need for constant physical intervention and helping working parents manage their time more effectively.

4. **Optimization of Sleep Environment**
   by maintaining optimal ambient conditions such as temperature and humidity, the system ensures a stable and comfortable sleeping environment, thereby improving infant sleep quality and reducing disturbances.

1.6 Project Methodology

The development of the IoT-Driven Smart Baby Care System follows a structured methodology designed to ensure reliability, efficiency, and scalability. This methodology is divided into distinct stages, each contributing to the successful realization of the project's objectives.

1. Research and Requirements Gathering

Conduct a thorough analysis of existing baby monitoring systems and relevant IoT technologies.

Identify the core challenges in infant care, such as environmental monitoring (temperature, humidity), cry detection, wetness detection, and the need for real-time alerts.

Define functional and non-functional requirements, focusing on data accuracy, real-time notifications, and automated soothing actions.

Select appropriate hardware components, including the DHT11 sensor (temperature, humidity), ELECHOUSE Voice Recognition Module V3, rain sensor (used as a wetness sensor), Arduino Uno microcontroller, servo motor, GSM module, and ESP32-CAM module for live video streaming.

Determine suitable software tools for programming and data handling, primarily using the Arduino IDE and SMS-based notifications via GSM.

2. System Design

Develop a high-level architecture of the system, outlining how the sensors, Arduino Uno, and communication modules (GSM and ESP32-CAM) interact.

Design data flow diagrams to represent how sensor data is collected, processed, and used to trigger automated responses and caregiver alerts.

Create conceptual mockups for SMS-based notifications to ensure clarity and ease of use for caregivers.

Define safety measures and error-handling mechanisms to maintain system stability and ensure infant safety.

3. Hardware Development

Procure and assemble the selected hardware components, connecting them to the Arduino Uno using breadboards and necessary wiring.

Configure sensors for accurate data collection and set up the servo motor for automated cradle rocking based on the baby's cries.

Program and test individual sensors and modules separately to ensure they meet the required accuracy and reliability standards.

4. Software Development

Develop Arduino code to read sensor data, process it to detect critical conditions (baby crying, wet diaper, high temperature), and control the servo motor for rocking.

Implement GSM-based SMS notifications to instantly alert caregivers when specific events are detected.

Develop and test separate code for the ESP32-CAM module to provide live video streaming, ensuring it operates independently from the main Arduino-based system.

5. Integration and Testing

Integrate all hardware and software components into a cohesive system that can operate in real-time.

Conduct testing in simulated caregiving scenarios to validate system performance, including data accuracy, responsiveness of the cradle rocking, and reliability of SMS alerts and video streaming.

Debug and optimize the integrated system to ensure it functions reliably and consistently under real-world conditions.

1.7 Project Plan

The IoT-Driven Smart Baby Care System will be developed and implemented following a structured project plan, divided into sequential phases to ensure a smooth and systematic workflow. The project plan is outlined as follows:

Phase 1: Requirement Analysis and Research

Identify all hardware and software components required for the system, including sensors, the Arduino Uno microcontroller, GSM module, ESP32-CAM module (for video streaming), and additional actuators such as the servo motor.

Conduct background research on system integration strategies and similar IoT-based baby monitoring solutions.

Define the system's functional and performance requirements, including cry detection, wetness detection, environmental monitoring, cradle rocking, and remote communication features.

Phase 2: System Design

Design the architecture of the system, detailing how the sensors, actuators, Arduino Uno, and communication modules will interact to deliver real-time monitoring and automated responses.

Create a block diagram that represents the data flow from sensors and actuators to the caregiver's interface via GSM-based SMS alerts and ESP32-CAM video streaming.

Develop real-time SMS alerts and video streaming to ensure caregivers receive immediate updates and remote monitoring access. By integrating instant notifications with live visual feedback, the system enhances infant safety and caregiver responsiveness without relying on additional platforms.

Phase 3: Hardware Integration

Procure and test all sensors and modules, including the DHT11 (for temperature and humidity), Voice Recognition Module (for cry detection), rain sensor (for wetness detection), GSM module (for SMS alerts), and servo motor (for cradle rocking).

Assemble and connect all hardware components to the Arduino Uno microcontroller, using a breadboard or PCB for initial integration.

Conduct initial hardware testing to ensure sensor data accuracy and reliable component operation.

Phase 4: Software Development

Develop Arduino code to interface with sensors and actuators, process sensor data, and trigger automated actions (such as cradle rocking upon cry detection).

Implement GSM communication code to send real-time SMS alerts to caregivers when critical events are detected.

Separately program the ESP32-CAM module for live video streaming, as it operates independently from the main system.

Ensure modular and maintainable code to facilitate future enhancements.

Phase 5: Testing and Validation

Conduct comprehensive system-level testing to verify that all integrated components work together smoothly and meet the defined functional requirements.

Simulate different scenarios (e.g., baby crying, temperature/humidity, wetness detection) to validate sensor accuracy and system reliability.

Debug and optimize both the hardware and software to ensure stability, responsiveness, and long-term reliability in real-world environments.

1.8 Project Constraints

The IoT-Driven Smart Baby Care System faces several constraints that can affect its development and deployment. These constraints are categorized into technical, operational, and financial aspects, detailed as follows:

1. Technical Constraints

Hardware Limitations:

The Arduino Uno microcontroller has a limited number of GPIO pins and processing capabilities, which may restrict the number of sensors and actuators that can be connected simultaneously. This can impact system scalability and performance when multiple modules are active at once.

Sensor Accuracy and Reliability:

Affordable sensors such as the DHT11 for temperature and humidity monitoring and the sound sensor for cry detection may have limited precision in real-world environments, leading to occasional measurement errors or false alerts.

Data Transmission and Connectivity:

While the main system's sensors and actuators work together on the Arduino platform, the ESP32-CAM module was programmed and tested separately for live video streaming. Since it is not integrated with the other modules, real-time video updates are not automatically linked to the other system alerts.

Power Management

Continuous operation of the servo motor (for cradle rocking) and wireless modules (GSM for SMS notifications) increases power demands. Ensuring a stable and reliable power supply is essential to maintain consistent operation.

2. Operational Constraints

System Integration Challenges:

The main system components (sensors, GSM module, and servo motor) were successfully integrated to work together in real time. However, the ESP32-CAM operates separately and was not combined with the other system modules, limiting a fully unified solution.

Maintenance and Calibration:

Sensors and actuators require periodic calibration and maintenance to ensure long-term accuracy and reliability. This involves some operational effort from caregivers or technical staff.

Environmental Influences:

External factors like temperature fluctuations, humidity, and physical interference in the baby's environment can affect sensor readings and the overall system performance.

3. Financial Constraints

Budgetary Limitations:

While the project uses cost-effective components, modules like the GSM and ESP32-CAM can add to the total cost. This requires careful budgeting to ensure project affordability and sustainability.

1.9 Literature Review

This literature review aims to explore technologies used in monitoring children using sensors and cameras, identifying challenges and privacy related to this topic and previous studies on the topic.

1. Technologies Used in Child Monitoring

Different technologies are used in the baby monitoring system, including smart cameras and various types of sensors such as temperature and sound sensors. These technologies are essential for the efficiency and effectiveness of the system.

Smart Cameras: Cameras used to monitor children provide continuous visual surveillance and can be integrated with other systems to analyze children's behavior, such as detecting dangerous situations or signs of distress. Many studies have explored the use of smart cameras for this purpose.

Temperature Sensors: These sensors are used to monitor the child's body or room temperature to ensure it remains within a safe and comfortable range. Alerts can be sent when abnormal temperatures are detected, helping prevent heat stress or cold exposure.

Voice Recognition Module: Voice Recognition Module detect crying or unusual noises. They help identify when a baby is distressed or in need of attention, triggering alerts for caregivers or activating automatic calming systems.

Humidity Sensors: These sensors monitor moisture levels in diapers and notify caregivers when a change is needed. This ensures the child's comfort and helps prevent skin irritations or infections.

2. Challenges and Ethics:

There may be some challenges when using this system, including:

- Reducing Parents liability: Using such systems may make parents feel more secure and overly dependent on them.
- Privacy and security: Maintaining data privacy and security is one of the potential challenges.

Chapter Two: System Analysis

2.1 Overview

This chapter provides a technical and functional analysis of the IoT-Enabled Smart Baby Monitoring and Rocking System. The main goal of this section is to thoroughly examine the proposed model, considering the requirements for successful implementation, as well as the economic, technical, and operational feasibility of the system. This analysis aims to establish a clear roadmap for understanding, designing, and implementing the system effectively.

2.2 Proposed Model

The proposed system combines IoT sensors, edge computing devices, and cloud-based analytics to enable real-time monitoring and control of a smart baby care environment. The key components include:

IoT Sensors: These sensors continuously monitor critical parameters such as temperature, sound (cry detection) and humidity, providing comprehensive insight into the baby's comfort and safety.

Edge Devices: The ESP32 microcontroller acts as the edge device, locally processing and filtering sensor data to reduce latency and ensure quick response, such as triggering the rocking motor or sending alerts.

The cloud acts as a central hub for securely storing data and running advanced analytics, including trend analysis and automated notifications to caregivers via SMS alerts. It ensures efficient data processing and real-time access to critical information, enhancing responsiveness and system reliability.

This layered architecture supports scalability, reliability, and efficiency, making it suitable for home use to enhance infant care through automation and remote supervision.

2.3 System Requirement

The system requirements define the detailed functional and non-functional specifications.

2.3.1 Functional Requirements

The IoT-Driven Smart Baby Care System integrates various sensors and modules to automate infant care and monitoring effectively. Key functions include:

1. Servo Motor Function

The servo motor provides precise angular motion to swing the cradle automatically when a baby cries. It moves the cradle in a smooth, semi-circular motion, calming the baby.

2. Wet Sensing Sensor Function

The Rain sensor detects wet diapers and sends data to the Arduino board, which processes the information and sends an SMS alert to parents via the GSM module. This ensures the baby stays dry and comfortable, promoting health and hygiene.

3. Voice Recognition Module Function

The Voice Recognition Module detects when the baby cries by recognizing sound levels above a predefined threshold. When triggered, the system activates the servo motor to swing the cradle and simultaneously sends a notification via GSM to alert parents.

4. Temperature and Humidity Monitoring

The DHT11 sensor continuously monitors the baby's environment, ensuring temperature and humidity are within optimal ranges. Any deviation sends a real-time SMS notification to parents, ensuring timely corrective action.

5. GSM Module Function

The GSM module is the communication hub of the system. It sends SMS alerts to parents for various conditions, such as:

1. Crying detection.

2. Diaper wetness.

3. Unsuitable temperature or humidity levels.

This ensures that parents remain informed even when they are not physically present.

6. ESP32-CAM Function

The ESP32-CAM module provides live video streaming, enabling parents to visually monitor their child through a smartphone app, ensuring constant connectivity and reassurance.

7. Arduino Uno Function

The Arduino Uno acts as the system's core, integrating inputs from all sensors (sound, Rain, temperature, etc.), processing the data, and triggering appropriate actions like servo motor activation or GSM alerts.

2.3.2 Non-Functional Requirements

Usability:

•The system should have an intuitive design, minimizing the learning curve for users.

•Alerts and notifications should be clear and easily understood.

Reliability:

•The system must be reliable, with consistent operation and minimal downtime.

•High availability is critical to ensure the baby's safety.

Scalability:

•The design should support the addition of new features or sensors, such as air quality monitoring or AI-based behavioral analysis.

Maintainability:

•The system should have straightforward maintenance processes, including sensor replacements or firmware updates.

•Error detection mechanisms should enable quick troubleshooting.

Portability:

•The system should be easily deployable in different rooms or homes, requiring minimal reconfiguration.

Reusability:

•Components like sensors and the GSM module should be designed for use in other systems or projects with minimal modifications.

Flexibility:

•The system should allow modifications, such as sensitivity adjustments, feature upgrades, or integration with other smart home devices.

2.4 Hardware Requirements:

This system integrates various hardware components to achieve accurate real-time monitoring and reliable automated care for infants. The following components are utilized:

❖ **Jumper wire**

A jump wire – also known as jumper wire, or jumper – is an electrical wire, or group of electrical wires in a cable, with a connector or pin at each end – or sometimes without them, simply "tinned"–, which is used to interconnect the components of a breadboard or other prototypes together. Three types of jumping wires were used: female-to-female, female to male wires.

Figure 2.4.1: Wires

❖ **Cable USB**

USB Cable A Male to Type B printer is a USB male head used to connect the computer’s USB interface; the other end is the mouth of the mouth, connecting a USB printer, Arduino Uno boards, and Arduino mega. We can use it to connect any board to your computer's USB female A port, including Arduino Uno, Arduino Mega 2560, Arduino 101, and more.

Figure 2.4.2: Cable USB

❖ Arduino Microcontroller:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs light on a sensor, a finger on a button, or a Twitter message and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Fig 2.4.3: Arduino UNO

❖ ELECHOUSE Voice Recognition Module

ELECHOUSE Voice Recognition Module is a compact and easy-control speaking recognition board.

This product is a speaker-dependent voice recognition module. It supports up to 80 voice commands in all. Max 7 voice commands could work at the same time. Any sound could be trained as command. Users need to train the module first before let it recognizing any voice command.

This board has 2 controlling ways: Serial Port (full function), General Input Pins (part of function). General Output Pins on the board could generate several kinds of waves while corresponding voice command was recognized.

Figure 2.4.4: ELECHOUSE Voice Recognition Module

Voltage	4.5-5.5V
Current	<40mA
Digital Interface	5V TTL level for UART interface and GPIO
Analog Interface	3.5mm mono-channel microphone connector + microphone pin interface
Size	31mm x 50mm
Recognition accuracy	99% (under ideal environment)

Table 2.4: ELECHOUSE Voice Recognition Module Specifications

- **ESP32-CAM**

The ESP32-CAM is a compact, cost-effective development board built around the ESP32 microcontroller, featuring integrated Wi-Fi and Bluetooth connectivity along with an onboard camera interface. It is a key component in the IoT-Driven Smart Baby Care System, enabling real-time video streaming for remote infant monitoring. With support for camera modules like OV2640, external storage through a TF card slot, and multiple communication interfaces (UART, SPI, I2C), the ESP32-CAM allows for seamless data transmission and system integration. Its high-performance dual-core CPU, combined with onboard RAM and external PSRAM, ensures efficient image processing and low-latency streaming, making it highly suitable for smart surveillance applications in baby care.

Figure 2.4.5: ESP32-CAM

Microcontroller	ESP32 (Dual-core LX6, up to 160MHz)
Storage	TF Card Slot (max 4GB)
Power Supply	5V
Dimensions	27 × 40.5 × 4.5 mm

Table 2.4.2: Specifications of the ESP32-CAM

❖ Servo motor

A servo motor is a highly specialized motor designed for precise control of rotary or linear motion. It's a rotational or translational motor that employs a feedback mechanism to ensure exact positioning, typically using a control signal that dictates the motor's movement to a desired position. This mechanism allows for precise control of various components, making servo motors crucial in applications where precise positioning and smooth motion are required. Their reliability, compact size, and ability to hold specific positions make them a popular choice for various mechanical and IoT applications.

Figure 2.4.6: Servo motor

Feature	**Details**
Type	SG90 Micro Servo Motor
Operating Voltage	4.8V – 6V
Torque	1.8 kg·cm at 4.8V
Control Signal	PWM (Pulse Width Modulation)
Dimensions	22.2 × 11.8 × 31 mm

Table 2.4.3: Servo Motor Specifications

❖ DHT11 sensor:

The DHT11 is a widely used digital sensor that integrates a resistive-type humidity sensing component and an NTC (Negative Temperature Coefficient) thermistor for temperature measurement. It features a calibrated digital signal output and connects to an internal high-performance 8-bit microcontroller to ensure reliable and accurate readings. Known for its simplicity, cost-effectiveness, and anti-interference capability, the DHT11 provides adequate precision for non-critical applications. It measures temperature from 0°C to 50°C and humidity from 20% to 90%, making it suitable for real-time environmental monitoring in IoT-based systems such as smart homes and baby care projects.

Figure 2.4.7: DHT11 Sensor

❖ GSM:

The SIM800L GSM Module is a compact and cost-effective solution for enabling cellular communication in embedded IoT systems. In the Smart Baby Care System, it is responsible for sending real-time SMS alerts to parents when critical events are detected, such as high temperature, humidity anomalies, or wet diapers. By utilizing the Global System for Mobile Communications (GSM) network, the module ensures reliable

Remote communication without dependence on local Wi-Fi infrastructure. The SIM800L supports quad-band frequencies, allowing it to function globally. It communicates with the Arduino Uno via serial (TX/RX)

Interface, making integration straightforward. The module operates at 3.7V–4.2V and requires a stable power supply to function correctly.

Figure 2.4.8: GSM Module

Feature	Details
Model	SIM800L
Voltage Range	3.7V – 4.2V
Antenna	External antenna included for signal boost
Operating Temperature	-40°C to +85°C

Table 2.4.4: GSM Module Specifications

❖ **Rain Sensor**:

The rain sensor is a device used to detect the presence of rain by measuring moisture levels. It typically includes a sensor board that detects water and a control module that processes the signals. When the rain sensor board becomes wet, its resistance decreases, triggering the control module to send an alert or activate a response. In the context of the Smart Baby Care System, the rain sensor can be utilized to monitor the baby's environment, detecting moisture or wetness. This could be helpful in alerting parents if the baby's diaper is wet, ensuring timely care.

Figure 2.4.9: Rain sensor

- **Breadboard**

  A breadboard is a construction base designed for prototyping circuits without the need for soldering. It is a reusable and easy to use element for creating temporary prototypes and experimenting with circuit design, as it also exists in various sizes.

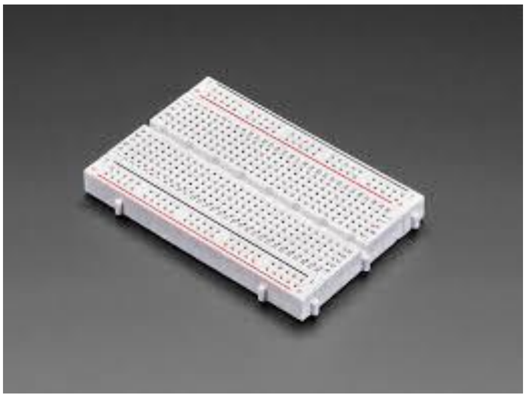

Figure 2.4.10: Breadboard

- **LCD Display (I2C Module):**

  The LCD 16x2 with I2C module is a character display that shows 2 lines with 16 characters each. The I2C interface significantly reduces the number of pins required for connection from 6 or more (in parallel) to just 2 (SDA and SCL), enabling easier integration with microcontrollers like Arduino. It operates at 5V, includes a built-in potentiometer for contrast adjustment, and is ideal for displaying real-time sensor data and system status in embedded applications.

Figure 2.4.11: LCD

- **Fan:**

The Fan plays a critical role in maintaining a comfortable environment for the baby by regulating the ambient temperature inside the crib or baby room. It is activated automatically when the temperature, measured by the DHT11 sensor, exceeds a predefined threshold. Controlled via a digital output pin from the Arduino Uno, the fan provides a simple but effective method to lower room temperature, especially during hot weather conditions. This automation ensures optimal thermal comfort without requiring manual intervention, which is particularly beneficial for working parents or during nighttime.

Figure 2.4.12: Fan

Feature	Details
Operating Voltage	5V or 12V
Control Method	Digital ON/OFF
Noise Level	Low (<30 dB)

Table 2.4.5: Fan Specification

- **Relay**

This is a simple single-channel relay module. Connect power and then switch the relay on by applying a low signal to the "IN" pin. The module has three screw terminal connections connected to the relay that include common, normally open (N.O.), and normally closed (N.C.).

This module doesn't include any mounting holes. Some mounting options would be to 3D print an enclosure for the module or to hot glue the bottom of the module or zip-tie it to a non-conductive surface.

Figure 2.4.13: Relay

Supply Voltage:	3.75 to 6 V
Supply Current with Relay De-Energized	2 mA
Supply Current with Relay Energized	70 to 72 mA
Input Control Signal	Active Low
Relay Max Contact Voltage	250 VAC or 30 VDC
Relay Max Contact Current	10 A

Table 2.4.6: Relay Specifications

2.5 Projects Costs

#	Item	Number	Price
1	Arduino UNO	1	35 NIS
2	Breadboard	1	15 NIS
3	ELECHOUSE Voice Recognition Module	1	190 NIS
4	Fan	1	20 NIS
5	Dht11	1	15 NIS
6	Servo motor	2	45 NIS
7	Wires	20	10 NIS
8	Rain sensor	1	15 NIS
9	GSM	1	60 NIS
10	Esp32 Cam	1	45 NIS
11	LCD I2C	1	25 NIS
12	Cable USB	1	10 NIS
13	Relay	1	6 NIS
Total			**526 NIS**

Table 2.5: Projects cost

2.7 Gantt Chart:

Is an essential project management tool that helps plan and organize tasks and activities visually. This plan allows the project to be divided into specific phases and tasks, specifying the time required for each task. By displaying the chronology of tasks, a Gantt chart helps identify interrelationships between different activities and clarify the dependencies between them.

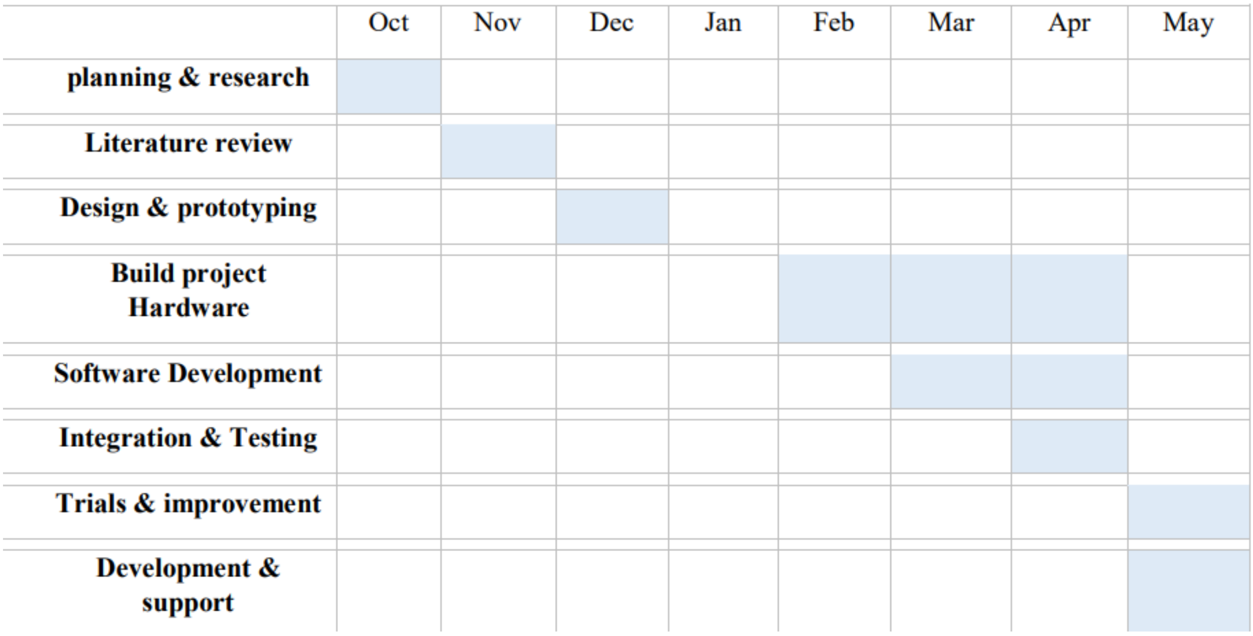

Figure 2.7: Gantt chart

Project Timeline and Workflow:

Our project follows a structured development process, utilizing a Gantt chart to track progress and manage interdependencies. Below is a breakdown of each phase:

- Planning & Research

The foundation of our project begins with strategic planning. We define

Objectives, break the project into smaller tasks, and allocate necessary resources.

During the research phase, we focus on understanding the challenges mothers face when caring for their children. Through surveys and discussions, we identified key concerns, leading to innovative ideas like incorporating a moisture sensor to detect diaper wetness.

- Literature Review

This phase involves an in-depth analysis of existing studies related to childcare technology. We compiled relevant findings, forming the basis of Chapter Two of our research documentation.

- Design & Prototyping

With a clear understanding of user needs, we moved on to sketching designs and drafting. This step ensures that our ideas are translated into practical and functional solutions.

- Development Phases

The subsequent months are dedicated to hardware construction, software

Development, integration, and rigorous testing. Each step contributes to refining our prototype, ensuring reliability and efficiency.

Our trial and improvement phase allows for continuous feedback and enhancements, leading to the final development and support phase, where we prepare for full implementation.

Chapter Three: System Functional Requirements

3.1 Overview

The IoT-Driven Smart Baby Care System is designed to monitor the baby's environment and behavior in real time and respond automatically to specific conditions. Its functional requirements define how the system collects data, processes it, and takes appropriate actions such as sending alerts or rocking the crib.

Key functions include cry detection, temperature and humidity monitoring, wetness detection, live video streaming, and automated cradle rocking. These features work together to ensure infant safety and provide caregivers with timely notifications and remote access.

2.3 Context Diagram

This context diagram presents the overall structure and external interactions of the Smart Baby Care System. The system is designed to monitor the baby's condition and environment in real time, providing automated alerts and responses to assist caregivers.

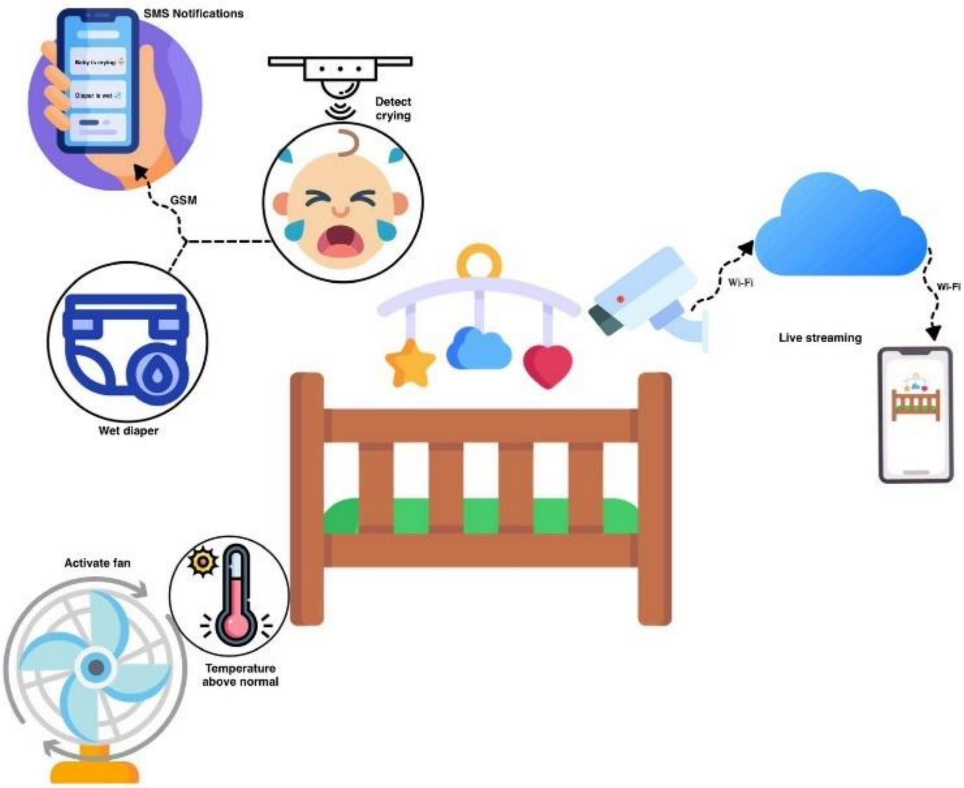

Figure 3.2: Context Diagram

The system includes the following main components:

- ✓ An ELECHOUSE Voice Recognition Module V3 detects when the baby is crying and triggers an SMS alert to the caregiver via the GSM module.
- ✓ A rain sensor is used to detect diaper wetness and also sends a notification through GSM.
- ✓ A temperature sensor (DHT11) monitors the surrounding environment. If the temperature exceeds the safe limit, the system activates a fan and notifies the caregiver.

An ESP32-CAM module provides live video streaming, sending footage via Wi-Fi to a cloud platform, which the caregiver can access through a smartphone.

The caregiver is the primary external actor, receiving:

- ✓ SMS notifications for real-time alerts.
- ✓ Live video access via a mobile device for remote monitoring.

This diagram summarizes how sensors, actuators, and communication modules are integrated to create a responsive and smart baby care system that enhances safety, comfort, and convenience.

3.3 Block Diagram

The block diagram illustrates the functional architecture of the Smart Baby Care System. This system integrates multiple components with the Arduino Uno to enable real-time monitoring and responsive care for infants. The rain sensor detects moisture and alerts parents via SMS through the GSM module. A sound sensor is used to detect a baby's cries, automatically triggering the rocking mechanism using a servo motor to soothe the infant. The DHT11 sensor continuously monitors temperature and humidity levels, with real-time data displayed on an LCD screen via I2C communication. Additionally, the ESP32-CAM module provides live video streaming, allowing parents to remotely observe their baby’s status. This smart system offers automation, timely alerts, and improved convenience for modern parenting.

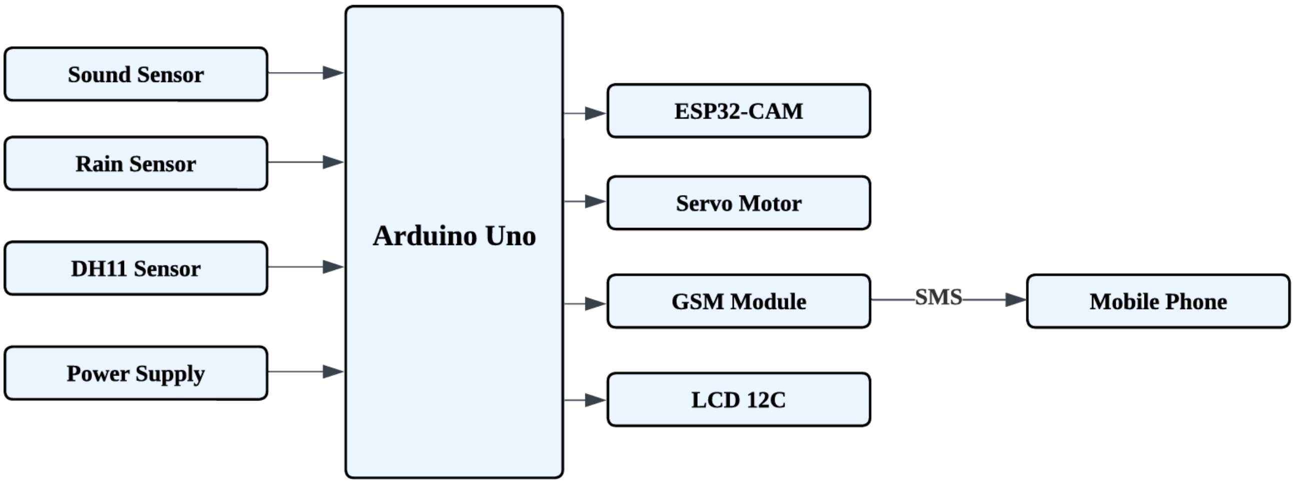

Figure 3.3: Block Diagram

3.4 Use-Case Diagram

The Use Case Diagram shown above represents the functional interaction between the external actors and the IoT-Driven Smart Baby Care System. This diagram is designed to illustrate the core functionalities that the system must support and how both human users and environmental triggers initiate or interact with these functions. It plays a crucial role in defining the system's functional boundaries and user expectations during the analysis and design phases.

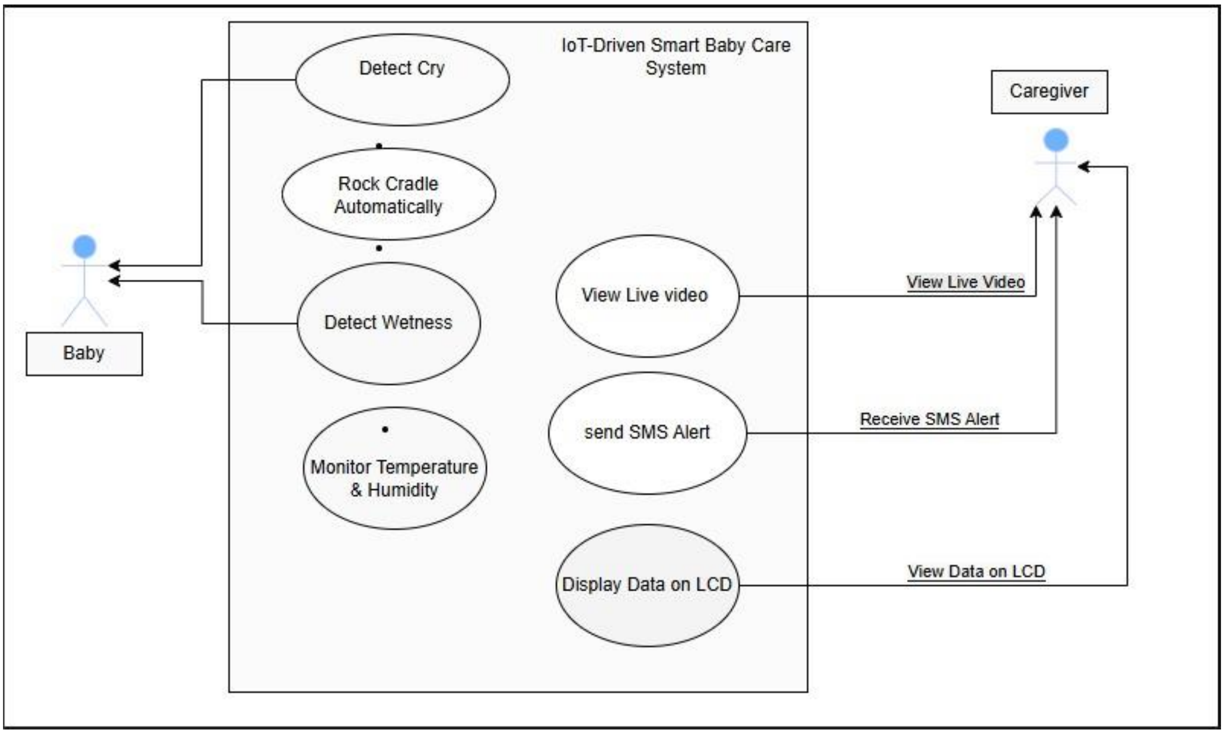

Figure 3.4: Use-Case Diagram

Actors in the System

1. **Baby (Passive Actor)**

The baby does not actively operate the system but serves as the primary trigger for most system functionalities. Events like crying, movement, or wetness are detected by sensors and used to initiate system actions.

2. **Caregiver (Active Actor)**

The caregiver is the system's end user who receives alerts, monitors live video, and views environmental data. This actor interacts with the system primarily through mobile notifications and live feedback.

Description

- Detect Cry: The system uses a Voice Recognition Module to detect the baby's cry. This triggers an alert or other system actions.
- Rock Cradle Automatically: A servo or stepper motor is activated automatically to rock the cradle once crying is detected.
- Detect Wetness :A rain sensor detects if the diaper is wet and initiates an alert.
- Monitor Temperature & Humidity :The DHT11 sensor continuously monitors the baby's environment to ensure safe conditions.
- Send SMS Alert :The system sends a real-time SMS alert via the GSM module in case of critical conditions (crying, wetness, temperature).
- Display Data on LCD : Real-time data such as temperature, humidity, and system status is displayed on an LCD screen.
- View Live Video :The ESP32-CAM module streams a live video feed, allowing the caregiver to remotely view the baby.

3.5 Functional Requirements Diagram

The Functional Requirements Diagram represents the core features and functionalities that the IoT-Driven Smart Baby Care System must perform to achieve its objectives of automated infant monitoring, comfort, and caregiver notification. This diagram organizes the key tasks of the system into clearly defined chains of input, processing, and output. It provides a structured view of how the system operates under different baby care scenarios, and serves as a foundational reference for the design, implementation, and validation of the system.

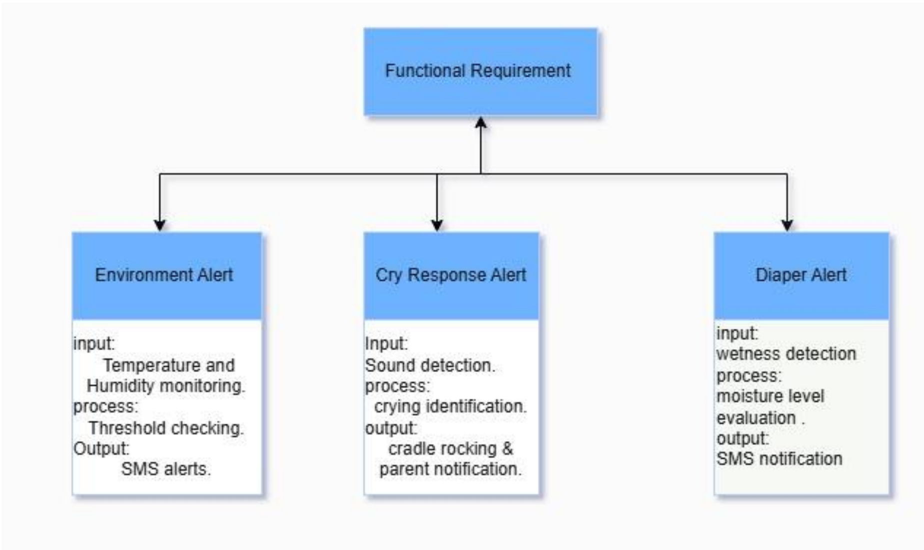

Figure 3.5: Functional Requirements Diagram

Description:

The Functional Requirements Diagram outlines three key automated response chains that define the behavior of the IoT-Driven Smart Baby Care System. The first is the Environment Alert, where the DHT11 sensor collects temperature and humidity data, which is then compared to safe threshold values. If abnormal conditions are detected, the system sends an immediate SMS alert to the caregiver, ensuring a comfortable and secure environment for the baby. The second function is the Cry Response Alert, in which sound input from a microphone is analyzed to detect crying patterns. Upon confirmation, the system simultaneously rocks the crib using a servo or stepper motor and sends an SMS notification to inform the caregiver. The third function, the Diaper Alert, relies on a rain sensor placed beneath the baby to detect moisture levels; once wetness is confirmed, the system triggers an SMS notification to prompt a diaper change. Together, these functional chains represent an intelligent, sensor-driven caregiving system that enhances real-time baby monitoring, automates essential comfort responses, and reduces the need for constant physical supervision by the caregiver.

3.6 Non-Functional Requirements Diagram:

The Non-Functional Requirements Diagram illustrates the key quality attributes of the system, such as performance, reliability, usability. These requirements define how the system operates, ensuring it meets user expectations and performs efficiently under various conditions.

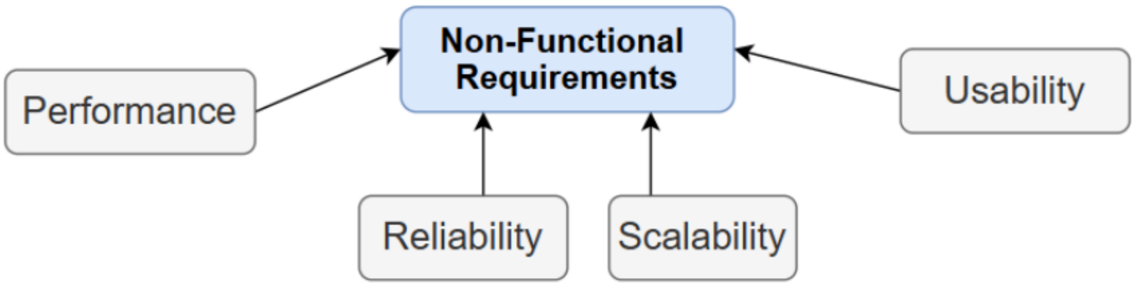

Figure 3.6: Non-Functional Requirements Diagram

The Non-Functional Requirements Diagram highlights the essential quality attributes that define the system's performance and user experience. Key aspects include performance, ensuring the system processes data in real time to provide immediate updates; reliability, ensuring high system availability to support uninterrupted operation. Scalability, enabling the integration of additional sensors or components as needed; and usability, ensuring the application is intuitive and user-friendly. These requirements collectively ensure that the system is robust, adaptable, and accessible to all users.

Chapter Four: System Design and Development

4.1 Overview

This chapter explores the essential aspects of designing and implementing the IoT-Driven Smart Baby Care System. It offers a comprehensive analysis of the system's operation, beginning with a Sequence Diagram that outlines the chronological flow of events. An Activity Diagram is used to illustrate the logical flow of actions and decision points within the system. Additionally, the Hardware Implementation section details the physical configuration and integration of key hardware components. Collectively, these elements present a clear picture of the system's functionality, workflow, and architectural structure, supporting the successful realization of the project objectives.

4.2 Sequence Diagram

A Sequence Diagram is used to represent the flow of data and interactions between various components of the system over time. In this project, the sequence diagram illustrates how sensor data—such as sound, temperature, humidity, and rain detection—is collected and processed by the Arduino Uno and ESP32-CAM. The data is either displayed locally on an LCD or transmitted remotely via the GSM module and live video stream. Notifications and alerts are sent to the parent's mobile device to ensure real-time awareness of the baby's condition. This diagram provides a clear visualization of the system's operational flow and communication between hardware and external interfaces in the IoT-Driven Smart Baby Care System.

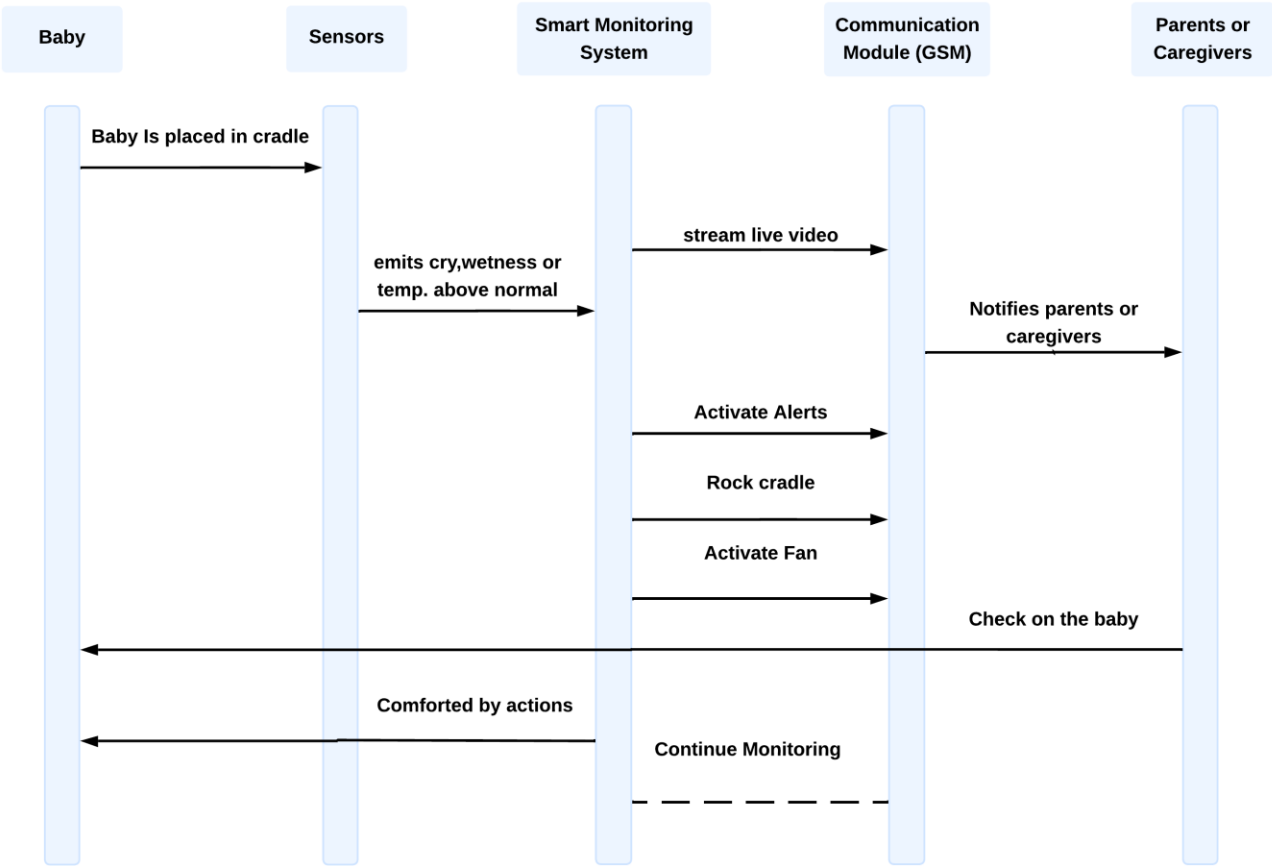

Figure 4.2: Sequence Diagram

4.3 Flowchart diagram:

The flowchart for the IoT-Driven Smart Baby Care System outlines the step- by-step process of how the system monitors and responds to the baby's needs. The process begins with environmental monitoring, where temperature and humidity sensors continuously track the room conditions. If the environment goes out of the optimal range, an alert is sent to the parents. Concurrently, the system monitors audio and visual cues, detecting sounds such as crying or cooing. When the baby cries, the system sends a notification to the parents and can automatically rock the cradle to soothe the baby. The system also monitors diaper wetness with sensors placed discreetly under the baby's sheet or mattress. If wetness is detected, an alert is sent to the parents, ensuring timely diaper changes. Throughout these processes, the system provides continuous live video streaming to keep parents informed about their baby's activities and surroundings.

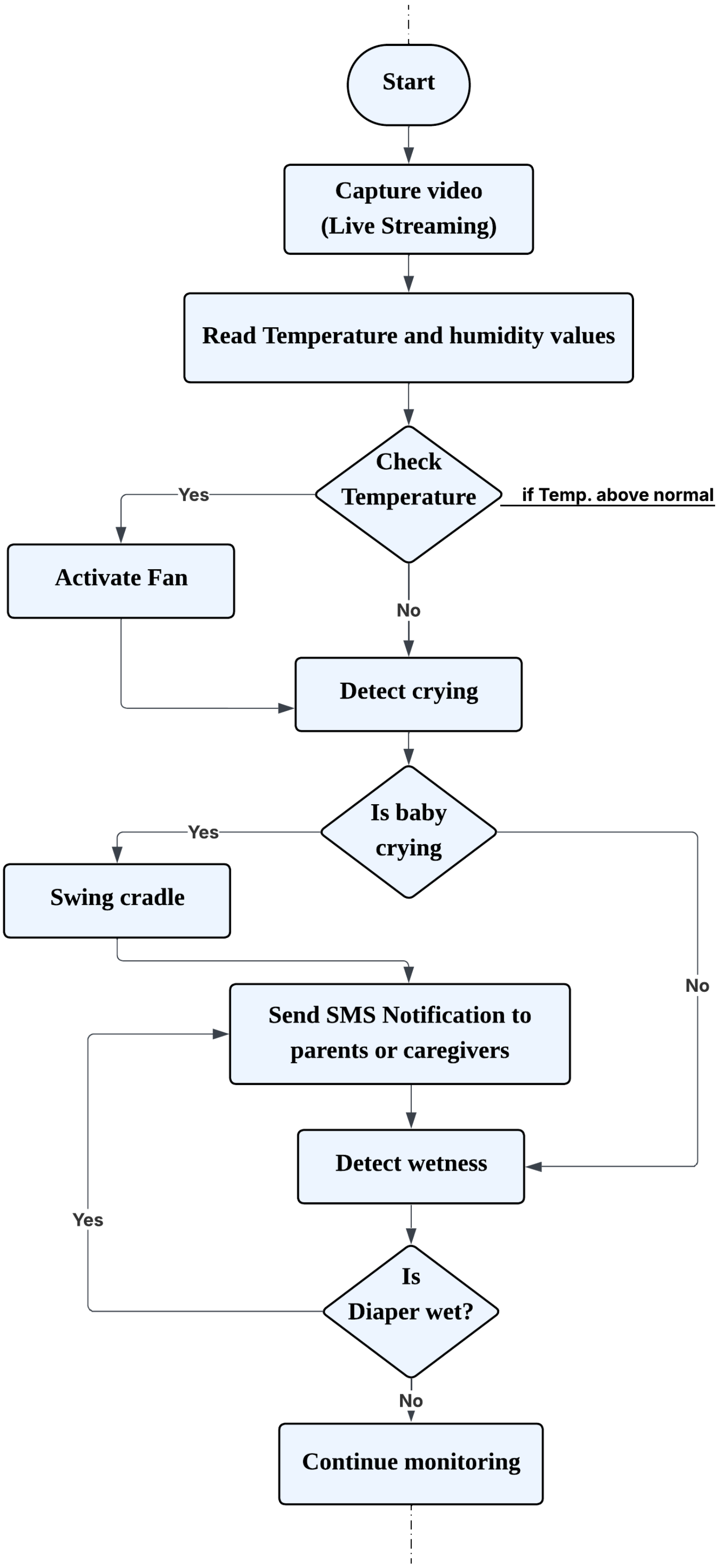

Figure 4.3: Flowchart Diagram

4.4 Hardware Implementation

The system was designed using various electronic components to ensure integration and efficiency. These components are interconnected to achieve stable and precise system performance.

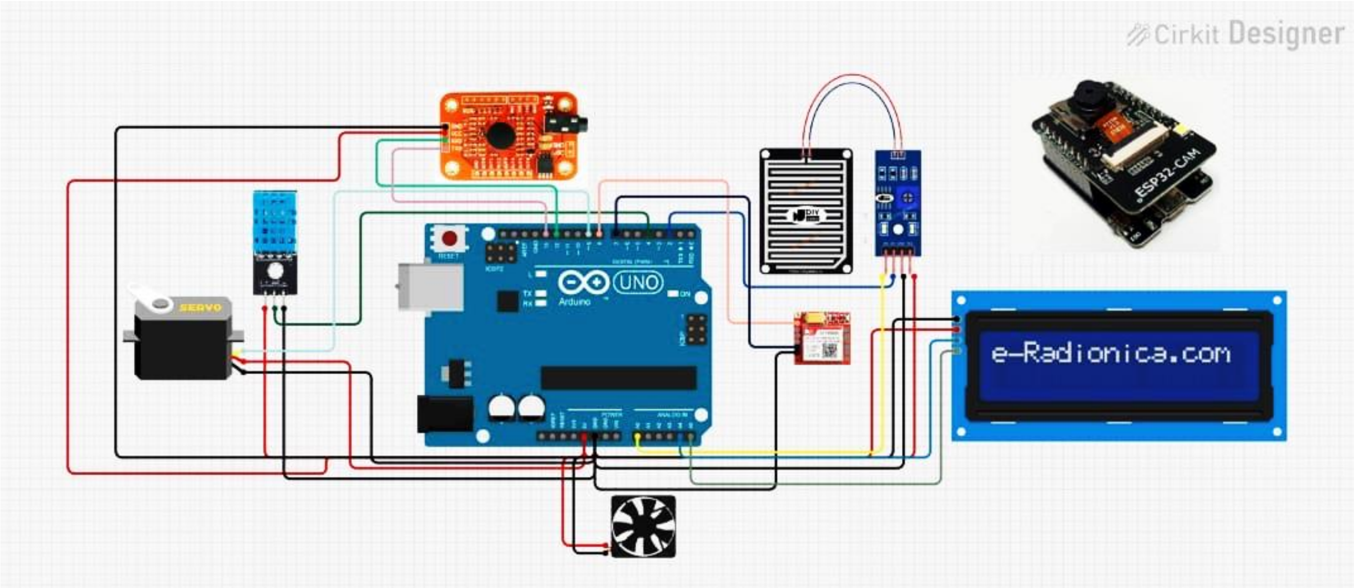

Figure 4.4: Hardware Implementation

The circuit diagram showcases the integration of various sensors, actuators, and modules working together to create an efficient Smart Baby Care System. Here is an expanded explanation of each component's role and connection in the project:

• Voice Recognition Module: this module analyzes vocal patterns, improving cry detection accuracy while minimizing false triggers from background noise

• Rain Sensor: Installed to detect diaper wetness, its output is connected to the Arduino's analog pin for variable moisture readings.

• DHT11 Sensor: Used for temperature and humidity monitoring, it connects to the Arduino through digital input pins.

• Servo Motor: Controls the swinging motion of the cradle, connected to a

PWM pin for smooth operation.

• GSM Module (SIM800L): Sends SMS notifications to a mobile phone when specific conditions are met, connected to the TX and RX pins for communication.

• ESP32-CAM: Streams live video for real-time monitoring, connected to the Arduino for control and external power for stability.

• LCD with I2C Module: Displays the system's status and sensor readings, interfaced via I2C communication (SDA and SCL pins).

• Fan: Activated when the temperature exceeds a set threshold. Connected to the Arduino via a digital pin using a transistor switch. Helps maintain a safe and comfortable environment for the baby.

5.1 Introduction

The software implementation involves Arduino programming to control various components, including temperature sensors, voice recognition, servo motors, and GSM-based SMS alerts. This system ensures real-time monitoring and response for a baby's environment, providing alerts for high temperature, wetness detection, and baby crying.

5.2 Code Explanation

The Arduino code is divided into several key parts:

- **Libraries Used**

The program utilizes the following libraries:

- Wire.h and LiquidCrystal_I2C.h for LCD display integration.
- DHT.h for temperature and humidity sensor (DHT11).
- Servo.h for servo motor control.
- SoftwareSerial.h and VoiceRecognitionV3.h for voice recognition.
- GSM communication via SoftwareSerial.h for SMS alerts.

- **Global Variables & Pin Definitions**

Several variables and constants define sensor pins, servo motors, voice command records, and message timing intervals.

- **Setup Function (setup())**

The setup() function initializes:

- LCD display and DHT sensor.
- Serial communication for debugging and GSM module.
- Pin modes for LED, rain sensor, relay (fan control).
- Voice recognition module records.

- **Loop Function (loop())**

The loop() function continuously monitors:

1. Voice Commands – Activating servo motors upon detecting a crying baby.
2. Temperature Sensor – Sending SMS alerts if the temperature exceeds 27°C and controlling the fan relay.
3. Wetness Detection – Checking diaper wetness and sending alerts if detected.
4. GSM Communication – Handling message transmission via sms() function.

- **Helper Functions**

– sms(text) – Sends alerts via GSM module.

– updateSerial() – Maintains communication between Arduino and GSM.

– printVR() – Displays voice recognition results.

5.3 Code Listing

Below is the full Arduino code implementation:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
 #include <SoftwareSerial.h>
#include <SoftwareSerial.h>
#include "VoiceRecognitionV3.h"
#include <Servo.h>      // مكتبة التحكم في السيرفو
Servo myServo;
Servo myServo2;
//Create software serial object to communicate with SIM800L
SoftwareSerial mySerial(3, 2); //SIM800L Tx & Rx is connected to
Arduino #3 & #2


LiquidCrystal_I2C lcd(0x27, 16, 9);

#define DHTPIN 9
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

unsigned long lastMsgTime = 0; // وقت آخر رسالة
const long msgInterval = 60000; // الفترة بين الرسائل (60 ثانية)
VR myVR(4,5);    // 2:RX 3:TX, you can choose your favourite pins.

uint8_t records[7]; // save record
uint8_t buf[64];
```

```
int led = 13;

#define onRecord    (0)
#define offRecord   (1)
int servoPin=7;
int servo2Pin=8;
int flag=0;
int rainPin = 10;     // البن الموصل عليه حساس المطر
int ledPin = 13;     // البن الموصل عليه LED
int RelayPin=12;
void setup() {
  lcd.init();
  lcd.backlight();
  dht.begin();

  pinMode(rainPin, INPUT);
  myServo.attach(servoPin);
  myServo2.attach(servo2Pin);    // توصيل السيرفو بالبن
  pinMode(ledPin, OUTPUT);
  pinMode(12, OUTPUT);  // مروحة على ريليه
  digitalWrite(12, LOW); // المروحة مطفأة بالبداية (الريليه فعالة عند LOW)

  Serial.begin(9600); // للمتابعة عبر السيريال


//Begin serial communication with Arduino and SIM800L
mySerial.begin(9600);

Serial.println("Initializing...");
delay(1000);

mySerial.println("AT"); //Once the handshake test is successful, it
will back to OK
updateSerial();

 myVR.begin(9600);

  Serial.println("Elechouse Voice Recognition V3 Module\r\nControl LED
sample");

  if(myVR.clear() == 0){
    Serial.println("Recognizer cleared.");
  }else{
    Serial.println("Not find VoiceRecognitionModule.");
    Serial.println("Please check connection and restart Arduino.");
    while(1);
  }
```

```
  if(myVR.load((uint8_t)onRecord) >= 0){
    Serial.println("onRecord loaded");
  }

  if(myVR.load((uint8_t)offRecord) >= 0){
    Serial.println("offRecord loaded");
  }


}

void loop() {
int ret;
  ret = myVR.recognize(buf, 50);
  if(ret>0){
    switch(buf[1]){
      case onRecord:
        /** turn on LED */
    sms("baby craing");
    myServo.write(90);
          myServo2.write(90); // تحريك السيرفو إلى زاوية 90 (مثلاً لهزّ السرير)
delay(1000);                // الانتظار 1 ثانية
myServo.write(0);
myServo2.write(0);           // العودة إلى الوضع الابتدائي

        break;

      case offRecord:
        /** turn off LED*/
        break;
      default:
        Serial.println("Record function undefined");
        break;
    }
    /** voice recognized */
    printVR(buf);
  }
  float temp = dht.readTemperature();

  if (isnan(temp)) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Error reading");
    lcd.setCursor(0, 1);
    lcd.print("temp from DHT!");
    } else {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temp: ");
```

```
    lcd.print(temp);
    lcd.print(" C");


    if (temp > 27 && (millis() - lastMsgTime > msgInterval)) {
      sms("Warning: High temperature detected in the baby's room");
      Serial.println("High temperature alert sent");
      digitalWrite(12, HIGH);
      delay(100); // منع تكرار الرسائل بسرعة
      lastMsgTime = millis();
      Serial.println(millis());
    }
  }

if (temp <= 27) {
  digitalWrite(12, LOW); // إيقاف المروحة
}

  // قراءة حساس المطر وتشغيل LED بدون التأثير على شاشة LCD
  int rainState = digitalRead(rainPin);

  if (rainState == LOW  && flag==0) {  // أو HIGH حسب الحساس عند البلل
    digitalWrite(ledPin, HIGH);
    sms("There is wetness in the diaper");
    Serial.println("Wet detected");
    flag=1;
  }
   if(rainState == HIGH  && flag==1){
 flag=0;
     digitalWrite(ledPin, LOW);

  }
  else {
    digitalWrite(ledPin, LOW);

  }
  updateSerial();
}

void sms(String text){


mySerial.println("AT+CMGF=1"); // Configuring TEXT mode
updateSerial();
mySerial.println("AT+CMGS=\"+972568740293\""); //change ZZ with country
code and xxxxxxxxxxx with phone number to sms
updateSerial();
mySerial.print(text); //text content
```

```
updateSerial();
mySerial.write(26);

}

void updateSerial() {
delay(500);
while (Serial.available()) {
mySerial.write(Serial.read()); //Forward what Serial received to
Software Serial Port
}
while (mySerial.available()) {
Serial.write(mySerial.read()); //Forward what Software Serial received
to Serial Port
}
}
void printSignature(uint8_t *buf, int len)
{
  int i;
  for(i=0; i<len; i++){
    if(buf[i]>0x19 && buf[i]<0x7F){
      Serial.write(buf[i]);
    }
    else{
      Serial.print("[");
      Serial.print(buf[i], HEX);
      Serial.print("]");
    }
  }
}

void printVR(uint8_t *buf)
{
  Serial.println("VR Index\tGroup\tRecordNum\tSignature");

  Serial.print(buf[2], DEC);
  Serial.print("\t\t");

  if(buf[0] == 0xFF){
    Serial.print("NONE");
  }
  else if(buf[0]&0x80){
    Serial.print("UG ");
    Serial.print(buf[0]&(~0x80), DEC);
  }
  else{
    Serial.print("SG ");
    Serial.print(buf[0], DEC);
```

```
  }
  Serial.print("\t");

  Serial.print(buf[1], DEC);
  Serial.print("\t\t");
  if(buf[3]>0){
    printSignature(buf+4, buf[3]);
  }
  else{
    Serial.print("NONE");
  }
  Serial.println("\r\n");
}
```

ESP32-CAM code:

```
#include "esp_camera.h"
#include <WiFi.h>
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#include "camera_pins.h"
const char *ssid = "iPhone";
const char *password = "Ff@f2020";

void startCameraServer();
void setupLedFlash(int pin);

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sccb_sda = SIOD_GPIO_NUM;
  config.pin_sccb_scl = SIOC_GPIO_NUM;
```

```
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.frame_size = FRAMESIZE_UXGA;
  config.pixel_format = PIXFORMAT_JPEG;  // for streaming
  //config.pixel_format = PIXFORMAT_RGB565; // for face
detection/recognition
  config.grab_mode = CAMERA_GRAB_WHEN_EMPTY;
  config.fb_location = CAMERA_FB_IN_PSRAM;
  config.jpeg_quality = 12;
  config.fb_count = 1;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG
quality
  //                      for larger pre-allocated frame buffer.
  if (config.pixel_format == PIXFORMAT_JPEG) {
    if (psramFound()) {
      config.jpeg_quality = 10;
      config.fb_count = 2;
      config.grab_mode = CAMERA_GRAB_LATEST;
    } else {
      // Limit the frame size when PSRAM is not available
      config.frame_size = FRAMESIZE_SVGA;
      config.fb_location = CAMERA_FB_IN_DRAM;
    }
  } else {
    // Best option for face detection/recognition
    config.frame_size = FRAMESIZE_240X240;
#if CONFIG_IDF_TARGET_ESP32S3
    config.fb_count = 2;
#endif
  }

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  sensor_t *s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit
saturated
```

```
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1);        // flip it back
    s->set_brightness(s, 1);   // up the brightness just a bit
    s->set_saturation(s, -2);  // lower the saturation
  }
  // drop down frame size for higher initial frame rate
  if (config.pixel_format == PIXFORMAT_JPEG) {
    s->set_framesize(s, FRAMESIZE_QVGA);
  }

#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

#if defined(CAMERA_MODEL_ESP32S3_EYE)
  s->set_vflip(s, 1);
#endif

// Setup LED FLash if LED pin is defined in camera_pins.h
#if defined(LED_GPIO_NUM)
  setupLedFlash(LED_GPIO_NUM);
#endif

  WiFi.begin(ssid, password);
  WiFi.setSleep(false);

  Serial.print("WiFi connecting");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}

void loop() {
  // Do nothing. Everything is done in another task by the web server
  delay(10000);
}
```

Chapter 6: Testing

This chapter outlines the testing process conducted to evaluate the performance, reliability, and accuracy of the IoT-Driven Smart Baby Care System. The system was tested under various conditions to verify that all components and features operate as expected, ensuring the safety and comfort of infants through continuous monitoring and automated responses.

6.1 Testing Objectives:

The primary objectives of testing were to:

•Validate sensor functionality (wetness, temperature, sound)

• Confirm proper operation of actuators (fan, cradle, toy hanger)

• Ensure timely and correct notification delivery to parents

• Verify stable and continuous video streaming

• Assess overall system performance in real-time conditions

6.2 Test Environment and Tools:

The testing was conducted in a controlled environment simulating real-world scenarios. The hardware setup included a microcontroller, wetness sensor, temperature sensor, sound sensor, fan, servo motor (for cradle), and a camera module. The software components included firmware programmed in Arduino IDE and notification reception.

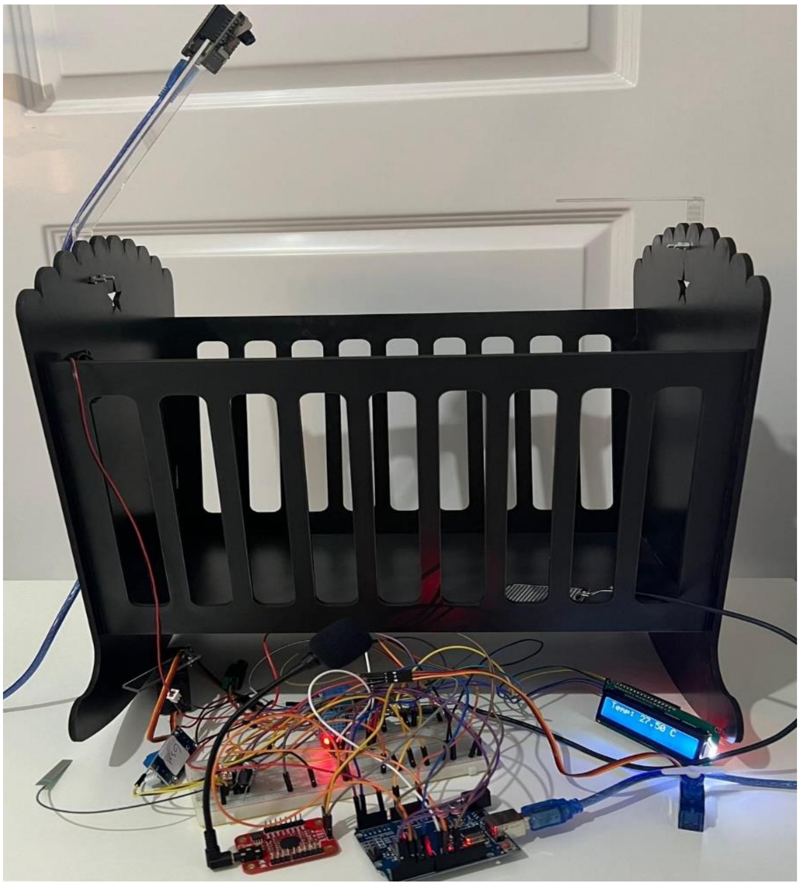

Figure 6.1: IoT-Driven Smart Baby Care System

Test Cases and Results:

Test Case ID	Feature	Input Condition	Expected Output	Actual Output	Result
T01	Diaper Wetness	Wetness sensor triggered	SMS Notification sent to parent	SMS Notification received	Pass
T02	Cry Detection	Baby crying sound detected	Cradle rocks, SMS alert sent, toy hanger moves	Cradle activated, SMS alert received	Pass
T03	Temperature Check	Temp > 25°C	Fan turns on	Fan activated	Pass
T04	Video Streaming	System powered on	Continuous live stream	Stable video feed	Pass
T05	Multi-trigger	Wet diaper + Crying + High Temp	Notifications + Fan + Cradle + Toy Hanger	All responses executed	Pass

Table 6.1: Test Cases and Results

6.3 Screenshots and Visual Results

Visual documentation was captured during testing, including:

- Notifications received for wetness, crying and high temperature

Alerts for wetness, crying, and high temperature successfully triggered.

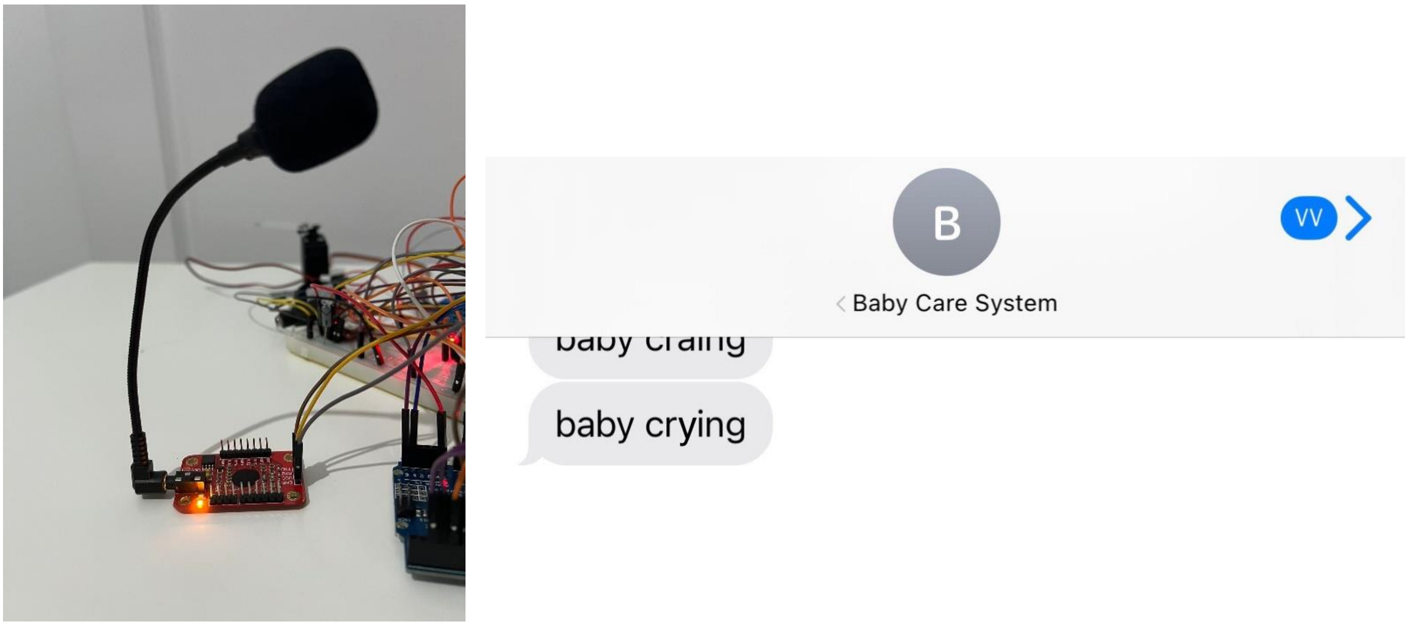

Figure 6.2: Notifications Received – Alerts for Crying

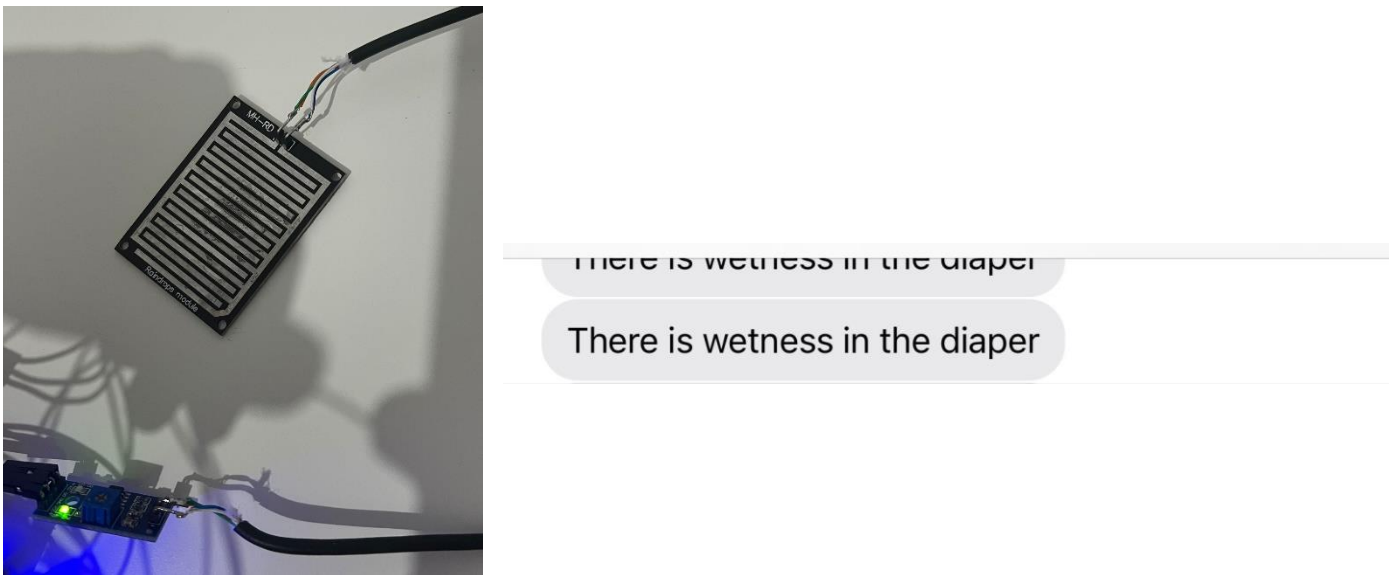

Figure 6.3: Notifications Received – Alerts for wetness

Figure 6.4: Notifications Received – Alerts for High Temperature

- Cradle rocking mechanism in action

The cradle rocking feature is fully functional, but we couldn’t capture it as an image in the test documentation.

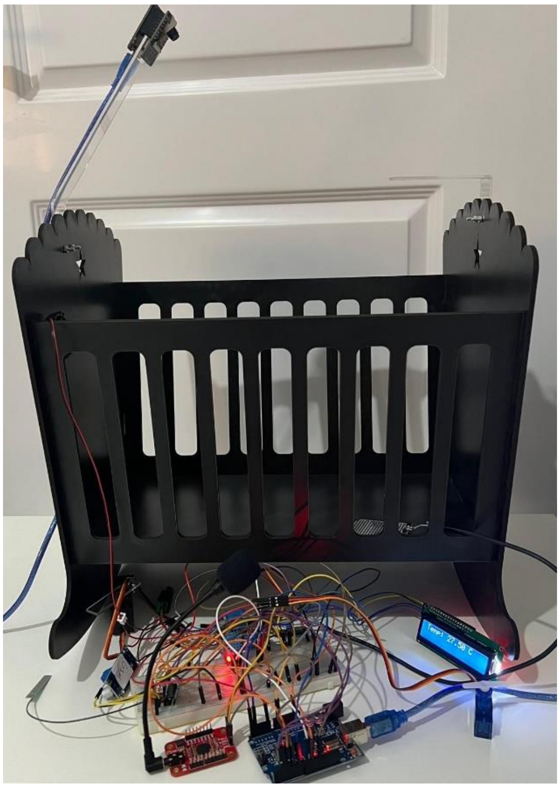

Figure 6.5: Cradle Rocking Mechanism

- Fan operating when high temperature detected

Automatically activates when high temperature is detected, ensuring baby comfort.

Figure 6.6: Fan Operation

- Live video feed displayed

Real-time streaming displayed, allowing remote monitoring

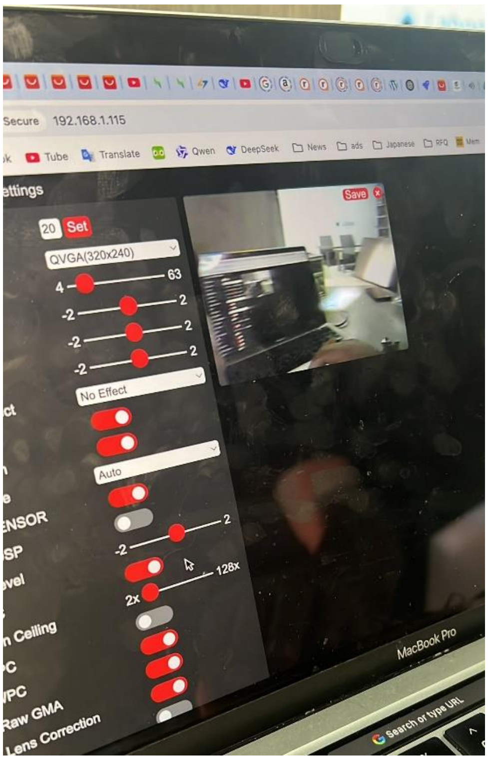

Figure 6.7: Live Video Feed

The testing results confirm that the IoT-Driven Smart Baby Care System performs reliably under different conditions. All major functionalities were validated successfully, demonstrating the system's capability to enhance infant care through intelligent automation and real-time monitoring.

Chapter Seven: Conclusion and Future Works

7.1 Conclusion

The IoT-Driven Smart Baby Care System represents a cutting-edge solution to modern childcare challenges, combining IoT technologies with automation to deliver a safe, efficient, and intelligent baby monitoring system. By integrating sensors for environmental monitoring, cry detection, and diaper wetness, the system ensures real-time responsiveness to the baby's needs.

Features such as automated cradle rocking and GSM-based notifications provide both convenience and peace of mind for parents, even when they are away.

7.2 Future Works

Although the system met its intended objectives, there is still room for future improvements and enhancements. Some suggested future works include:

- **Mobile App Development:**
  Creating a dedicated custom mobile application instead of relying on third-party platforms like Blynk, to improve user experience and system security.
- **Battery-Powered Operation:**
  Developing a version of the system that runs on rechargeable batteries to ensure portability and uninterrupted operation during power outages.
- **Medical Monitoring Integration:**
  Expanding the system to include medical sensors (heart rate, oxygen level) for more comprehensive child health monitoring.

References

1. Diagrams drawing tool:

   [Flowcharts: Lucidchart]

2. Icons:

   [Vector Icons and Stickers - PNG, SVG, EPS, PSD and CSS]

3. Arduino UNO datasheet:

   https://docs.arduino.cc/hardware/uno-rev3/

4. GSM Modem(SIM800)datasheet :

   https://components101.com/wireless/sim800l-gsm-module-pinout-datasheet-equivalent-circuit-specs

5. Servo motor datasheet:

   https://components101.com/motors/mg996r-servo-motor-datasheet

6. LCD I2C datasheet:

   https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf

7. ELECHOUSE Voice Recognition Module datasheet:

   https://www.elechouse.com/elechouse/images/product/VR3/VR3_manual.pdf

8. Rain sensor datasheet:

   https://www.datasheethub.com/rain-drops-%20detection-sensor-   module/

9. ESP32 camera datasheet:

   https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf

10. Relay datasheet:

    https://components101.com/switches/5v-single-channel-relay-module-pinout-features-applications-working-datasheet

11. Fan datasheet:

    https://www.farnell.com/datasheets/1878245.pdf

12. Website for diagram design:

    https://app.diagrams.net/

13. Kaittan, A. S., Hameed, S. M., Ali, N. K., & Ali, M. H. (Year). Smart management system for monitoring and control of infant baby bed. Department of Electrical Power and Machines Engineering, University of Diyala, Iraq; Department of Computer Engineering, University of Diyala, Iraq; Department of Electrical Engineering, Heteen Secondary School, Iraq.

14. Shirisha, K., Durgam, A., & Adidela, N. (Year). Smart Baby Cradle System using IoT. International Journal of Research Publication and Reviews. ISSN 2582-7421. Retrieved from www.ijrpr.com.

15. Pratap, N. L., Anuroop, K., Devi, P. N., Sandeep, A., & Nalajala, S. (2021, January). IoT based smart cradle for baby monitoring system. In 2021 6th International Conference on Inventive Computation Technologies (ICICT) (pp. 1298-1303). IEEE.

16. Basu, N., Kumar, R., & Sarma, A. (Year). Smart Monitoring System For Baby. This study discusses the importance of developing smart infant monitoring systems based on sensors and cameras. It examines how breathing, temperature, and movement can be monitored to provide accurate data in real time, enhancing child safety and reducing potential risks.

17. AI chatgbt

https://chat.openai.com/c/ebb17566-26a0-4950-b403-7dde2e33bdfa