University of Padua

Department of Mathematics “Tullio Levi-Civita”

Master Degree in Computer Science

MCCXXII

Real-Time Communications and Power Management for Battery-Powered Connected Devices.

Master Thesis

20/09/2024

Supervisor
Prof. Tullio Vardanega
Industrial Advisor
Engr. Francesco Zanella

Student
Matteo Mariani
Matr. 2029214

Accademic Year 2023-2024

“The most exciting phrase to hear in science, the one that heralds the most discoveries, is not "Eureka!" (I found it!) but ’That’s funny...”

— Isaac Asimov.

Acknowledgements

I would like to express my gratitude to my professor Tullio Vardanega, who provided invaluable guidance and support throughout throughout all this work, offering direction and encouragement when needed. His guidance was indispensable, from the initial choice of topic to the final stages of writing. I would like to thank my tutor at the company Francesco Zanella, for the great helpfulness and professionalism shown to me during these months of internship, for always supporting me and teaching me so much. Thanks also to all my colleagues at Vimar, for welcoming me and putting me at ease from day one, always ready to help me and give me advice. I would like to extend my gratitude to my parents for giving me the chance to pursue my aspirations and offering me continuous encouragement. I would also like to thank my lifelong friends for their support and for being there for me, even in the most challenging times. Special thanks to my fiancée Silvia, for always being my anchor, the person who most of all was able to understand and support me in difficult times. At last, I would like to take this opportunity to thank my brother for being my companion during my time in Padua. I may not have mentioned it often enough, but your presence and support have truly meant a great deal.

Padua, September 2024 *Matteo Mariani*

Abstract

The Internet of Things (IoT) is prompting significant advancements in the field of smart home technologies. As the demand for security monitoring increases, the need for robust real-time communication and effective power management, in devices that operate with limited power resources, has been the subject of considerable attention. In this thesis, we addressed these challenges by developing a smart peephole designed to provide real-time audio and video transmission, while seamlessly integrating with an existing enterprise smart home ecosystem. The research aims to identify an optimal, cost-effective platform, guarantee smooth and reliable audio-video transmission, and explore strategies for efficient power management. The provided solution successfully balances the need for responsive communication with the demands of battery efficiency, demonstrating how smart home devices can be integrated into a broader ecosystem without compromising performance or energy conservation.

Contents

1 Problem statement **1**
- 1.1 Context overview . . . 1
- 1.2 Objectives of this work . . . 2
  - 1.2.1 Fishbone diagram . . . 3
- 1.3 Market overview . . . 4
  - 1.3.1 Target market . . . 5
  - 1.3.2 Competitors overview . . . 6
  - 1.3.3 SWOT analysis . . . 9
- 1.4 Technical constraints . . . 10
  - 1.4.1 Constraints . . . 10

2 Technical challenges **13**
- 2.1 Hardware . . . 13
  - 2.1.1 Microcontroller vs microprocessors . . . 13
  - 2.1.2 Microcontrollers . . . 14
  - 2.1.3 Platform comparison . . . 17
- 2.2 Battery . . . 23
  - 2.2.1 State of charge . . . 25
  - 2.2.2 Approaches to SoC estimation . . . 26
- 2.3 Real-Time video streaming protocol . . . 29
  - 2.3.1 Protocols comparison . . . 30

3 Project outcomes **33**
- 3.1 Case study . . . 33
  - 3.1.1 Project needs and requirements . . . 35
- 3.2 Development platforms . . . 37
  - 3.2.1 Software development kit . . . 37
  - 3.2.2 DevKit board . . . 41
- 3.3 Product demo . . . 44
  - 3.3.1 Cyclop . . . 45
  - 3.3.2 Audio device . . . 46
- 3.4 Power management . . . 48
  - 3.4.1 Hardware . . . 49
  - 3.4.2 Sleep modes . . . 51
  - 3.4.3 Communication protocol . . . 56
  - 3.4.4 Proposed solution . . . 59
  - 3.4.5 Results . . . 62

4 Conclusions **71**
4.1 Retrospective 71
4.2 Future works 73

A Needs and Requirements **76**
A.1 Needs 76
A.2 Requirements 77

Bibliography **ii**

List of Figures

1.1 Smart Home Devices. 1
1.2 Fishbone diagram of the problem. 3
1.3 Smart home market in Italy in 2023. 5
1.4 Amazon Ring Door View Cam. 6
1.5 EZVIZ CP4. 7
1.6 Xiaomi Smart Doorbell 3. 7
1.7 SWOT Analysis 10

2.1 Microncontrollers and microprocessors applications. 14
2.2 Microncontrollers processing cycle. 15
2.3 Typical microncontrollers components. 16
2.4 Raspberry Pi Pico W Board. 19
2.5 STM32WB module. 19
2.6 ESP32 WROVER-E module. 20
2.7 Nordic nRF5340 SoC (no antenna). 21
2.8 Comparison of different battery types. 24
2.9 Battery State of Charge and Depth of Discharge relation. 25
2.10 Composite correction factor. 29

3.1 Current consumption declared by Espressif in automatic light sleep mode with frequency management enabled. 34
3.2 SDKs components. 38
3.3 Espressif IoT Development Framework. 39
3.4 Espressif Audio Development Framework. 40
3.5 Freenove ESP32-WROVER Board. 42
3.6 ESP32-LyraT V4.3 board layout overview. 43
3.7 ESP32-S3-Korvo-2 V3.0 board layout overview. 44
3.8 Cyclop main features. 45
3.9 Audio device main features. 47
3.10 Functioning ordering of fields (t) and (c) of the SDP offer. . . . 48
3.11 AMS1117 Pinout. 50
3.12 FireBeetle 2 ESP32-E. 50
3.13 ESP32 functional components. 51
3.14 Active mode. 53
3.15 Modem sleep. 53
3.16 Light sleep. 54
3.17 Deep sleep. 55
3.18 Hibernation. 56

3.19 Bluetooth Low Energy stack. 57
3.20 Communication between a GATT client and a GATT server. . . 59
3.21 Flowchart of polling method. 60
3.22 BLE connection interval. 61
3.23 Current consumption declared by Espressif in automatic light sleep mode with frequency management enabled. 62
3.24 Equipment used to measure energy consumption. 63
3.25 ESP32-S3-Korvo2 measurement WiFi/RTSP activation event. . 65
3.26 ESP32-S3-Korvo2 measurement of RTSP audio/video streaming event. 66
3.27 FireBeetle2 ESP32-E measurement of Automatic light sleep event. 67
3.28 Depth of discharge versus cycle life of the lithium-ion battery. . 69
3.29 Connection interval impact on energy consumption. 70

4.1 Comparison of devkit with optimized vs. non-optimized voltage regulator. 74
4.2 Current consumption declared by Espressif in automatic light sleep mode with frequency management enabled. 75

List of Tables

2.1 Microcontroller platform comparison. 22

3.1 Daily energy consumption 68

A.1 Needs group 1. 76
A.2 Needs group 2. 76
A.3 Needs group 3. 77
A.4 Requirements group 1. 77
A.5 Requirements group 2. 78
A.6 Requirements group 3 (Part 1). 78
A.7 Requirements group 3 (Part 2). i

Chapter 1

Problem statement

1.1 Context overview

In recent years, the idea of a ’smart home’ has become increasingly common, thanks to the advent of Internet of Things (IoT) technologies. A smart home is cyber-physical systems that integrate IoT devices, computers, and smart appliances, facilitating human interaction through in-home communication networks and the Internet [119]. It is a home equipped with internet-connected devices that enable the monitoring and control of various home systems such as lighting, temperature, security, often through a central device such as a smartphone or voice assistant (Fig. 1.1).

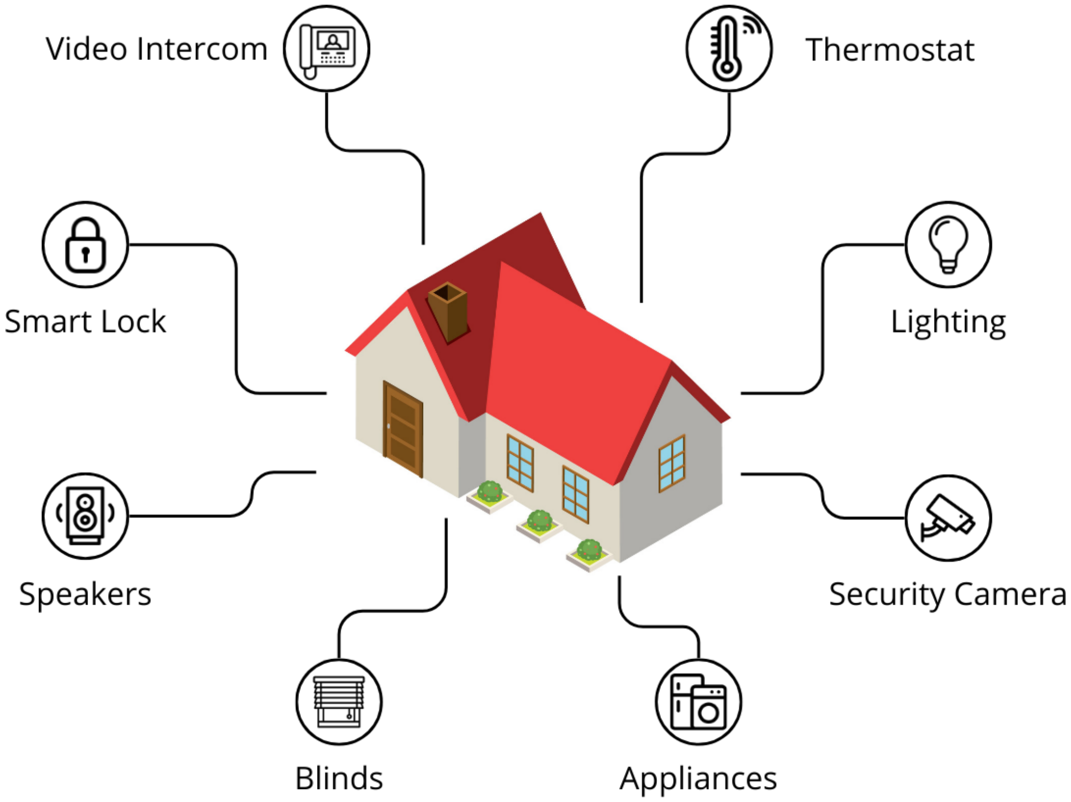


Figure 1.1: Smart Home Devices.

These smart devices offer a number of significant advantages:

- **Convenience**: Smart devices can be controlled remotely, allowing users to manage their homes even when they are away;
- **Energy efficiency**: The ability to monitor and control energy use in real time can help reduce energy consumption and costs;
- **Security**: Smart home technologies include advanced surveillance systems, such as security cameras and motion sensors, which significantly improve home security;
- **Automation**: devices can be programmed to perform certain tasks automatically, enhancing the comfort and convenience of daily life.

Nevertheless, as the adoption of smart home devices continues to expand, a number of new technical challenges have emerged. One of the most significant challenges is the necessity for real-time audio and video communications, which are crucial in home security and monitoring contexts where any latency can compromise the system's effectiveness. The transmission and reception of high-quality audio and video data in real time is a fundamental aspect of ensuring an enjoyable user experience. Additionally, a significant proportion of these smart home devices are designed to be wireless and are required to operate for extended periods on battery power. This necessitates the implementation of highly efficient energy management solutions. Ensuring that these devices maintain optimal performance while conserving battery life represents a complex problem that requires innovative approaches to power management. Traditional security solutions, such as mechanical peepholes, are no longer sufficient to meet the growing security and convenience needs of modern users.

In this context, Vimar identified a critical need: to develop a smart peephole solution that could be integrated into their existing technology ecosystem.

The development of a smart peephole is a convergence of diverse technological domains, including embedded systems, the Internet of Things (IoT), real-time communication protocols and power management. Embedded systems represent the fundamental component of these devices, ensuring both efficient and reliable operation. The Internet of Things (IoT) framework enables these devices to connect and communicate over the internet, thereby facilitating functionalities such as remote monitoring and control. The utilisation of real-time communication protocols ensures the transmission of audio and video streams with minimal latency, thereby providing a seamless user experience. Finally, power consumption must be carefully optimised in order to maintain the functionality and responsiveness of the smart peephole at all times, without the need for frequent recharging.

1.2 Objectives of this work

In the context of developing a smart peephole for smart home environments, it is evident that several critical objectives must be addressed in order to ensure that the device meets user needs and functions effectively within a connected home ecosystem.

1.2.1 Fishbone diagram

In order to address these challenges in a systematic manner, we employ a fishbone diagram, also known as an Ishikawa diagram or cause-and-effect diagram.

A fishbone diagram is a visual tool employed to facilitate a rigorous analysis of the root causes of a specific problem or effect [99]. The diagram is shaped like a fish's skeleton, with the problem statement at the "head" and the causes extending outwards like "bones" along the spine. Each major bone represents a broad category of potential causes, which are then broken down into more specific sub-causes. The identification of the various underlying causes contributes to a complex problem, rather than merely addressing the superficial symptoms. Furthermore, the fishbone diagram offers a visualisation of relationships, thus facilitating an understanding of the interconnections between different factors.

In this research, the fishbone diagram (Fig. 1.2) assists in the dissection and analysis of the principal challenges associated with the development of real-time communication and efficient energy management for battery-powered connected devices. Consequently, it facilitates the organisation of the primary objectives to be attained and the challenges that must be overcome in order to achieve them.

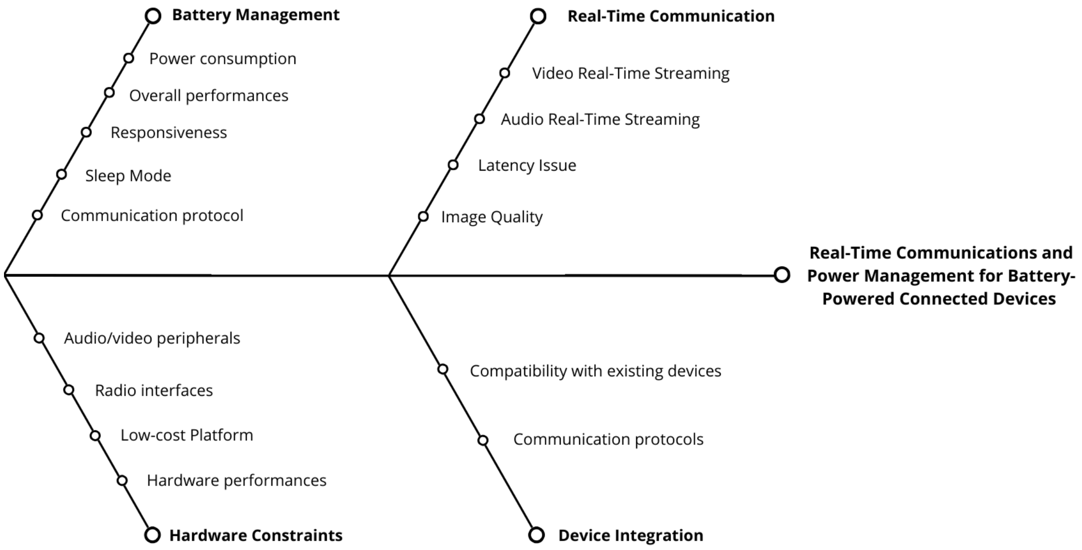


Figure 1.2: Fishbone diagram of the problem.

The first objective centers on ensuring high-quality real-time audio and video transmission. In smart home security contexts, the ability to transmit and receive high-quality audio and video data instantaneously is essential for a good user experience. Therefore, selecting the appropriate communication protocols that can handle the data requirements without significant latency is crucial. By doing so, the smart peephole can provide users with immediate and clear visual and audio information, enhancing the overall security and usability of the device.

The second objective is to optimise battery consumption without compromising the performance of the device, including maintaining a high responsiveness. As is the case with many other Internet of Things (IoT) devices, smart peepholes frequently necessitate periods of battery operation and maximum power [26]. The implementation of advanced power management techniques, which alternate between sleep and active phases, is essential to ensure that the device remains functional for as long as possible and is always responsive. This can be achieved through the use of sleep modes and low-power communication protocols. These techniques facilitate the extension of the device's battery life, thereby reducing the frequency of recharges or replacements.

In order to make the smart peephole accessible and scalable, it is essential to utilise a low-cost development platform. This necessitates the meticulous selection of cost-effective hardware that is nevertheless capable of supporting the requisite functionalities and performance standards. The optimal balance between cost and capability ensures that the device can be produced and distributed on a large scale, making it economically viable for both manufacturers and consumers. The utilisation of a low-cost platform also permits for a more extensive market penetration, facilitating the broader accessibility of smart home technologies.

The final objective is to ensure that the smart peephole can be integrated consistently with existing devices within the enterprise's technology ecosystem. This necessitates the assurance of compatibility with existing hardware and software, as well as adherence to established communication standards. Effective integration enhances the overall functionality of the smart home system, providing a cohesive and user-friendly experience [70]. Seamless integration permits users to readily incorporate the smart peephole into their existing smart home configuration, enhancing the system's overall security and convenience without necessitating significant alterations or additional financial outlay.

By addressing these objectives, this work explores the latest advancements in real-time communication protocols, power management techniques, and low-cost platform utilization, providing a holistic approach to overcoming the challenges inherent in developing a state-of-the-art smart home device.

1.3 Market overview

The Vimar smart peephole project is a proof of concept (POC) that has the ambition of becoming a commercially available product.

The Italian smart home market is projected to experience considerable growth in revenue, in 2023, the smart home market demonstrated continued growth, reaching a value of €810 million, representing a 5% increase compared to the previous year [89] (Fig. 1.3), reaching an estimated €1,242.0 million by 2024. It is forecast that this growth will continue at an annual rate of 9.28%, resulting in a projected market volume of €1,771.0 million by 2028 [105]. Italian consumers are demonstrating a growing interest in smart home devices that offer convenience, energy efficiency, and enhanced security. They also express a strong preference for seamless integration and control of household functions via mobile apps or voice assistants. One noteworthy trend in the Italian market is the growing prevalence of smart thermostats and lighting systems. Such products not only facilitate energy savings but also enable users to personalise their domestic environment in accordance with their preferences. Furthermore, there is a growing tendency to utilise smart security systems, including cameras and smart locks, as Italians increasingly prioritise the safety of their homes [105].

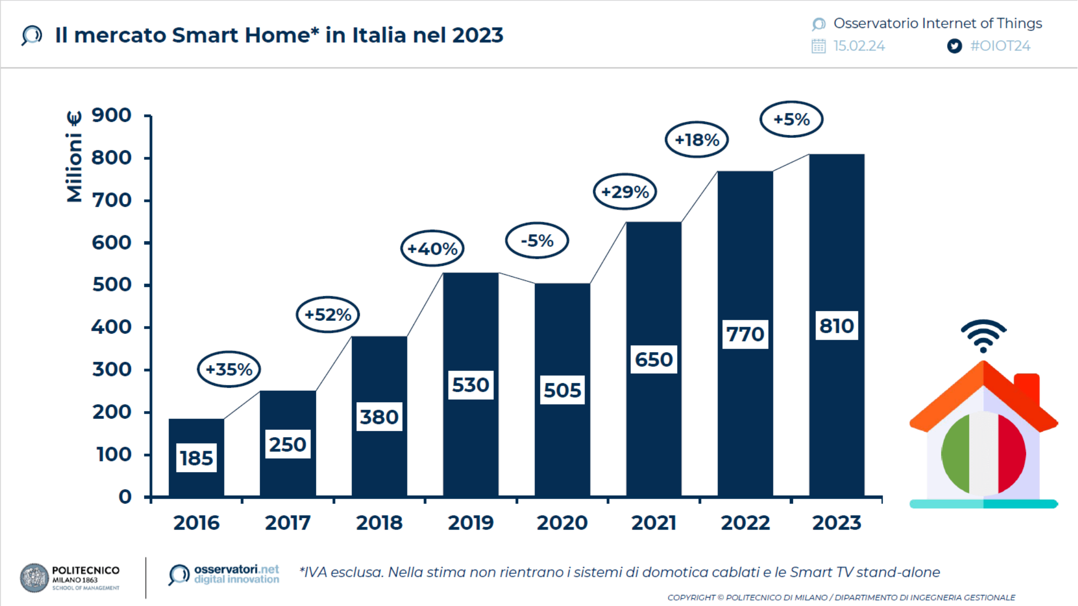


Figure 1.3: Smart home market in Italy in 2023.

Source [89]

The following section analyses the specifics of the market context in which the eventual product would be launched in order to identify its relative positioning within that context.

1.3.1 Target market

In 2023, in Italy, security solutions, including cameras, door/window sensors, and connected locks, led the smart home market, generating €195 million (24% of the market) and experiencing a growth rate of 30%, up from 20% in 2022 [89]. This specific growth in recent years is indicative of a growing awareness among consumers regarding the security and control of their homes.

The adoption of these technologies has transformed traditional homes into highly connected and integrated smart homes, thereby significantly improving the quality of life and safety of users. The market for smart peepholes represents a particularly promising sub-segment within the home security sector. These devices integrate the conventional functions of peepholes with advanced technological capabilities. In contrast to conventional peepholes, smart peepholes provide a transparent and precise view of the exterior of the door through a digital display and the capacity to observe and interact with visitors via smartphones and other connected devices. The Italian smart peephole market is characterised by increasing competitiveness and a diversification of supply. The principal market participants include companies that have developed a specialisation in home security solutions and major technology brands that have expanded their product range to encompass smart home solutions. Some of the most prominent brands offer models with advanced features, such as night vision and cloud recording, while others provide more economical options with a more limited range of capabilities.

1.3.2 Competitors overview

This section presents an analysis of the competitive market landscape in Italy, with a particular focus on the strategies employed by competitors in the distribution of smart peepholes. The objective is to identify the competitive elements of our product that will enable it to differentiate itself within the market.

Figure 1.4: Amazon Ring Door View Cam.

Source [8].

Amazon Ring Door View Cam. The Amazon Ring Door View Cam (Fig. 1.4) offers a convenient and straightforward solution for monitoring the entrance door. It is a smart camera designed to be installed in place of the traditional peephole, providing integration into existing door hardware. The device enables the user to observe, communicate with and interact with visitors via their smartphone, even when they are not present at the residence. The device is capable of recording high-definition video, detecting motion, and transmitting notifications in real time. Additionally, it is compatible with the Amazon Alexa ecosystem and is equipped with a rechargeable battery. It is notable that the product also presents certain disadvantages. It necessitates a subscription for the purpose of unlimited video recording, and furthermore, it exhibits a considerably higher energy consumption in comparison to traditional solutions [8].

The Amazon Ring Door View Cam is available for purchase at a recommended retail price of €129.99. Furthermore, supplementary subscription plans for cloud-based video recording are available for purchase at a cost of €3.99 per month.

Figure 1.5: EZVIZ CP4.

Source [43].

EZVIZ CP4. The EZVIZ CP4 (Fig. 1.5) is a Wi-Fi surveillance camera that is suitable for use in both indoor and outdoor settings. The camera is equipped with a 1080p HD resolution, an 112° viewing angle, and dual-band Wi-Fi connectivity. It also features a mobile application for iOS and Android devices, along with motion detection capabilities and push notifications. The camera offers both continuous and event-triggered video recording options, as well as cloud storage with an optional subscription. It is designed for both indoor and outdoor use, with an IP65 degree of protection. Additionally, the camera incorporates infrared night vision and a built-in microphone. Nevertheless, this device also exhibits certain disadvantages. These include a non-premium construction quality and materials, a requirement for a cloud subscription for advanced functionality, and a lack of integration with popular smart home ecosystems. The EZVIZ C4P is available for purchase at a recommended retail price of €134.99 [44].

Figure 1.6: Xiaomi Smart Doorbell 3.

Source [122].

Xiaomi Smart Doorbell 3. The Xiaomi Smart Doorbell 3 represents a low to mid-range option within the smart video door entry camera market. It offers a range of advanced features, including 2K video resolution, motion detection with push notifications, two-way audio/video communication, Wi-Fi connectivity and the ability to use the Mi Home mobile app, event-triggered video recording and infrared night vision. It is therefore an interesting alternative to more expensive solutions currently on the market. The product also displays significant defects, including inadequate reliability and manufacturing quality, an absence of integration with prevalent smart home ecosystems, prolonged response latency, and an incompatibility with external applications. The Xiaomi Smart Doorbell 3 is available for purchase at a recommended retail price of €59.99 [123].

A detailed restatement of project milestones Restating the thesis objectives and results Analysis of thesis objectives and outcomes Retrospective analysis Outcomes analysis
A comparative analysis of the principal competitors reveals a number of common elements and areas for improvement, which Vimar could exploit in order to establish a favourable position in the market.

The following common elements were identified:

1. **Video quality.** The majority of competitors offer high-definition (HD) or higher video resolutions, which is a standard feature in the market;
2. **Motion detection and notifications.** All of the analysed devices include motion detection and push notifications, which are essential for the purposes of security and monitoring;
3. **Design and installation.** The competitors display variation in terms of their design and ease of installation, with a growing tendency towards the provision of compact and straightforward installation solutions.

A critical analysis of the products offered by competitors reveals a number of gaps in the market.

1. **Battery life.** It is a common complaint among users that the battery life of many devices does not meet their expectations. Vimar could differentiate itself in the market by offering longer battery life or alternative or interchangeable power solutions according to user preferences, such as continuous power supply options via cable;
2. **Ecosystems integration.** All competitors are distinguished by their integration with their respective ecosystems, which confers a competitive advantage in instances where the customer already possesses products from the same company, but represents a disadvantage otherwise. Similarly, Vimar could develop a device that integrates with its extensive ecosystem, thereby encouraging users to prefer a solution that is seamlessly integrated with the system they are already familiar with and already possess, as opposed to a poorly integrated third-party solution;
3. **Personalized experience.** By integrating sophisticated AI capabilities, including facial recognition and the capacity to learn and adapt to users' behaviours, while also leveraging data from other devices in the smart home ecosystem, Vimar has the potential to outperform its competitors by providing more proactive and tailored security solutions.

Vimar has the opportunity to gain a competitive advantage in the market by exploiting the gaps in current products and offering advanced solutions that meet the growing needs of consumers. In particular, the differentiation strategy for this project is based on addressing the primary weakness of competitors in the market: ecosystem integration. Full integration with the Vimar ecosystem represents a distinctive competitive advantage over competitors that often exhibit limited compatibility or require the use of multiple platforms. In the context of a smart home, the ability to manage and control all devices, including intercoms, thermostats, lighting, smart locks, through a centralised platform is essential in order to optimise efficiency and ease of use, thereby ensuring a smoother and more cohesive user experience. A unified management platform ensures that the smart peephole integrates seamlessly with the Vimar home automation ecosystem, which is already present in many homes and which users are already familiar with.

1.3.3 SWOT analysis

Following an investigation of the market context and the distinctive features of the smart peephole, it is beneficial to conduct a SWOT analysis (Fig. 1.7), which enables a comprehensive assessment of the potential benefits and risks associated with the launch of Vimar's smart peephole.

A SWOT analysis is a strategic planning tool that is employed to evaluate the strengths, weaknesses, opportunities and threats associated with a company, project or product. A strength is defined as a positive internal characteristic of product that represents a competitive advantage. Weaknesses are defined as negative internal characteristics of a product that could potentially represent a disadvantage. Opportunities are defined as positive external factors that product can exploit to its advantage. Threats are defined as negative external factors that could represent challenges or risks for a product.

Strengths

Ecosystem Integration: Strong integration with the Vimar ecosystem, allowing centralised and simple management of all smart devices.

Battery Life: Long battery life, reducing the need for frequent recharging and user intervention.

Low cost: Mid-range price positioning, which could provide access to a wider range of consumers.

Weaknesses

Hardware Bonding: For correct operation, the smart peephole needs Vimar intercom hardware.

Opportunities

Rapid Technological Evolution: Rapid technological advancement can make current functionalities obsolete, requiring continuous upgrades.

Barriers to Entry: The introduction of a new product type requires overcoming natural barriers to entry.

Threats

Growing Security Demand: The growing perception of insecurity is driving consumers to invest in advanced security solutions.

Home Automation: The spread of the Internet of Things (IoT) and smart homes offers significant growth opportunities.

Figure 1.7: SWOT Analysis

1.4 Technical constraints

In the context of a corporate project, two distinct realms can be identified: firstly, the constraints imposed by the organisation itself, and secondly, the "*spaces of freedom*" where creativity and innovation can flourish. The technical constraints pertain to the specific protocols, frameworks or systems that the project must align with, which are often dictated by the existing infrastructure and ecosystem of the company. These constraints function as boundaries, influencing the fundamental approach and capabilities of the project. In contrast, the "*spaces of freedom*" represent the areas where the project team can engage in independent research and make unfettered choices, without being constrained by restrictions. This freedom allows for the exploration of the latest technological advancements and industry best practices, without being constrained by the limitations of the company's current setup. The dynamic interplay between these constraints and freedoms forms a crucial aspect of the project design process, where the team must balance organisational requirements with innovative thinking in order to deliver a successful outcome. The following section provides a more detailed examination of these constraints and freedoms.

1.4.1 Constraints

Low Cost Platform. One of the principal technical limitations imposed on the project is the necessity for a low-cost hardware platform. This constraint is of great consequence for a number of reasons:

- **Cost-effectiveness.** By employing a low-cost hardware solution, the project can attain a more advantageous price point for the final product. This is fundamental to ensuring the product is accessible to the target market and that it remains competitively priced in comparison to alternative options;

- **Accessibility** A low-cost hardware platform enables the solution to be accessible to a more diverse customer base, including those with budgetary limitations. This facilitates the expansion of the project's reach and appeal to a more diverse user base, including those who may otherwise be excluded from premium offerings;
- **Resource optimisation.** The constraint of hardware costs allows the project to reallocate resources towards the development of advanced software features, enhancements to the user experience, and other value-adding aspects of the solution. This strategic optimisation facilitates the unlocking of greater overall value.

Communication protocol. The communication protocol represents a significant challenge for the design of the peephole, as it must facilitate the integration of the device with the company's existing ecosystem. This constraint is of fundamental importance for a number of reasons:

- **Interoperability.** The standardised communication protocol ensures that the peephole is able to establish a proper interface and exchange data with other systems and devices that are already present within the company's ecosystem. This interoperability is fundamental for the effective functioning of the entire system;
- **Security.** The use of a company-approved and supported communication protocol ensures that the data transmitted by the peephole is protected and integrated with existing security mechanisms. This mitigates vulnerability risks and ensures corporate compliance;
- **Centralised management.** The communication protocol allows the peephole to be integrated into the company's centralised management and monitoring infrastructure, facilitating remote control, updating and maintenance of the device within the ecosystem.

In regard to the "*space of freedom*", in addition to the constraint of the low-cost hardware platform, the selection of the most suitable hardware for the peephole will be at the design team's discretion. This will include the selection of microcontrollers and microprocessors, while maintaining the objective of low cost. Furthermore, cutting-edge solutions for real-time video streaming protocols and intelligent battery management will be investigated to guarantee optimal performance despite hardware limitations.

To conclude, this chapter presented the background to the problem and the market analysis for Vimar's development of a new smart peephole. The comparative analysis of the principal competitors revealed a number of common features and potential areas for improvement, which Vimar could exploit in order to gain a competitive advantage in the market. In particular, potential source of differentiation was identified, specifically in relation to integration with the Vimar ecosystem. The SWOT analysis revealed the strengths, weaknesses, opportunities and threats inherent to the introduction of the new Vimar smart peephole. Furthermore, the principal technical constraints that the project is required to meet, such as the utilisation of a cost-effective hardware platform and integration with the existing Vimar ecosystem communication protocol, were examined. These constraints present a challenge, but they also offer opportunities for exploring innovative solutions while remaining within the confines of the restrictions. The subsequent chapter provides a detailed analysis the various aspects identified within the "*space of freedom*", including a comparison of the available state-of-the-art solutions and an evaluation of their respective advantages and disadvantages.

Chapter 2

Technical challenges

2.1 Hardware

In order to achieve the objectives set out by Vimar for the development of a smart peephole, it is essential to make an informed decision regarding the selection of appropriate hardware components at the outset. The selection of suitable hardware will have a direct impact on the performance, efficiency and user experience of the final product. In particular, the choice of the microcontroller and the management of the battery is of great importance in order to ensure that the smart peephole meets the key requirements of conncectivity, real-time video communication, efficient power consumption and the need for a low-cost platform.

2.1.1 Microcontroller vs microprocessors

Definition. Microcontrollers and microprocessors are indispensable components in the field of computing. Each serves a distinct purpose and is employed in a variety of applications. A microcontroller, as defined by [58], is a comprehensive microprocessor system encapsulated within a single chip. It differs from multi-purpose microprocessors commonly found in personal computers, in that it already incorporates essential components like memory and programming support. In contrast, a microprocessor, described as a digital device on a chip that executes instructions from memory, is more versatile and used in a wide array of computing devices [82].

Applications. Microcontrollers, being self-contained systems, are commonly employed in real-time applications within embedded systems where precise program execution timing is crucial, as highlighted by [65]. These devices are typically utilized in applications that require direct control of hardware (Figure 2.1), such as in home automation systems [59], wearable devices [32], and industrial controls [53].

In contrast, microprocessors are more general-purpose and are utilised in a broader spectrum of devices, thanks to their flexibility and computational power [63]. They are typically found in personal computers, servers, and mobile devices where complex computations and multitasking are required.

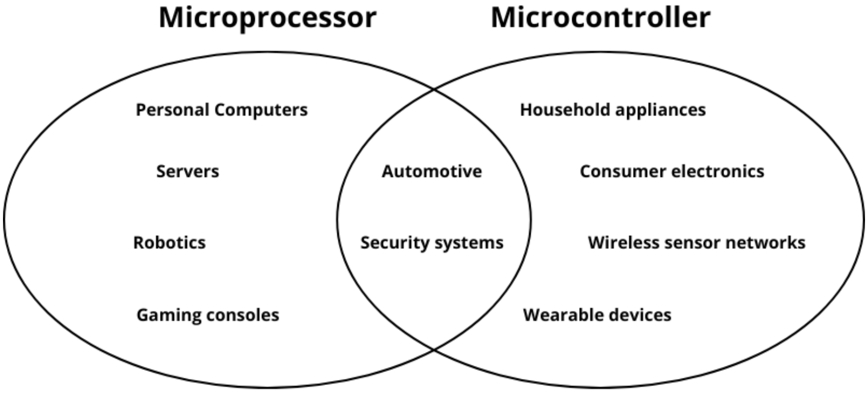


Figure 2.1: Microncontrollers and microprocessors applications.

Power efficiency. Power efficiency is a critical aspect of both microcontrollers and microprocessors, especially in battery-powered and portable devices. Microcontrollers are generally more energy-efficient due to their specialized design and ability to enter low-power modes when not performing active tasks [69] [83]. Microprocessors have incorporated techniques such as dynamic voltage scaling (DVS) to manage power consumption effectively, but achieving power efficiency while meeting performance requirements remains a challenge [121]. Microcontrollers are optimized for control tasks within embedded systems, offering low power consumption and cost-effectiveness, while microprocessors provide higher performance and flexibility for general-purpose computing, consuming more power.

In consideration of the specific requirements at hand, it becomes evident that a microcontroller represents the optimal choice. Microcontrollers are particularly well-suited for real-time embedded systems due to their integrated design and precise timing capabilities. Although microprocessors offer greater versatility and enhanced processing power, rendering them more applicable to a wider range of computational tasks, microcontrollers are uniquely positioned to strike an equilibrium between cost, power efficiency and processing capacity. This makes them particularly well-suited for applications that demand precise control, efficient power usage, real-time operations and a compact form factor. Consequently, for a system such as our smart peephole, a microcontroller emerges as the most optimal choice.

2.1.2 Microcontrollers

The history of microcontrollers dates back to the 1970s when they were first introduced, revolutionizing the field of embedded systems by providing a compact and cost-effective solution for processing tasks in real-time [111] . Over the years, microcontrollers have significantly evolved, becoming more powerful, energy-efficient, and capable of handling complex operations.

One of the key reasons for the widespread use of microcontrollers is their ability to facilitate data acquisition and processing in a compact and efficient manner [85]. Equipped with various peripherals, these devices can interact with sensors, actuators, and other external components, making them ideal for applications that require real-time monitoring and control (Figure 2.2). Additionally, microcontrollers offer the advantage of being programmable, allowing developers to customize their functionality based on specific requirements. This flexibility makes them suitable for a diverse range of applications, from simple automation tasks to sophisticated control systems.

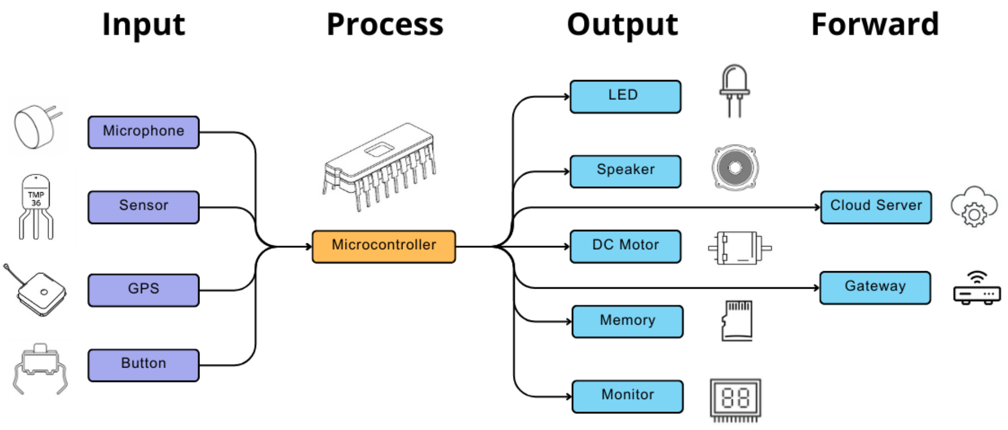


Figure 2.2: Microncontrollers processing cycle.

Despite their advantages, microcontrollers do have limitations. One common challenge is their limited processing power compared to larger computing systems such as PCs or servers [98]. This constraint can restrict the complexity of tasks that can be performed directly on the microcontroller, necessitating offloading certain computations to external devices or cloud services. The memory and storage capacity of microcontrollers are typically constrained, which can impact the amount of data that can be processed or stored locally. However, advancements in microcontroller technology have led to the development of more powerful variants that address some of these limitations.

As detailed by [52], the components of a microcontroller include several key elements that collectively enable its functionality. At the heart of the system is the Central Processing Unit (CPU), which is responsible for executing instructions and performing calculations. The memory units, comprising Random Access Memory (RAM) and Read-Only Memory (ROM), serve to store the data required by the processor during operation. The ROM stores permanent data, including command sets and programs, while the RAM is used for temporary data storage. Variants of the ROM include PROM, EPROM, and EEPROM. Input/output interfaces allow the microcontroller to receive data from external units and transmit data, thereby facilitating communication with other devices. The Analog-to-Digital Converter (ADC) is responsible for converting analog signals into digital data, which can then be processed. Pulse Width Modulation

(PWM), on the other hand, is used to control the power delivered to electrical devices. The Arithmetic Logic Unit (ALU) performs essential arithmetic and logical operations ans special registers serve to store intermediate data and control information. Moreover, additional components or peripherals may be incorporated, contingent on the specific microcontroller model and its intended applications. Such features may include timers, counters, serial communication interfaces (such as UART, SPI, and I2C), digital-to-analog converters (DAC), and network interfaces (such as Ethernet or CAN). Furthermore, an antenna for Wi-Fi or Bluetooth connectivity may be present, facilitating wireless communication and expanding the microcontroller's range of applications.

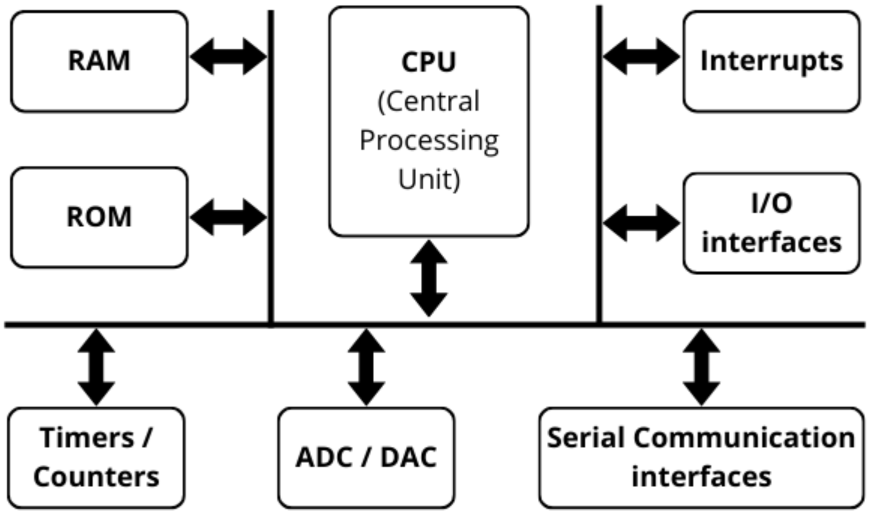


Figure 2.3: Typical microncontrollers components.

As previously stated, there is a considerable range of microcontrollers on the market that differ in terms of the components and peripherals they incorporate. It is therefore essential to select an appropriate microcontroller platform in order to ensure the success of a project and to achieve its desired outcomes. For instance, the ESP32 is frequently selected due to its integrated Wi-Fi and Bluetooth functionality, making it an optimal choice for Internet of Things (IoT) applications. The Raspberry Pi Pico, renowned for its cost-effectiveness and robust performance in simple interfacing, may be the preferable choice for projects where there is a need to balance battery life with computing power. The STM32 family, celebrated for its high performance and extensive peripheral set, may be the optimal selection for more demanding applications. Selecting the appropriate microcontroller platform ensures that the hardware capabilities align with the project requirements, thereby facilitating the successful implementation and operation of the intended application.

2.1.3 Platform comparison

The smart peephole must be capable of efficiently managing real-time video streaming, handling fast and secure communications, maintaining a long battery life, and ensuring seamless integration with the Vimar ecosystem. Each microcontroller platform possesses distinctive features, strengths, and limitations that must be meticulously evaluated against the specific project requirements.

There are several critical features and parameters to consider when selecting a microcontroller for a project as demanding as a smart peephole. These factors ensure that the chosen platform can meet the technical and functional requirements while maintaining efficiency and cost effectiveness. The main characteristics and parameters to be taken into account are listed below:

- **Computational power.** This is particularly important when handling real-time audio and video streaming, which requires robust processing capabilities. A microcontroller with sufficient on-chip RAM is essential to handle the data-intensive operations associated with high-definition video streaming and audio processing. In addition, the microcontroller's clock speed plays a critical role in how quickly it can process instructions. Higher clock speeds enable faster data processing and more efficient handling of complex tasks, reducing latency and ensuring a smooth user experience [113].

- **Connectivity.** For a smart peephole, the microcontroller must support both Wi-Fi and Bluetooth to facilitate communication with other smart home devices and the user's smartphone. Wi-Fi connectivity is essential for streaming video to remote devices and integrating with home networks, while Bluetooth can be used for local device pairing and low-power communication. These connectivity options increase the versatility and interoperability of the smart peephole, making it more adaptable to different user needs and home environments;

- **Cost.** The financial aspect is of primary importance in any project, particularly in the case of consumer electronics such as smart peepholes, where market competitiveness hinges on affordability. The selection of a low-cost microcontroller that does not compromise on essential features, ensures that the final product can be offered at a competitive price point. It can be seen that achieving an equilibrium between cost and functionality is of great consequence with regard to the marketability and potential for a product to gain a substantial user base.

- **Power consumption.** The microcontroller must be constructed with energy-efficient characteristics and power management abilities, including sleep modes, in order to extend battery life and reduce the frequency of recharging. Effective power usage has a direct impact upon the longevity of the device and the convenience of the user. Implementing effective sleep modes is especially important in minimising energy consumption during periods of inactivity, as this is critical in maintaining a balance between functionality and battery efficiency in a device which must be operational for extended periods of time without maintenance.

- **Ease of development.** The concept of development simplicity, or user-friendliness, is an important parameter in this context. The ease of programming and integration of a microcontroller can lead to a notable reduction in development time and effort. Factors that contribute to this include the provision of a well-documented software development kit (SDK), the availability of comprehensive libraries and support for common programming languages. Additionally the availability of an open-source SDK is particularly beneficial. It facilitates easier integration with the company's ecosystem by providing access to source code, libraries, and tools that can be customized and extended. This transparency promotes a more direct integration process and allows for greater flexibility in adapting the microcontroller to specific project requirements [108]. It also enables the development community to contribute improvements and support, further enhancing the development experience and long-term maintenance of products.

- **Community support.** The presence of a large and active developer community is a significant advantage, as it provides access to a wealth of valuable resources, including forums, tutorials, shared code, and open-source libraries. Such open-source resources not only facilitate accelerated development but also encourage innovation and problem-solving through the leveraging of community contributions.

- **Peripheral support.** The microcontroller should have sufficient peripherals and expansion capabilities to support additional functionality. This includes support for sensors, ADCs, DACs and communication interfaces such as UART, SPI and I2C. These features allow the integration of various sensors and modules that can enhance the capabilities of the smart peephole.

Having established the essential characteristics to be considered in the selection of a microcontroller, we may now proceed to evaluate the specific microcontroller options that meet these criteria. The following sections provides a detailed comparison of several prominent platform.

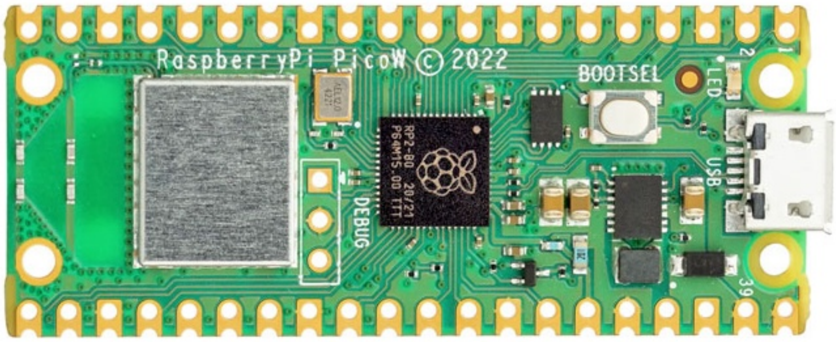


Figure 2.4: Raspberry Pi Pico W Board.

Raspberry Pi Pico (W). It is a microcontroller board that has gained attention for its versatility and capabilities. This microcontroller has been utilized in various applications, it has been integrated into different projects such as security systems, hydroponic plant protection systems, solar tree systems, and educational applications, showcasing its adaptability across various domains [92]. The energy-efficient performance of the Raspberry Pi Pico renders it an appropriate selection for applications that require low power consumption [104]. It has also been used in IoT applications, and surveillance systems [104]. The W variant of the Pi Pico (Fig. 2.4) was enhanced with the introduction of wireless connectivity (2.4GHz LAN b/g/n) through the incorporation of the wireless interface provided by Infineon CYW43439. The Raspberry Pi Pico and Pico W are supported by a robust development environment, with extensive documentation, libraries, and support for programming in languages like MicroPython and C/C++. This makes them accessible to both beginners and experienced developers, enabling a wide range of applications from simple DIY projects to more complex embedded systems.

Figure 2.5: STM32WB module.

STMicroelectronics STM32. The popularity of the STM32 microcontroller family has grown significantly in recent years, and it is now widely used in a variety of applications due to its high performance, low power consumption, and cost-effectiveness [106]. These microcontrollers are based on ARM Cortex-M cores, offering an excellent balance between power efficiency and processing power. This makes them suitable for a diverse array of applications, from simple embedded systems to complex industrial automation. The STM32 family comprises a variety of core variants, including the Cortex-M0 and M0+, which are ideal for low-power applications, such as sensor management and battery-operated IoT devices; the Cortex-M3, which offers a balance of performance and efficiency, making it a popular choice for general-purpose embedded systems; and the Cortex-M4 variant includes a floating-point unit, which enhances its capability for digital signal processing tasks, making it suitable for audio processing, motor control, and other real-time applications. The STM32W series (Fig. 2.5) is a variant of the STM32 microcontrollers that has been designed to integrate wireless connectivity, including Bluetooth, Zigbee, and Thread. It features a dual-core architecture, with an ARM Cortex-M4 responsible for primary tasks and a Cortex-M0+ dedicated to managing wireless protocols.

Figure 2.6: ESP32 WROVER-E module.

Espressif ESP32. The ESP32 is a microcontroller developed by Espressif Systems that offers a high level of processing power. It is a widely utilised microcontroller in a multitude of IoT (Internet of Things) applications, primarily due to its versatility, low power consumption and wireless connectivity. The ESP32 is based on the Tensilica Xtensa LX6 microprocessor and features dual-core 32-bit processors operating at speeds of up to 240 MHz. One of the ESP32's most notable attributes is its integrated Wi-Fi and Bluetooth connectivity, encompassing 802.11 b/g/n Wi-Fi and Bluetooth Low Energy (BLE). This facilitates communication with other devices and networks, making it an optimal selection for projects that necessitate wireless connectivity, including but not limited to home automation systems, smart devices, and sensor networks. In addition to its wireless capabilities, the ESP32 offers a wide range of peripheral interfaces, including SPI, I2C, UART, and ADC, which provide the means for interfacing with a variety of sensors, displays, and other external devices. Programming the ESP32 is possible through the Arduino Integrated Development Environment (IDE), MicroPython, or the open-source ESP-IDF (Espressif IoT Development Framework), which allows for low-level access to the hardware features of the chip. This grants developers the ability to utilise the ESP32's capabilities fully and create bespoke firmware tailored to their specific requirements. The ESP32 also incorporates sophisticated security features, including hardware encryption, secure boot, and flash encryption, which collectively guarantee the protection of data transmission and storage against unauthorized access.

Figure 2.7: Nordic nRF5340 SoC (no antenna).

Nordic nRF5340. The Nordic nRF5340 is a dual-core system-on-chip (SoC) designed for wireless applications, particularly in the Internet of Things (IoT) domain. The device features two ARM Cortex-M33 processors, one of which is optimised for high-performance tasks and the other for energy-efficient operation. This provides a flexible balance between power and performance. The primary core, operating at a maximum frequency of 128 MHz, is intended for the processing of applications and is capable of handling complex algorithms, data processing, and user interface tasks. The secondary core, which is optimised for low-power operations, is typically employed for the management of network protocols such as Bluetooth Low Energy (BLE), Thread, and Zigbee. The dual-core architecture of the nRF5340 enables the device to perform complex tasks while maintaining low power consumption, which makes it an ideal choice for battery-operated devices. In addition to its processing capabilities, the nRF5340

Board	**ESP32**	**Raspberry Pi Pico W (Board)**	**STM32WB**	**nRF5340**
Processor	Tensilica LX6	RP2040 (ARM Cortex-M0+)	ARM Cortex-M4 / ARM Cortex-M0+	ARM Cortex-M33
Clock Speed	240 MHz	133 MHz	64 / 32 MHz	128 / 64 MHz
Flash Memory	4 MB	2 MB	Up to 1 MB	1 MB
RAM	520 KB	264 KB	256 KB	512 KB
Communication Interfaces	I2C, SPI, UART, Wi-Fi, Bluetooth	I2C, SPI, UART, WiFi, Bluetooth	SPI, I2C, UART, Bluetooth, IEEE 802.15.4	SPI, I2C, UART, Bluetooth, IEEE 802.15.4
Price	2 - 4€	5 - 6€	5 - 7€	6 -7€

Table 2.1: Microcontroller platform comparison.

is compatible with a range of wireless protocols, including BLE 5.2, which offers features such as long-range communication, high throughput, and direction finding. Furthermore, the SoC incorporates an advanced security suite comprising Arm TrustZone, which facilitates trusted execution through the implementation of a division between secure and non-secure Flash, RAM, peripherals and GPIOs. Additionally, CryptoCell-312 provides hardware-accelerated cryptography, and in conjunction with the key management unit (KMU) peripheral, root-of-trust and secure key storage are implemented.

In order to provide a clear overview of the options considered for the smart peephole project, a comparison of the technical features of the microcontrollers evaluated is presented in the Table 2.1. This comparison serves to highlight the relative strengths and limitations of each option, thereby facilitating the selection of the most suitable microcontroller for the project. In the process of selecting the optimal microcontroller for the smart peephole project, the ESP32 emerges as the most promising candidate, offering a number of advantages. Firstly, in contrast to the STM32 and Nordic nRF5340, the ESP32 integrates both Wi-Fi and Bluetooth capabilities, which are essential for the project. Although Bluetooth is a useful technology for lightweight communications, it is not a sufficient solution for the full range of requirements of our smart peephole project. In particular, its low data transfer rate makes it unsuitable for real-time video streaming, which requires a stable, high-bandwidth connection. While the STM32WB and the Nordic nRF5340 demonstrate considerable processing capabilities and low power consumption, they lack the integration of Wi-Fi, necessitating the incorporation of supplementary components, which would consequently increase the complexity and cost. When comparing the ESP32 and the Raspberry Pi Pico W for a project requiring real-time video transmission, it is evident that the computational capabilities of each play a significant role. The ESP32, equipped with dual-core 32-bit Tensilica Xtensa LX6 processors operating at up to 240 MHz, offers considerably greater processing power than the Raspberry Pi Pico W, which features a dual-core ARM Cortex-M0+ processor operating at a lower clock speed of 133 MHz. The transmission of video in real time is a computationally demanding task, necessitating the encoding, compression, and streaming of video data in an efficient manner. The higher clock speed of the ESP32 facilitate more effective handling of these tasks, ensuring smoother video streaming and faster data processing, whereas the Pico

W may encounter difficulties in meeting the processing demands of continuous video transmission due to its lower computational power.

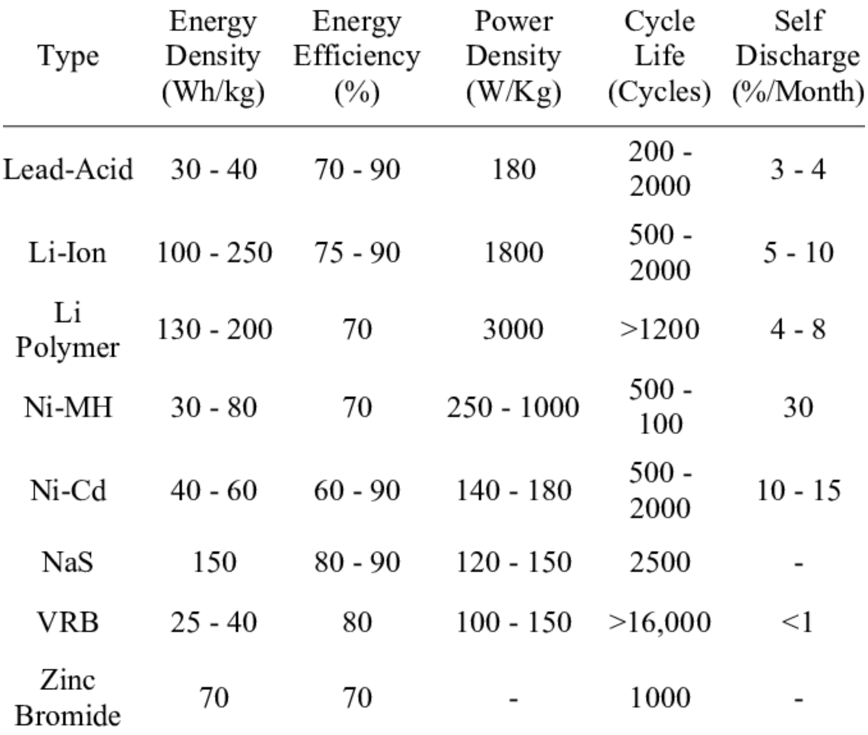
2.2 Battery

When selecting a battery type for a particular application, it is essential to consider a number of key characteristics in order to guarantee that the battery is capable of meeting the performance requirements and operational needs of the device in question. The following represents a subset of the factors that should be considered:

- **Energy density.** The term "energy density" is used to describe the amount of energy that can be stored in a battery, relative to its size or weight. This is typically expressed in watt-hours per kilogram (Wh/kg). A higher energy density indicates that the battery can store more energy in a smaller and lighter package, which is essential for portable devices or applications where space and weight are limited [48];
- **Cycle life.** The term 'cycle life' is used to describe the number of complete charge and discharge cycles that a battery can undergo before a significant reduction in capacity occurs[120]. In the context of frequently used devices that require long-term reliability, a longer cycle life became essential;
- **Self discharge rate.** The term 'self-discharge rate' refers to the rate at which a battery loses its charge when not in use. This phenomenon is caused by internal chemical reactions within the battery that do not contribute to external current flow. The phenomenon of self-discharge is an inherent characteristic of batteries. It results in a reduction in the stored charge over time, even when the battery is disconnected from a load or device [96];
- **Charging capabilities.** The term "charging capabilities" in batteries is used to describe the ability of a battery to be charged at an accelerated rate, which significantly reduces the time required for a full recharge. This feature is of particular importance in a number of applications, including electric vehicles and portable electronic devices, where the ability to recharge rapidly is essential. [72];
- **Energy efficiency.** The term "energy efficiency" is used to describe the ratio of the useful energy output of a system to the energy input, which is typically expressed as a percentage. It indicates the proportion of the stored energy that can be recovered during discharge, in comparison to the energy originally stored during charging. It is typically expressed as a percentage (%). High energy efficiency implies a reduction in energy loss during storage and conversion, it influences the lifespan of a battery before requiring a recharge and the amount of energy dissipated as heat;
- **Power density.** The term "power density" is used to describe the rate at which energy can be delivered by the battery in relation to its mass or

> volume. The power density of a battery is typically expressed in units of watts per kilogram (W/kg) or watts per litre (W/L). This measurement indicates the battery's ability to provide a high power output in a short duration. [79].

Lithium-ion batteries have become the preferred option for a diverse range of smart home applications, largely due to their exceptional performance characteristics (Figure 2.8). These batteries are distinguished by their high energy density, extended cycle life, low self-discharge rate, and rapid charging capabilities [93] [64], rendering them particularly well-suited for integration into portable devices such as smartphones, laptops, and tablets, as well as smart sensors and actuators that are integral to the Internet of Things (IoT) in modern smart homes [33].



Type	Energy Density (Wh/kg)	Energy Efficiency (%)	Power Density (W/Kg)	Cycle Life (Cycles)	Self Discharge (%/Month)
Lead-Acid	30 - 40	70 - 90	180	200 - 2000	3 - 4
Li-Ion	100 - 250	75 - 90	1800	500 - 2000	5 - 10
Li Polymer	130 - 200	70	3000	>1200	4 - 8
Ni-MH	30 - 80	70	250 - 1000	500 - 100	30
Ni-Cd	40 - 60	60 - 90	140 - 180	500 - 2000	10 - 15
NaS	150	80 - 90	120 - 150	2500	-
VRB	25 - 40	80	100 - 150	>16,000	<1
Zinc Bromide	70	70	-	1000	-

Figure 2.8: Comparison of different battery types.

Source: [20]

The high energy density of lithium-ion batteries is attributed to their capacity to store a considerable amount of energy in relation to their dimensions and mass [64]. This attribute is crucial for optimising energy storage capacity in a confined space, which is a fundamental requirement for a device such as the smart peephole. The operational lifespan of such a device must be considerable in relation to its overall dimensions, which must conform to a compact form factor.

Another significant advantage of lithium-ion batteries is their greater cycle life, which allows them to withstand a greater number of charge-discharge cycles before significant capacity degradation occurs. This is particularly important given the high level of use expected for the device. A longer cycle life ensures that the smart peephole remains reliable over time, reducing the need for frequent battery recharging and consequently, enhancing the overall user experience.

Furthermore, lithium-ion batteries are aligned with the increasing global emphasis on sustainability and the reduction of greenhouse gas emissions. Their efficiency and capacity for energy storage contribute to the development of a cleaner energy ecosystem, thereby facilitating the transition towards more sustainable power sources [118].
The high energy and power densities, combined with extended cycle life, quick charging capabilities, high energy efficiency, low self-discharge rate and environmental sustainability, make lithium-ion batteries the optimal choice for our smart peephole project, ensuring both performance and longevity in a compact and efficient power solution.

Once the battery type has been selected, monitoring the state of charge (SoC) becomes necessary in order to guarantee optimal performance and extend the battery's lifespan. The following section examines the importance of SoC and the various techniques for accurately estimating it.

2.2.1 State of charge

The state of charge (SoC) of a battery is a key parameter to monitor when working with battery-powered devices, as it provides insight into the available capacity of the battery at a given point in time. [17] (Figure 2.9).

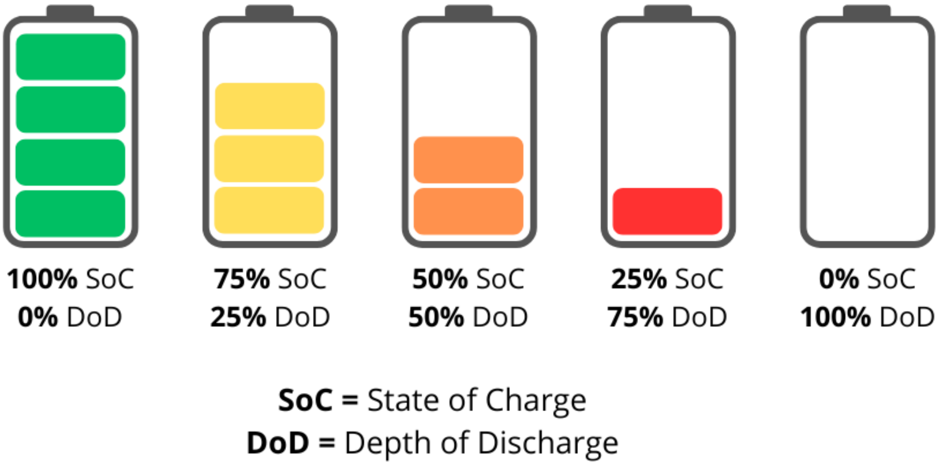


Figure 2.9: Battery State of Charge and Depth of Discharge relation.

It is defined as the ratio of the available capacity to the maximum capacity when the battery is fully charged [84], typically expressed as a percentage. This can be represented by the following formula [117]:

$$Soc = \frac{actually\ available\ amount\ of\ charge}{maximum\ available\ amount\ of\ charge} \times 100 \tag{2.1}$$

The monitoring of the state of charge (SoC) enables users and systems to make well-informed decisions regarding power usage, charging, and device operation. A precise knowledge of this parameter in battery-powered devices confers a number of advantages. Firstly, it enables the implementation of effective energy management strategies, thereby ensuring that devices operate at optimal efficiency without the occurrence of unexpected shutdowns due to depleted batteries [12]. This is of particular importance in critical applications, such as medical devices or emergency equipment, where an uninterrupted power supply is essential [97]. Furthermore, an awareness of the SoC enables the avoidance of overcharging or over-discharging, which can have a considerable impact on the lifespan and performance of batteries [54].

Moreover, an accurate assessment of the state of charge can be employed for safety-related purposes. Monitoring the SoC helps in preventing potential hazards such as overheating, which can lead to thermal runaway and even fires in lithium-ion batteries [66]. Lastly, predicting the remaining runtime of battery-powered devices, allowing users to plan their activities accordingly and avoid unexpected interruptions [77].

In the context of our smart peephole project, it is equally important to be able to determine the state of charge. The smart peephole is a battery-powered device that requires a consistent and reliable power supply in order to function optimally. The monitoring of the state of charge (SoC) is essential for ensuring that the device remains operational, thereby enabling the provision of real-time video streaming and audio capabilities without interruption. In the event of a sudden power loss, there is a risk of missed or delayed notifications of visitors, which could potentially compromise the safety and convenience that the smart peephole is designed to offer.

2.2.2 Approaches to SoC estimation

The estimation of the state of charge (SoC) in lithium-ion batteries is a challenging process, due to the presence of several interrelated factors.

Firstly, the non-linear relationship between voltage and state of charge (SoC), which can result in inaccuracies in estimation methods that rely solely on voltage measurements, as minor fluctuations in voltage may correspond to considerable shifts in charge levels [80]. Additionally, temperature variations can significantly affect battery performance and SoC readings, complicating the estimation process further [30]. Another challenge is the aging of batteries, which alters their capacity and internal resistance, making it difficult to apply standard estimation algorithms consistently [73]. These factors collectively impede the accurate estimation of the state of charge (SoC), thereby necessitating the continued development of more robust methodologies.

Several techniques have been developed and studied over the years. The most prevalent techniques include:

- **Direct measurement approaches:**

  - **Open-circuit voltage (OCV):** this method relies on the stable electromotive force of the battery in an open-circuit state to infer the SOC value. Although the OCV method is an effective means of estimating SOC, it has certain limitations. For instance, it requires the battery to be in a static state for over an hour, which makes it unsuitable for real-time applications due to its sensitivity to external factors [130]. Despite these drawbacks, the OCV method continues to be a prevalent approach in battery modelling, state estimation, and management, largely due to its simplicity and efficacy [29];
  - **Coulomb counting (CC)** or Ampere-Hour method, involves integrating the current over time to determine the amount of charge that has been consumed or replenished by the battery [130]. Although the method is relatively simple, it is susceptible to the accumulation of errors over time and requires an accurate initial SoC measurement;

- **Data-driven approaches.** These methods have gained significant traction due to advancements in machine learning and deep learning technologies [131]. These approaches leverage techniques such as Gaussian Process Regression (GPR) [126], convolutional neural networks (CNN) [56], recurrent neural networks (RNN) [116], and support vector machines (SVM) [129] to estimate SOC accurately. These methods can adapt to different battery types but require a large amount of training data;

- **Model-based approaches.** These methods utilize mathematical optimization techniques and advanced algorithms to estimate the SOC accurately. One common approach is to employ Kalman filters, such as the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF), to model the battery behavior and estimate the SOC [124] [74]. These filters aid in tracking the battery's state over time by integrating dynamic battery models and measurement data;

- **Hybrid methods.** These methods integrate distinct techniques (such as Coulomb counting with Kalman filtering or data-driven methods with model-based approaches) to capitalise on the strengths of multiple methodologies, thereby enhancing the accuracy and robustness of SoC estimation. Hybrid methods seek to strike a balance between computational complexity, accuracy, and adaptability.

SoC estimation methods can be classified according to their calculation complexity, which allows for the categorisation of these methods as either simple or complex algorithms. The simplest of these methods are open-circuit voltage (OCV), ampere-hour (Ah), and internal resistance methods, which estimate the state of charge (SOC) by measuring voltage, current, or internal resistance [125].

In the development of the smart peephole, it is essential to select a SoC estimation methods that offer a balance between accuracy and minimal computational overhead, given that the device operates with limited computing power. While advanced algorithms like the Extended Kalman Filter (EKF) and machine learning models are effective for SoC estimation, they may be computationally demanding [14].

A potential solution that balances accuracy with computational simplicity is the one employed by [125].

A novel practical state of charge estimation method: an adaptive improved ampere-hour method based on composite correction factor.

The research addresses the challenge of accurately estimating the state of charge (SOC) of lithium-ion batteries (LIBs). The current methodologies for SOC estimation, including the ampere-hour (Ah) method, are constrained by external factors such as temperature, battery ageing and measurement noise, which impinge upon their accuracy. While sophisticated model-based and data-driven algorithms offer enhancements, they markedly increase computational complexity, rendering them less appropriate for real-time applications. This study seeks to refine the traditional Ah method by introducing an adaptive enhanced method that accounts for the primary disturbance factors affecting battery performance.

Improvement Strategy of Ampere-hour Method. The traditional ampere-hour (Ah) method is founded upon a straightforward calculation of charge inflow and outflow, wherein constant values are assumed for parameters such as Coulomb efficiency, charge-discharge efficiency, and the battery's total capacity. However, these parameters are not static and can vary significantly due to a number of factors, including temperature, current rate, and the state of health (SOH) of the battery. For instance, lower temperatures or higher discharge rates can reduce the battery's effective capacity, which may lead to inaccuracies in SOC estimation if these factors are not accounted for.

The paper puts forth a strategy for improvement with the aim of addressing the aforementioned inaccuracies. The strategy comprises the following elements:

- **Calibration of Initial SOC:** the initial SOC value is pivotal for precise estimation. Consequently, the enhanced methodology proposes a more accurate recalibration of this initial value through the utilisation of open-circuit voltage (OCV). The relationship between open-circuit voltage (OCV) and state of charge (SOC) is typically non-linear but stable, thereby enabling a more precise determination of SOC at the outset of the monitoring period;

- **Dynamic Adjustment of Parameters:** the improved method suggests modifying key parameters, such as the battery's available capacity and Coulomb efficiency, in accordance with real-time conditions, including temperature fluctuations and the battery's present operational state. This approach helps to mitigate errors introduced by the assumption that these parameters are constant;

- **Consideration of Measurement Errors:** the enhanced ampere-hour method reduces SOC estimation errors through the implementation of techniques designed to address the presence of noise and sampling inaccuracies. The method incorporates filtering to mitigate noise, employs dynamic parameter adjustment to compensate for changing conditions, and utilises a composite correction factor (Figure 2.10) to address any drift in SOC calculations.

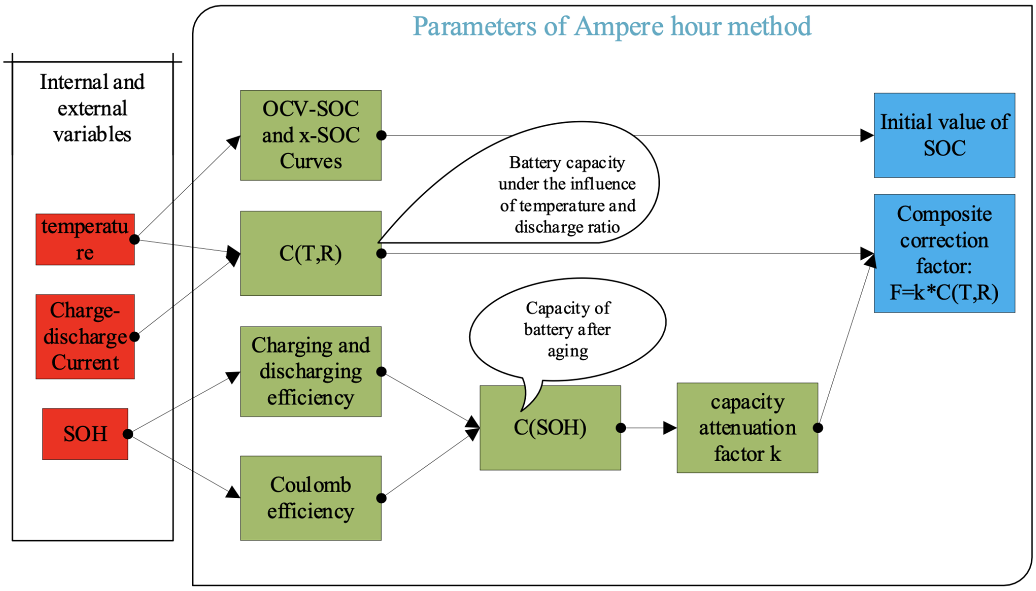


Figure 2.10: Composite correction factor.

Source [125].

The research presents significant findings regarding the effectiveness of the proposed improved ampere-hour method. The adaptive improved method achieved an estimation error of less than 2% under two comprehensive working conditions. This represents a substantial improvement compared to the traditional ampere-hour method, which exhibited an error range of 5-10%. A comparison of the extended Kalman filter algorithm with the adaptive improved method revealed that the latter exhibited superior performance in state of charge estimation. This suggests that the new method not only enhances accuracy but also outperforms established algorithms in practical applications.

2.3 Real-Time video streaming protocol

A Real-Time video streaming protocol is a set of rules and standards that govern the transmission of video data over the internet, enabling the delivery of live video content with minimal latency. In order to achieve this, a variety of protocols and technologies are employed to guarantee the efficient and seamless delivery of video content.

Real-time video streaming protocols are characterised by a number of key features, including:

- **Latency.** It is used to describe the delay between the capture of a video frame and its subsequent display to the viewer. It is a critical factor in ensuring a seamless viewing experience, whereby the time gap between the live event and its display on the viewer's screen is minimised. A number of studies have demonstrated that latency in video streaming is influenced by a range of factors, including limitations in network bandwidth, the processes of video encoding and decoding, and the transmission protocols employed [127] [62];

- **Bandwidth efficiency.** It is defined as the optimal utilisation of available bandwidth to deliver video content with high quality and minimal resource consumption. It comprises techniques and protocols that seek to optimise the quality of video streams while minimising the requisite bandwidth [90]. A variety of approaches have been developed to enhance bandwidth efficiency in video streaming services, including scalable video coding and adaptive rate-control schemes [35] [28];
- **Scalability.** Scalability in real-time streaming video protocols refers to the ability of a system to efficiently manage increasing numbers of users or data streams without compromising performance. This concept is critical as it assures the uninterrupted and responsive delivery of video content despite fluctuations in demand. It has been demonstrated that scalability can be attained through a number of different mechanisms. One such mechanism is adaptive bitrate streaming, which adjusts the quality of the video in accordance with the prevailing network conditions and the capabilities of the user, thereby optimising the utilisation of available resources.
- **Compatibility and integration.** The protocol needs to be compatible with the hardware and software platforms utilised in the Smart Peephole. This encompasses the microcontroller, operating system and any additional components involved in video processing and transmission. Furthermore, the protocol should facilitate straightforward integration with existing smart home systems and applications, thereby enhancing the overall user experience through the provision of seamless interaction with other connected devices.

The selection of an appropriate Real-Time Video Streaming Protocol directly impacts the functionality and user experience of the device in question. The Smart Peephole, designed with the objective of enhancing home security by allowing users to remotely monitor the area outside their door, relies heavily on the reliable transmission of video data.

2.3.1 Protocols comparison

Several protocols have been developed to cater to different needs and scenarios. Among the prominent protocols are Real-Time Streaming Protocol (RTSP), Web Real-Time Communication (WebRTC), Dynamic Adaptive Streaming over HTTP (DASH), and Secure Reliable Transport (SRT). Each protocol has unique features and parameters that make it suitable for specific applications.

Web Real-Time Communication. WebRTC is a suite of communication protocols that enables real-time communication over peer-to-peer connections. This type of communication minimizes latency and favours scalability, by allowing direct connections between users. It facilitates a range of functionalities, including video streaming, web conferencing, chat, and data exchange, through web applications that utilise JavaScript and HTML5 technologies [15]. The open-source protocol is implemented on the User Datagram Protocol (UDP), thereby enabling low-latency video streaming [61]. The WebRTC protocol is also recognised for its capability to adapt to changing network conditions, specifically in terms of modifying data rates to enhance performance. In their study, [46] highlighted that WebRTC estimates available bandwidth and conveys this information to the encoder to establish a target bitrate. To guarantee the security of WebRTC connections, the communication needs to be encrypted. One of the principal methods of achieving this is through the utilisation of the Secure Real-time Transport Protocol (SRTP). The deployment of SRTP facilitates end-to-end encryption, thereby protecting against eavesdropping and data tampering [19];

Real-Time streaming protocol. The Real-Time Streaming Protocol (RTSP) is an application-level protocol designed for controlling the delivery of real-time data, particularly for streaming media content [112]. RTSP allows for the establishment and management of media streaming sessions, facilitating the control of data transmission with real-time constraints over a network [11]. RTSP operates in conjunction with the Real-Time Protocol (RTP) for streaming the actual media content, with negotiation and control handled by RTSP before streaming commences [88]. RTSP is widely used in various applications that require real-time image streaming. For example, it is ha been employed in automated port entry and exit management systems [1];

Dynamic adaptive streaming over HTTP. DASH is a protocol that enhances multimedia streaming by allowing adaptive quality adjustments based on available bandwidth and device capabilities [5]. This technology segments videos into smaller parts called chunks or segments, which are then delivered to users via HTTP requests and responses over TCP connections [55]. DASH is widely used for providing high-quality video streaming over the Internet, offering adaptive and dynamic multimedia streaming solutions to various end systems [7]. Ensuring the security of DASH involves implementing security mechanisms within the DASH agent framework [87] to protect communication and data exchange processes, particularly for sensitive or proprietary content;

Secure reliable transport. This protocol is a significant element in ensuring the confidentiality, integrity, and reliability of data transmission over networks. It combines aspects of security protocols like Transport Layer Security (TLS) and reliability mechanisms to establish a robust communication framework [16]. SRT enhances this by integrating mechanisms to improve the reliability of data transmission, making it suitable for applications requiring both security and dependability. In the realm of wireless sensor networks (WSNs), where vulnerabilities to attacks can jeopardize data integrity and reliability, the development of secure transport protocols like SRT is crucial [25].

A comparison of real-time video streaming protocols reveals a diverse landscape, with each protocol catering to a specific set of requirements. Protocols such as RTSP and WebRTC are designed for use in interactive applications, while DASH is used for adaptive streaming and SRT for security transmission. Each protocol offers a distinct set of features and parameters, providing flexibility in meeting the varying needs of different applications.

WebRTC provides a versatile and efficient solution for real-time communication, enabling a wide range of applications across various domains. The aforementioned features, namely peer-to-peer communication, scalability and security, render it a preferred option for modern communication systems in comparison to traditional protocols such as RTSP. Nevertheless, RTSP (Real-Time Streaming Protocol) was selected over WebRTC and other alternatives for a number of significant reasons. The decision was influenced by the fact that the ESP32 microcontroller, which was selected as the hardware platform, already provides an RTSP library within its software development kit (SDK). This resulted in a markedly more straightforward implementation process and a reduction in development time. By leveraging this existing library, the integration of RTSP into the system was streamlined, ensuring a smooth and efficient setup. Another factor was that Vimar had already been utilising RTSP for other applications, thus establishing an infrastructure and experience with the protocol. Although not a mandatory requirement, this alignment facilitated integration with our current systems and practices, reducing potential compatibility issues and enhancing maintenance. Ultimately, RTSP was the optimal choice, balancing quality, ease of implementation, and compatibility with our existing ecosystem.

Chapter 3

Project outcomes

3.1 Case study

This chapter provides a detailed account of the stages that led to the realisation of the Proof of Concept (PoC), beginning with the design phase and concluding with the functional demonstrations. The chapter begins with a description of the case study, and then proceed to an analysis of the needs and requirements identified for the realisation of the project. Subsequently, the development platforms and devkits employed for implementation are presented, followed by a demo of the final products. Particular emphasis is placed on the sophisticated methodologies adopted to minimise energy consumption. These have been implemented and evaluated on the prototypes developed, but could be applied to any ESP32-based device with specific requirements. It was decided that a section should be dedicated to this topic, as although it was initially considered an optional objective, further investigation revealed a significant and intricate field that necessitates specific technical expertise at both the hardware and software levels. This investigation has also paved the way for potential future developments, which will allow the solutions adopted to be refined, as are discussed in the conclusions.

The project was undertaken in collaboration with VIMAR, a company with a significant presence in the field of electrical and electronic solutions for residential and commercial buildings. VIMAR, which was established in 1945, has become a leader in the field of electrical systems and home automation. The company is renowned for its extensive range of products, which encompasses home automation systems that facilitate the integration and control of diverse household functions, wiring devices that guarantee safe and efficient electrical connections, and sophisticated video door entry systems that enhance security and communication at the entry points of residential and commercial buildings. This thesis project was specifically aligned with the video door entry sector.

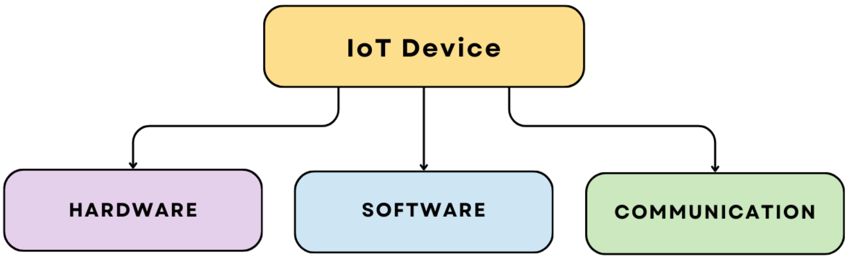


Figure 3.1: Current consumption declared by Espressif in automatic light sleep mode with frequency management enabled.

The development of a product within the context of the Internet of Things (IoT) necessitates a meticulous consideration of three fundamental aspects: hardware, software, and communications (Fig. 3.1) [50]. Each of these elements plays a pivotal role in the creation of an efficient IoT device. The hardware encompasses the physical components, including sensors, that enable the device to interact with its environment. It is therefore essential to ensure the robustness and reliability of the hardware design in order to guarantee the durability and functionality of the product. The software aspect of IoT product development encompasses the programming and algorithms that control the device's operations and process the data it collects. Effective software development ensures that the device performs its intended functions accurately and efficiently. Finally, communication represents the foundation of the Internet of Things (IoT) devices, enabling the transfer of data between devices and with central systems.

The development of the PoC followed a structured approach, divided into four primary phases:

1. **Requirements elicitation and analysis.** In the initial phase of the project, the essential requirements and needs of VIMAR were identified and gathered in order to ensure that the project was aligned with the company's strategic goals. Additionally, a thorough investigation was conducted into the existing solutions on the market in order to identify the optimal methodologies for integrating novel technologies into the project;

2. **Design.** In this phase, multiple design solutions were conceptualised, with a particular focus on the creation of a device that meets the requirements for real-time communication and power management. The critical components of each design were identified with a view to facilitating future enhancements and ensuring compatibility with other devices in VIMAR's ecosystem. The design process also involved the careful selection of the development kit and development platform, ensuring that they were aligned with the technical requirements and objectives of the project. The objective of this phase was to achieve a balance between innovation and practicality, ensuring that the proposed solutions were both feasible and effective;

3. **Implementation and verification.** Upon the conclusion of the design phase, the subsequent phase of development commenced. This phase

involved the construction of prototypes and the implementation of the designed solutions. Rigorous testing was conducted with the objective of evaluating the performance, reliability, and energy efficiency of the demo products. Any issues encountered during testing were duly documented and the solutions were then refined on an iterative basis in order to address these challenges;

4. **Evaluation and validation.** The final phase of the study was dedicated to a meticulous examination of the prototype. The device was then evaluated against the initial requirements and objectives, with particular attention paid to the strengths and areas for improvement. The evaluation also included a comparison of the final product with existing solutions.

This section on the various phases of product development has explored the steps involved in transforming a conceptual idea into a tangible product. From the initial conceptualisation and design to the prototyping, testing, and iterative refinements, each phase of the product development process plays a crucial role in shaping the final outcome. As the product development progresses, it becomes essential to align these processes with the specific needs and requirements of the project.

3.1.1 Project needs and requirements

In project management, a fundamental aspect of achieving project objectives is the comprehension of industrial needs. This section seeks to delineate these components, thereby offering a structured approach to project planning and execution. The clarity of these elements can considerably influence the success of the project, the satisfaction of the company, and the overall quality of the deliverables [24].

The concept of 'Project needs' refers to the higher-level objectives or issues that a given product or software is designed to address. These are typically expressed from the perspective of the end-user or stakeholder and represent the underlying motivations for developing a product or system. The nature of needs is often vague and general, focusing on the desired outcome or the problem to be solved rather than on the specific means of implementation. The following list comprises the needs that have been identified through discussions with stakeholders and analysis of business demands. In accordance with industrial requirements, the list of needs, as well as, requirements is divided into three groups.

The objective of the first group is the creation of a smart peephole that has the sole function of capturing and transmitting video from outside the door to the video intercom (Tab. A.1). The second objective deviates from this approach and is focused on developing a device that is solely intended for wireless audio transmission (Tab. A.2). The third group of needs is based on the knowledge gained from the first two devices. It comprises a smart peephole that provides both an audio and a video input stream (Tab. A.2).

In accordance with the identified needs, the requirements for the three products were defined.

Project requirements, derived from the identified needs, are more detailed and represent the specific conditions or capabilities that must be met for the project to be considered successful. Requirements can be classified into two principal categories: functional requirements and non-functional requirements. Functional requirements specify the precise behavioural and functional characteristics which the project must exhibit. In a software project, for example, such requirements may dictate that the application should enable users to create accounts, log in and perform actions [110]. In contrast, non-functional requirements concern the quality attributes of the project, including performance, security and usability [36]. These requirements are equally important, as they define the extent to which the project will perform its intended functions. The following list (Table A.4, A.5, A.6) details the requirements that have been derived from the needs that were identified in the previous phase.
The result of the process of identifying and defining the needs and requirements of the project has led to the formulation of three distinct products, the respective functionalities of which can be summarised as follows:

Smart peephole. The smart peephole is a device that is placed on the door of a residence and serves to replace the conventional optical peephole. It is equipped with a camera that enables the transmission of video images to the company's video intercom. The device offers three transmission initiation options: one that is triggered by the video door phone interface, another that is initiated when someone rings the doorbell, and a third that is initiated by a physical button on the peephole. Moreover, the device must be constructed on a low-cost platform in order to maintain affordable hardware costs. It must be capable of integrating with the company's existing ecosystem and of operating on battery power.

Audio device. The second device is a distinct entity from the aforementioned peephole, designed exclusively for audio transmission. It must also be based on a low-cost platform and must integrate seamlessly with the existing business ecosystem. Furthermore, it must possess the ability to establish audio calls with a target device, and to send specific commands to that device in order to trigger an action.

Smart peephole plus. The third and final device was conceived as a natural combination of the knowledge gained from the development of the previous two products. It is in fact a smart peephole, with all the features of the first device, but with the ability to capture and transmit both a video and audio stream.

In this section we have outlined the case study in question, identified the specific needs of the company and proceeded to the detailed definition of needs and requirements. These elements led to the conception of three different products, each of which aimed to satisfy the needs that emerged from the preliminary analysis. In the next section we examine the hardware platforms chosen for the development of the demos of these products. We describe the boards used, their main features, and the software development kit (SDK) used to implement the required functionality. This allows us to better understand the technological choices behind the development of the prototypes and to assess the potential of the proposed solutions.

3.2 Development platforms

Following the selection of Espressif's ESP32 as the low-cost platform for the project (Section 2.1.3), an investigation was conducted into the software development kits (SDKs) and available development kits (DevKits) board that could support the implementation of the products demo.

3.2.1 Software development kit

A software development kit (SDK) is defined as a collection of software tools, libraries, documentation, code samples, processes, and guides that are utilized by developers to create applications for specific platforms, frameworks, or hardware environments. SDKs are an indispensable component of the software development process, providing the essential building blocks for the creation of software solutions capable of interacting with specific systems, including operating systems, devices, and even larger applications such as games or enterprise-level systems. These kits are designed to streamline the development process for software engineers by providing pre-built components that can reduce development time, ensure consistency, and improve the overall quality of the software being created [94].

Figure 3.2: SDKs components.

An SDK is typically composed of multiple components that collectively assist in the process of developing software applications (Fig. 3.2). These components include [101]:

- **Libraries and frameworks.** SDKs commonly include pre-built libraries and frameworks that provide developers with reusable code for common functionalities. The utilisation of these libraries can markedly diminish the time and effort required for development, thereby enabling developers to concentrate on the design and construction of distinctive features.
- **Documentation.** The provision of detailed documentation is an integral component of any Software Development Kit (SDK). It furnishes developers with the requisite guidance on the effective utilisation of the tools and libraries. Documentation may include references to the application programming interface (API), tutorials, and an exposition of best practices.
- **Code samples.** It is common practice for SDKs to include sample code which illustrates the implementation of particular functionalities. The code samples provide practical examples for developers, facilitating comprehension of the process of integrating the SDK into their applications.
- **Development tools.** Software development kits (SDKs) may include a suite of tools designed to support the development process, including compilers, debuggers, and testing frameworks. Such tools facilitate the processes of code writing, testing and optimisation for developers.
- **Support and community resources.** A robust support system is an indispensable component of any SDK. Such resources may include online

forums, user groups, and access to technical support. Community resources can act as a catalyst for collaboration and the dissemination of knowledge among developers.

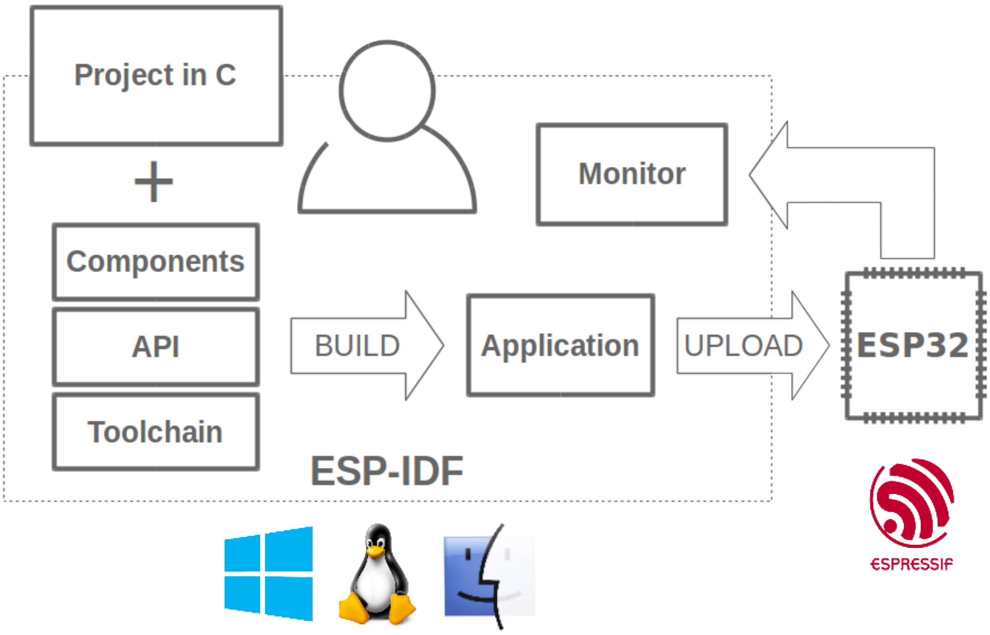


Figure 3.3: Espressif IoT Development Framework.

Source [57].

Espressif IDF. The Espressif IoT Development Framework (ESP-IDF) is the official software development framework provided by Espressif Systems for their ESP32 series of microcontrollers. This open-source framework has been developed with the intention of supporting the creation of Internet of Things (IoT) applications. It offers a wide range of tools, libraries and APIs that facilitate the development of robust, connected devices (Fig. 3.3). ESP-IDF has been developed with the specific requirements of IoT developers in mind, offering complete support for Wi-Fi, Bluetooth, and Bluetooth Low Energy (BLE) connectivity, which are integral features of the ESP32 chip. The framework offers components that enable developers to manage wireless protocols, handle networking, and interface with peripheral devices. One of the most notable features of ESP-IDF is its integration with FreeRTOS, a real-time operating system that enables concurrent multitasking on the ESP32. This is particularly useful for the development of applications that require precise timing and responsiveness, such as those involving sensor data processing, wireless communication, and real-time control systems. In addition to FreeRTOS, ESP-IDF provides drivers for the ESP32's hardware peripherals, including general-purpose input/output (GPIO), inter-integrated circuit (I2C), serial peripheral interface (SPI), and analog-to-digital converter (ADC). This facilitates the development of applications that require interaction with external sensors, actuators, and other hardware components. The build system employed by ESP-IDF is based on CMake, a robust and versatile tool that manages the compilation and linking of the code. This system enables developers to readily configure their projects, manage dependencies, and integrate supplementary libraries as required. ESP-IDF also incorporates a configuration tool (menuconfig) that provides a user-friendly interface for setting various build options and configuring hardware settings [41].

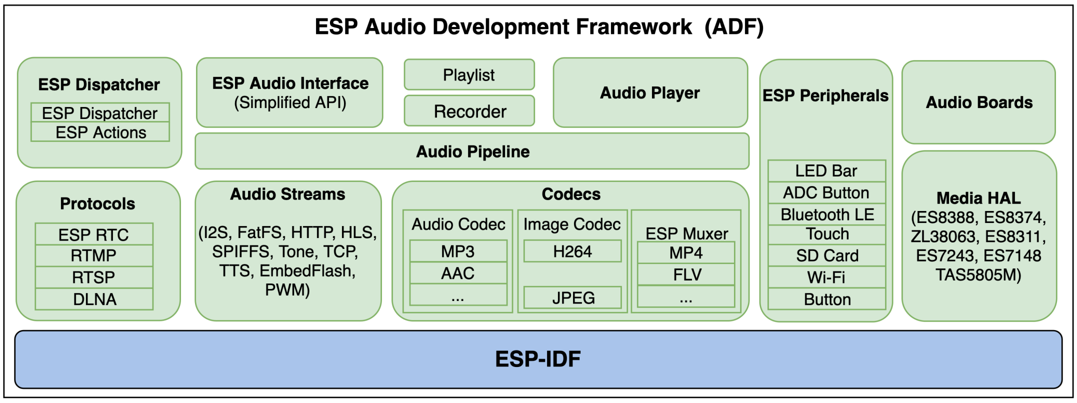


Figure 3.4: Espressif Audio Development Framework.

Source [2].

Espressif ADF. The Espressif Audio Development Framework (ESP-ADF) is a dedicated development framework for the creation of audio applications on the ESP32 microcontrollers and Espressif's DevKit. The ESP-ADF is available as a set of components to extend the functionality already delivered by the ESP-IDF (Fig. 3.4). ESP-ADF is intended for use in real-time audio processing applications and incorporates high-level features tailored to such applications. These include support for various audio/video codecs, audio effects, and input/output interfaces such as I2S (Inter-IC Sound), which is essential for connecting to external audio devices such as microphones, speakers, and digital audio converters. One of the principal features of ESP-ADF is its capacity to process audio streams and pipelines, thereby enabling developers to define and manage complex audio data flows within their applications. This abstraction facilitates the handling of tasks such as decoding audio files, processing the audio data, and the delivery of the resulting sound to different devices. ESP-ADF integrates support for popular streaming protocols, simplifying the development of network-connected audio and video applications, including internet radios, Bluetooth audio devices, and voice assistants. Furthermore, it implements also more advanced protocols such as RTSP (Real-Time Streaming Protocol) and VoIP (Voice over Internet Protocol), making it possible to create sophisticated audio applications that require real-time audio streaming, voice communication, and media transmission over the internet.

Although the Espressif IoT Development Framework (ESP-IDF) and the Espressif Audio Development Framework (ESP-ADF) are powerful and official tools provided by Espressif for developing applications on the ESP32, they are not the sole methods available for creating projects on this platform. It is also possible for developers to utilise alternative development environments and tools, such as the Arduino IDE: it is a widely used and accessible platform that provides support for the ESP32 through the inclusion of supplementary board packages. The user interface is straightforward, and the platform offers a broad library of components, which facilitates the design and implementation of basic projects. An alternative option is MicroPython, which is a compact and efficient implementation of Python 3 that is suitable for use with microcontrollers, including the ESP32. MicroPython is an optimal choice for developers who prefer to write scripts in Python rather than C or C++. Nevertheless, the utilization of ESP-IDF and ESP-ADF offers a number of advantages over alternative methodologies. The most significant benefit is the capacity for fine-grained control, which enables developers to assume greater control over hardware resources and low-level operations. This is of particular relevance in the context of optimizing performance and power consumption, particularly in the complex and resource-constrained domain of Internet of Things (IoT) applications. Moreover, it is also evident that ESP-IDF and ESP-ADF provide a stable and reliable foundation for developers engaged in the construction of professional-grade products that necessitate rigorous testing, certification, and long-term support, a quality that other tools may not offer to the same extent.

Following the decision to utilise ESP-IDF and ESP-ADF for the development process, the subsequent objective was to select the most appropriate development boards. The following section presents the development kits that were selected and utilised throughout the project, with a detailed account of how they met the specific requirements and facilitated the development process.

3.2.2 DevKit board

Development kits, or DevKits, are instrumental tools for engineers and developers engaged in the development of embedded systems and Internet of Things (IoT) projects. The kits typically include a microcontroller or microprocessor, along with a assortment of peripherals, connectors, and debugging interfaces that facilitate the rapid prototyping and testing of hardware and software. The appropriate development kit provides a stable platform that integrates with the selected development frameworks, ensuring that all necessary features, such as connectivity, sensor integration, and power management, can be tested and optimised effectively. This section examines the specific development kits selected for this project, discussing their key features and the advantages they offered in the development of product demos. A different development kit was selected for each set of requirements, as the technical specifications and peripheral devices required are distinct from one another (See Section 3.1.1).

Figure 3.5: Freenove ESP32-WROVER Board.

Source [47].

Freenove ESP32-CAM. The first product demo that we began to develop was a video-only smart peephole. Consequently, we selected the ESP32-CAM (or, more specifically the Freenove ESP32-CAM, a variant of the ESP32-CAM) as our development kit (Fig.3.5). The board is equipped with an ESP32 WROVER-E module, which features 4 MB of flash memory and 8 MB of PSRAM. This is a relatively simple board with exposed GPIOs, which permitted us to undertake a significant amount of testing and to become familiar with the Espressif development framework. The board is equipped with a OV2640 2-megapixel camera and it is capable of wireless connectivity via both Bluetooth and Wi-Fi. By modifying an open-source library [81], it was possible to achieve RTSP streaming with this board. Following an initial period of examination of the library, the primary parameters were identified, including JPEG quality, frames per second, and frame size. These were then made settable by a web server. A series of tests were conducted, and a limit was established for VGA (640×480) video quality at 12 fps to ensure a smooth video without frame loss. Having achieved the objective of smart peephole 1.0, phase two was initiated.

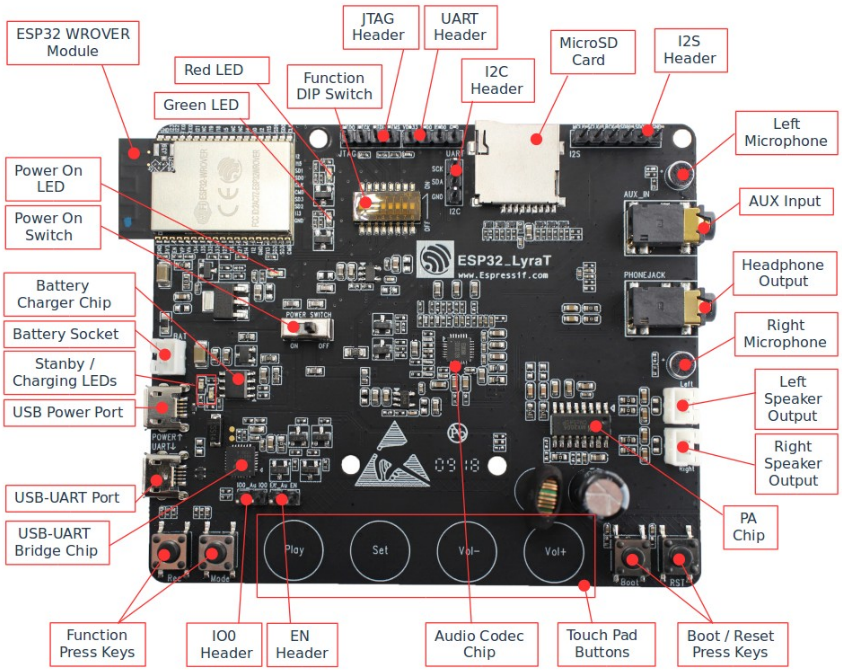


Figure 3.6: ESP32-LyraT V4.3 board layout overview.

Source [39].

ESP32-LyraT. The second phase of the project involves the development of a prototype audio device. In order to process audio, a new board was required, and the LyraT was selected as the optimal choice. The official Expressif devkit is equipped with the same module as the CAM, but with additional peripherals for audio processing, capture and playback. These include a codec (ES8388), microphone and audio jack or speaker output. The aforementioned board enabled access to the tools provided by the ADF framework. Through considerable effort (See Sec. 3.3.2), the board was employed to implement the SIP protocol and achieve bidirectional communication with a target device, utilising the ADF SIP library [4]. Upon completion of the second phase, we began exploring alternative methods for integrating video on the LyraT or audio on the CAM. We conducted research and identified potential solutions. However, we encountered limitations in the number of pins available on the LyraT for a camera, and we had considered external codecs, microphones, and speakers for the CAM. However, this approach would have led to compatibility issues with ADF, which we had previously utilized to implement SIP. Given the constraints in time for developing custom hardware, we decided to procure a third board.

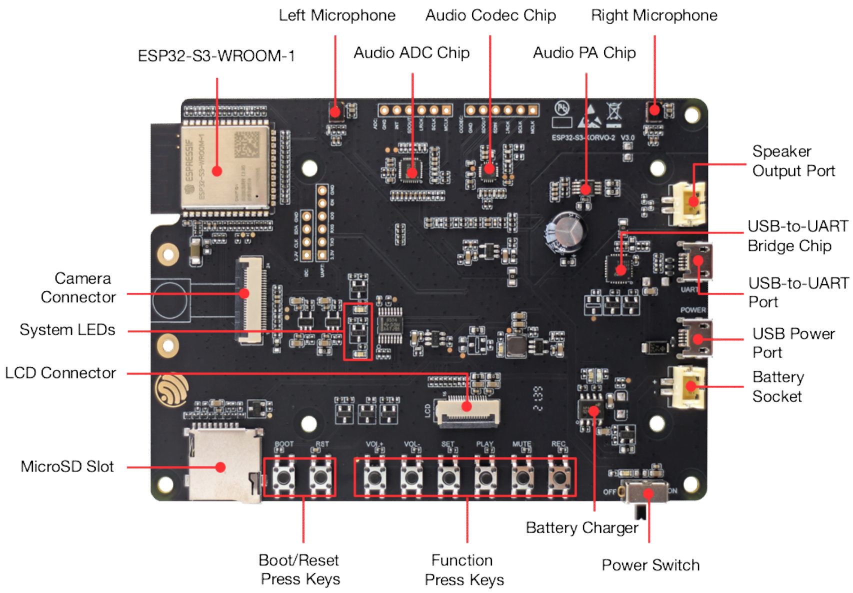


Figure 3.7: ESP32-S3-Korvo-2 V3.0 board layout overview.

Source [40].

ESP32-S3-Korvo-2. The Korvo2, the flagship of the Espressif-branded devKits, features a new S3 chip (aggiungere riferimento al confronto ESP32 vs ESP32S3, magari in appendice), camera connector, new codec (ES8311), separate ADC, microphones and speaker output. Furthermore, the device was purported to be capable of encoding H264 (per frame), a highly advantageous functionality given the objective of establishing SIP audio/video communications with a target device that exclusively accepts H264 as an input format. It became evident at an early stage that this was merely a marketing strategy. While the device is capable of encoding H264, the resulting quality is not satisfactory. The frames produced are of a resolution of 320x192 at a frame rate of 10 fps, which is clearly inadequate. Nevertheless, this board provided us with access to the RTSP protocol implemented by Espressif [3], which exploits the hardware in a markedly more performance-oriented manner than the open source library employed with the Freenove ESP32-CAM. Consequently, we were able to attain HD video quality (1280x720) at 12 fps. Furthermore, the availability of microphones enabled the transmission of not only video but also one-way audio during an RTSP stream.

3.3 Product demo

The creation of product demos is a central component of the development process. Demos provide a concrete visualisation of the product's concept and features, thus enabling stakeholders to visualise the final outcome at an early stage of the development process. They are instrumental in validating design choices, identifying potential issues, and gathering valuable feedback. By repeating the iterative process of creating and refining these demos, it is possible to ensure that the final product meets the desired requirements and expectations, thus leading to a more refined and successful outcome. Two demo products were created in order to meet the first and second sets of requirements, respectively, for the current project.

3.3.1 Cyclop

As previously stated, the objective of Cyclop, the first demo product, is to fulfil the requirements identified by needs group 1 (See Sec.A.1), a video-only smart peephole.

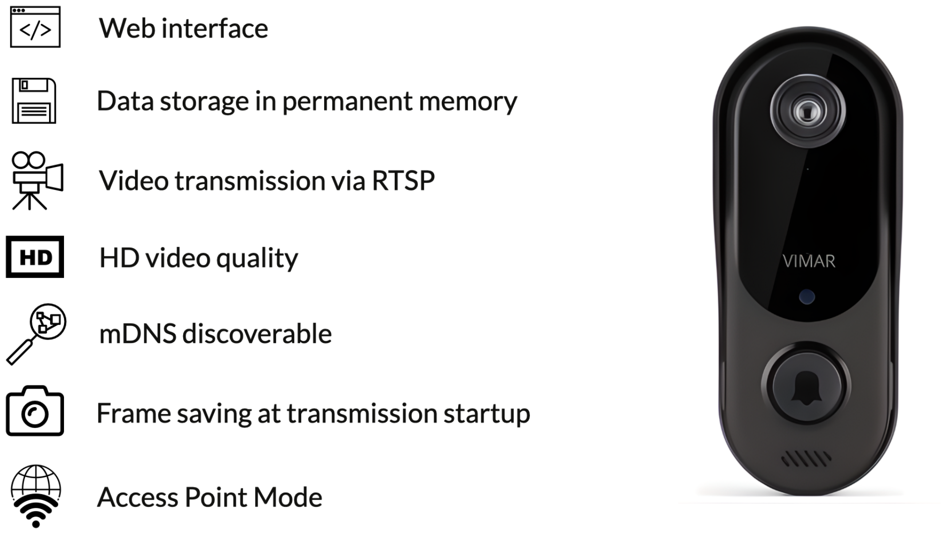


Figure 3.8: Cyclop main features.

The principal characteristics that have been introduced are outlined below.

- **Web interface.** Cyclop offers a suite of web interfaces through which the primary parameters of interest can be configured, including WiFi, camera quality, and video intercom communication. These interfaces are a web server provided by the peephole itself and accessible via mDNS naming. Cyclop and the web server exchange data via web sockets, ensuring that the information displayed in the web interfaces is always synchronized with the active peephole;

- **Data storage.** All configurations set through the Cyclop web interfaces are stored in persistent memory as JSON files within a dedicated partition in the flash memory. The specified files may be accessed at any time via the web interfaces. The management of these files is performed via the SPI Flash File System (SPIFFS) [107];

- **Video transmission.** In the initial phases of the project, Freenove's ESP32-CAM was employed to implement RTSP streaming utilising an open-source library. Following an initial period of evaluation, a VGA

(640x480) video quality with 12 fps was identified as the upper limit. Subsequently, a series of performance tests were conducted on the Korvo2. Leveraging the ADF framework, thanks to the Korvo2, the official Espressif RTSP library was utilized, and through more efficient hardware optimization, an HD quality (1024x720) was achieved at 12 fps. Furthermore, the ADF framework API was also employed to obtain a one-way audio stream, which again was handled via the RTSP protocol. For the integration with the video intercom, for testing purposes, a GStreamer pipeline was utilised to capture both video and audio. However, a complete integration could involve the presence of an RTSP client on the device requesting the audio/video flow.

- **mDNS discoverable.** The peephole can be accessed via mDNS naming, which also enables it to autonomously retrieve the IP address of the video intercom to which it must connect. This avoids potential issues that could arise in the event of a dynamic IP address on the target device.
- **Frame saving.** Upon establishing a connection with the peephole's RTSP streaming, a frame is stored for the initial five seconds. The initial plan was to save this data on a MicroSD, however, due to unforeseen issues with ADF and the Korvo2, specifically the camera's startup procedure, which appeared to interfere with the MicroSD's startup process [27], this approach was deemed impractical. It was therefore decided that, for the time being, a maximum of five frames should be saved in flash memory, which can be accessed via the web interface. However, as a future proposal, it was planned that the frames would be sent to a cloud bucket and a link shared with the user via email for viewing;
- **Access point mode.** By pressing a specific button on the peephole, regardless of its initial state (i.e., whether it has been first activation, is already connected to WiFi, or is in power save mode), it is possible to establish an access point through which all configurations can be modified.

3.3.2 Audio device

The Audio Device's primary functionality (Fig. 3.9) is its support for SIP, which has been achieved through the implementation of the official Espressif SIP API [42]. Specifically, the device supports two-way calling, the sending of SIP messages, and an action button that enables the execution of a predefined action via a SIP message with a specific formatting. Additionally, the device incorporates acoustic echo cancellation, which was only feasible to achieve on the Korvo2 due to the availability of a separate ADC. The remaining functionalities have been previously examined through Cyclop analysis.

Figure 3.9: Audio device main features.

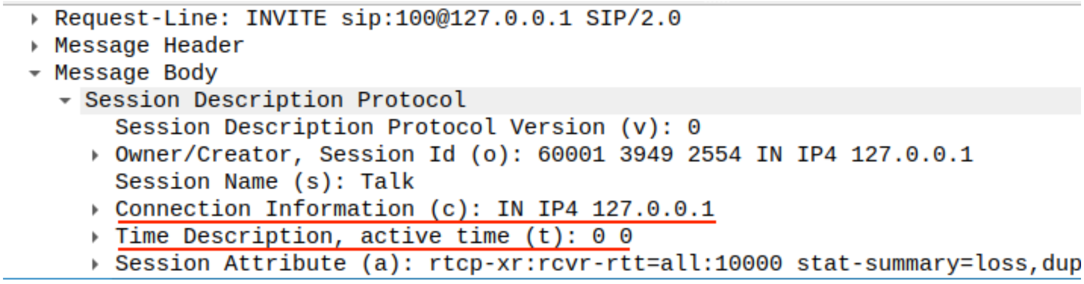
During the development of the Audio Device(AD), a number of issues were encountered. The majority of the issues encountered can be attributed to the ADF framework. Given that ADF is a more recent addition to the IDF ecosystem and has received comparatively less support from the community, having more specific functionalities, it is not surprising that it is less stable and mature than IDF. The issue that has posed the greatest challenge to the development of this AD is the lack of open-source software for some fundamental libraries within the ADF framework. For instance, all the real-time communication protocols are hidden, which has resulted in a series of complications, including delays in implementation, the discovery of bugs, and, in our case, the inability to make modifications. Let us proceed to an examination of the issues that arose during the development of the AD with a particular focus on the above-mentioned functionality (Fig. 3.9).

- **Malformed SDP.** Or, better said, not compatible with the SIP client installed on the targeted device. In particular, the time description **(t)** and connection information **(c)** fields (Fig. 3.10) in the SDP offer of the AD, in the situation where it is receiving an incoming call, are reversed in order. To solve this problem, it was necessary to intervene on the proxy server, which intercepts the SDP of the AD, corrects it, and sends it to the target device;

- **SIP messages not implemented.** The official SIP library from espressif lacks the necessary APIs for the transmission of SIP messages. The transmission of a SIP message by the AD effectively consists of the sending of an HTTP POST request to a proxy server. The proxy server then interprets the contents of the POST, constructs a SIP message and finally transmits this to the intended device;

- **Unable to distinguish SIP server domain and proxy IP.** The Espressif library does not permit the individual setting of these parameters, which is necessary for the authentication of devices in the enterprise ecosystem. This issue was "resolved" by the removal of the authentication process;

- **Early media not implemented.** Prior to responding to an incoming video call, the caller is presented with a preview of the incoming video stream. This feature is not supported by the Espressif library and therefore prevents even audio-only calls. The issue was "resolved" by disabling the specified functionality on the company device;



```
▸ Request-Line: INVITE sip:100@127.0.0.1 SIP/2.0
▸ Message Header
▾ Message Body
  ▾ Session Description Protocol
      Session Description Protocol Version (v): 0
    ▸ Owner/Creator, Session Id (o): 60001 3949 2554 IN IP4 127.0.0.1
      Session Name (s): Talk
    ▸ Connection Information (c): IN IP4 127.0.0.1
    ▸ Time Description, active time (t): 0 0
    ▸ Session Attribute (a): rtcp-xr:rcvr-rtt=all:10000 stat-summary=loss,dup
```

Figure 3.10: Functioning ordering of fields (t) and (c) of the SDP offer.

Before selecting the Espressif SIP protocol, meticulous research was conducted to identify a suitable library that was open source and compatible with IDF. However, the majority of the libraries analysed were found to be overly simplistic and outdated. It appears that following Espressif's SIP protocol implementation, the development community has collectively ceased pursuing alternative solutions. Had Espressif made their implementation of the SIP protocol open source, the issues previously outlined could have been readily addressed by us. However, this is not currently the case, consequently, we deem the Espressif SIP protocol **unsuitable** for a final product.

The Smart Peephole Plus was conceived as a natural combination of the knowledge acquired from the development of the first 2 product demo. However, following the negative response to the SIP protocol, which would have formed the basis of the Smart Plus Peephole's communications, its development was discontinued.

3.4 Power management

At the beginning of the project's power management phase, the audio/video transmission capabilities were assumed to be adequately addressed and the focus was placed on the development of sophisticated power-saving solutions. The objective was to design power management strategies that could be applied not only to the smart peephole, but also to any ESP32-based device with similar requirements. In particular, our research concentrated on devices that require:

1. High level of responsiveness;

2. Wireless remote activation;
3. Bluetooth communication.

This approach has allowed us to develop highly flexible and adaptable power optimisation techniques that can improve the efficiency and battery life of a wide range of IoT products while maintaining high operational performance.

This section presents an analysis of the hardware and software solutions that have been identified as the most effective in terms of maximising battery life while maintaining low response latency. Finally, the results obtained are evaluated, in order to assess their effectiveness, by applying the techniques developed to our smart peephole.

3.4.1 Hardware

The first objective was to create the most reliable environment possible for measuring energy consumption. To achieve this, an initial step was to conduct a detailed study of the various components of the development kits that were being used. By gaining an understanding of the specifics of each component, it was possible to ensure that the measurements would be accurate and reflective of real-world scenarios, thus providing a solid foundation for further optimisation efforts. The results of the conducted analyses have led to the identification of a specific component that could potentially have an impact on the efficacy of any software-based energy-saving strategy. This component is the voltage regulator.

A voltage regulator is an electronic component that is designed to maintain a constant output voltage level, irrespective of fluctuations in the input voltage or the electrical load. In embedded systems, voltage regulators are of particular importance as they protect sensitive components from fluctuations in voltage that could otherwise result in malfunction, instability or even permanent damage [6]. The functionality of embedded systems often relies on microcontrollers, sensors and communication modules that require precise voltage levels in order to operate correctly. If a microcontroller is subjected to a voltage that is too high, it may overheat or be damaged, while a voltage that is too low may prevent it from operating correctly. In selecting a voltage regulator for an embedded system, the quiescent current represents one of the most critical parameters. Often abbreviated as 'Iq', it refers to the amount of current consumed by an electronic device or circuit when it is in a no-load or idle state. This is defined as a state in which the device is powered on but not performing any active work. This current is typically drawn by the internal circuitry of components even when they are not driving any external loads. In the case of battery-powered devices, quiescent current is an essential parameter as it directly affects battery life. In such systems, a lower quiescent current is preferable as it minimises the power consumed when the device is in standby or low-power mode, thereby extending the overall battery life [71].

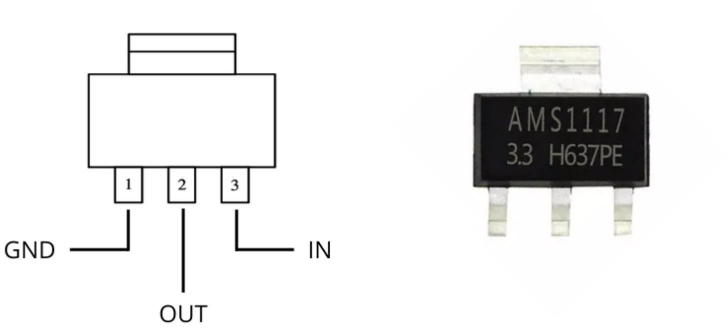


Figure 3.11: AMS1117 Pinout.

Source [9][10].

In all the development kits that have been employed thus far, the voltage regulator that has been installed is an AMS1117 (Fig. 3.11), which is reported to have a quiescent current of 5-11mA [9]. This level of power consumption in an idle state is incompatible with the objectives of our project, which aims to reduce power usage to ensure the viability of a battery-powered IoT device. The high quiescent current would result in a significant drain on the battery even when the device is not actively operating, thereby nullifying the efficacy of the ESP32's sleep modes [38]. Accordingly, a development kit designed for low-power operation was selected.

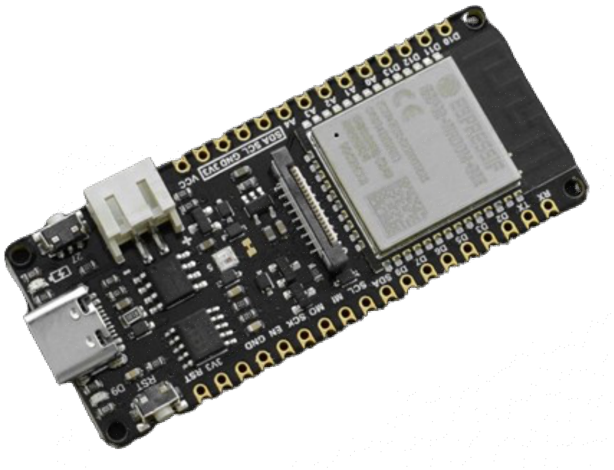


Figure 3.12: FireBeetle 2 ESP32-E.

Source [45].

FireBeetle 2 ESP32-E. The FireBeetle 2 ESP32-E (Fig. 3.12) is a compact, low-power development board designed specifically for Internet of Things (IoT) applications. It is constructed around the ESP-WROOM-32E module and features an integrated charging circuit, making it particularly suitable for measuring current and calculating energy consumption. A low-power solder jumper is present on the circuit board. The removal of the solder jumper results in a reduction of static power consumption by 500 uA. In Deep Sleep mode (Fig.

3.17), FireBeetle reports a power consumption of 13 uA, which is one of the closest power consumptions on the market to the official estimates declared by Espressif for its microcontrollers [38].

3.4.2 Sleep modes

Following the establishment of the hardware foundation for our low-power IoT solution, we proceed to examine the software strategies that were implemented with the objective of optimising power efficiency. In this analysis, we investigate the various software solutions that were implemented, including the configuration of sleep modes to extend battery life when in an idle state, the selection of communication protocols to ensure reliable data transmission, and other essential optimisations. We begin by presenting the various sleep modes that are available from Expressif.

The ESP32 offers five distinct power management modes, each of which can be configured and selected by the user. The chip is capable of switching between modes in accordance with the specific power requirements of the system. Different SoC functional parts are either switched off or paused in accordance with the different sleep modes.

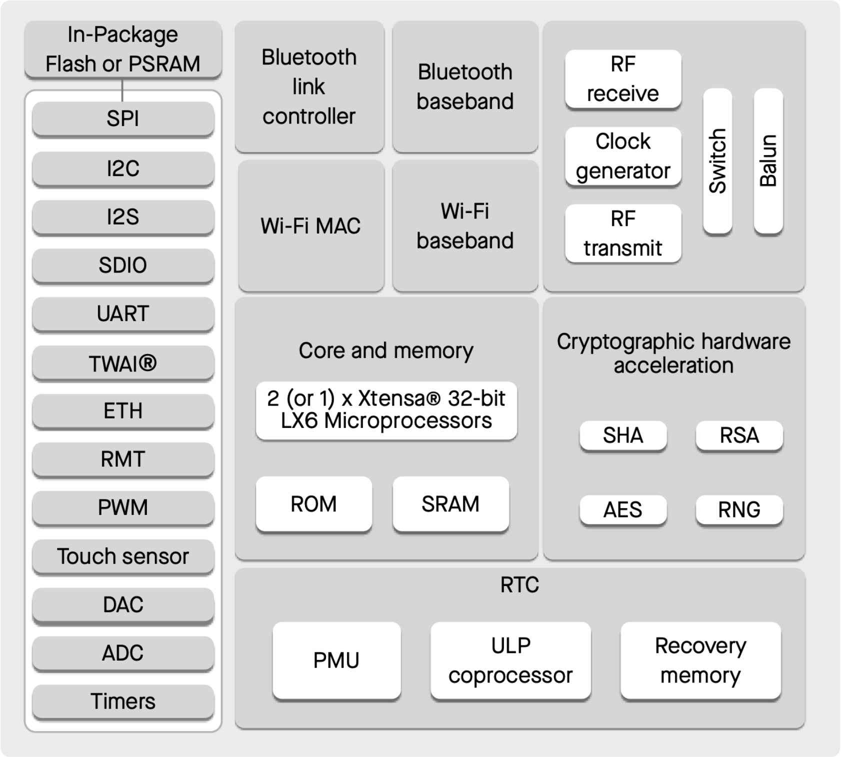


Figure 3.13: ESP32 functional components.

Source [38].

The principal components are as follows (Fig. 3.13):

- **Core CPU and memory.** The central processing unit and memory. The principal components of this section are the microprocessor, cache, static random-access memory (SRAM) and read-only memory (ROM).
- **Radio.** It comprises the essential components for radio frequency (RF) communications. These include The radio module comprises the 2.4 GHz receiver, 2.4 GHz transmitter, bias and regulators, balun and transmit-receive switch, and clock generator.
- **WiFi and bluetooth.** Specific components, such as WiFi MAC, WiFi Baseband, Bluetooth LE Link Controller and Bluetooth LE Baseband.
- **ULP coprocessor.** The Ultra Low Power co-processor has been designed to efficiently handle low-power tasks and sensor operations while the main CPU is in a deep sleep. The co-processor is composed of several components, including the ULP CPU core, memory, sensor interface, timers, and wake-up functionality.
- **RTC.** The Real-Time Clock (RTC) is a component that provides accurate timekeeping and time-related functions. It is used to maintain a record of time, perform time-based operations and schedule wake-up events. The RTC includes the RTC memory, power management unit and ULP.
- **Peripherals.** It encompasses an assortment of hardware components and functions integrated into the chip. These peripherals enhance the capabilities of the chip, facilitating its interfacing with sensors, external devices, and other components. Among the most noteworthy peripherals are UART, system timer, pulse counter, PWM, LED, LCD interface, I2S, I2C, camera interface, GPIO, USB serial/JTAG, and others.
- **Low power peripherals.** This could be considered an extension of the peripherals, designed to operate in sleep mode, thereby consuming a minimal amount of power. The key components include a real-time clock (RTC), a general-purpose input/output (GPIO) pin, an RTC analog-to-digital converter (ADC), an eFuse controller, an RTC watchdog timer and a touch sensor.

We now examine the five sleep modes: active mode, modem sleep, light sleep, deep sleep and hibernation [38].

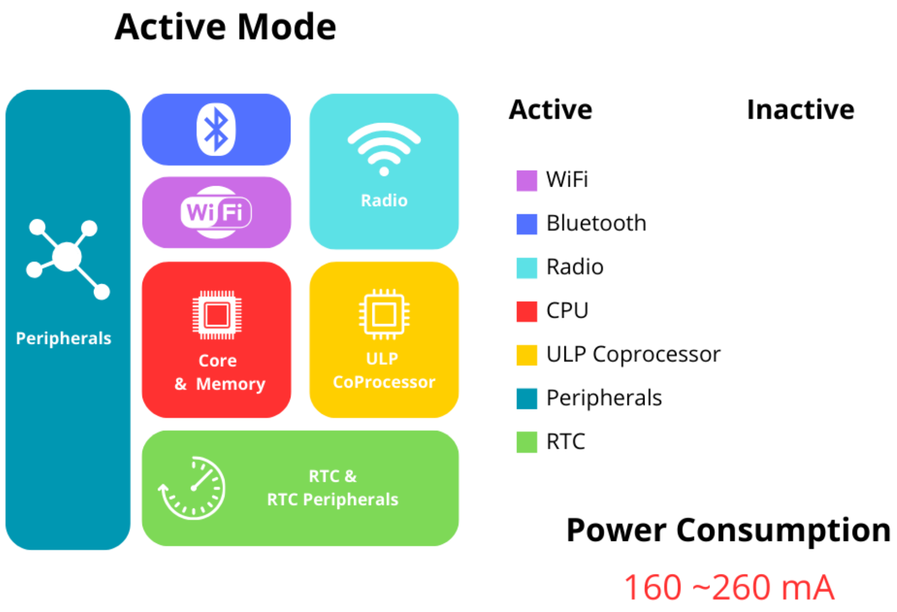


Figure 3.14: Active mode.

In its default state, the ESP32 operates in active mode (Fig. 3.14), which is the standard mode of execution for code. Although it provides the greatest processing power, it is also the mode with the highest power consumption. This is due to the fact that all components, including the radio chip, are active. It is capable of receiving, transmitting and listening to signals.

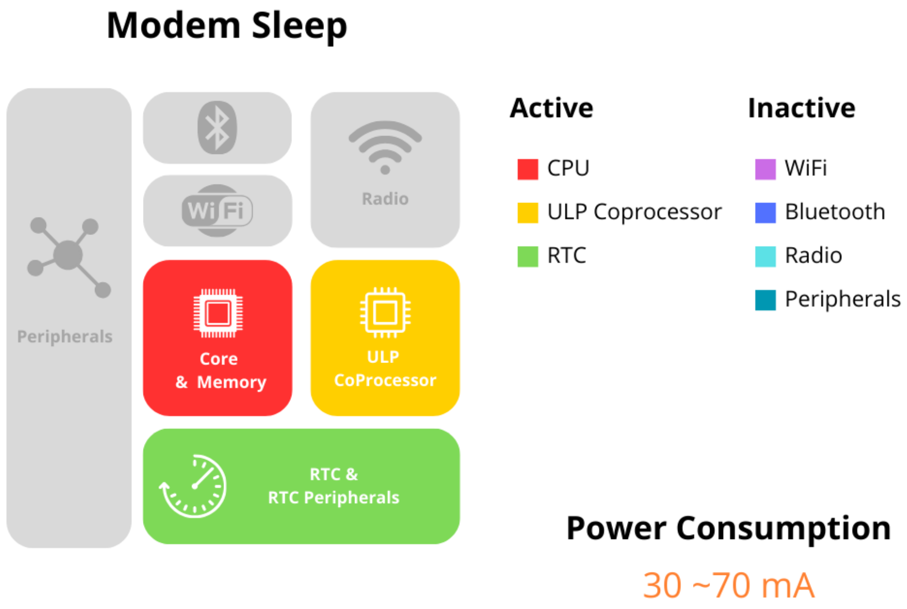


Figure 3.15: Modem sleep.

During the modem sleep mode (Fig. 3.15), the CPU is operational and the clock frequency may be configured as desired. The baseband and Wi-Fi/Bluetooth radio are disabled.

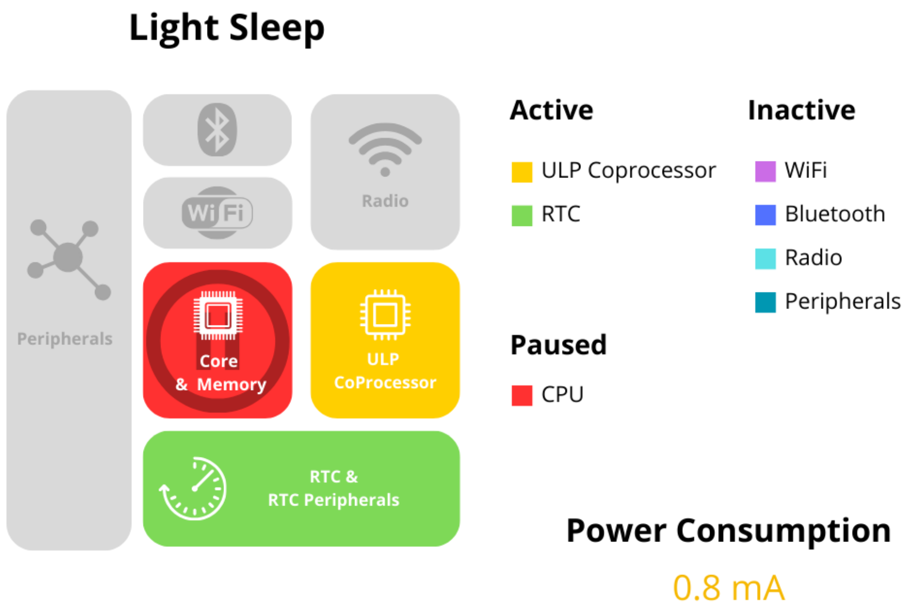


Figure 3.16: Light sleep.

In light sleep mode (Fig. 3.16), peripherals, the majority of RAM and CPUs are clock-gated and their supply voltage is reduced. This results in a reduction in power consumption compared to that observed in modem sleep mode. Upon exiting light sleep mode, the peripherals, RAM and CPU resume operation, and their internal states are preserved.

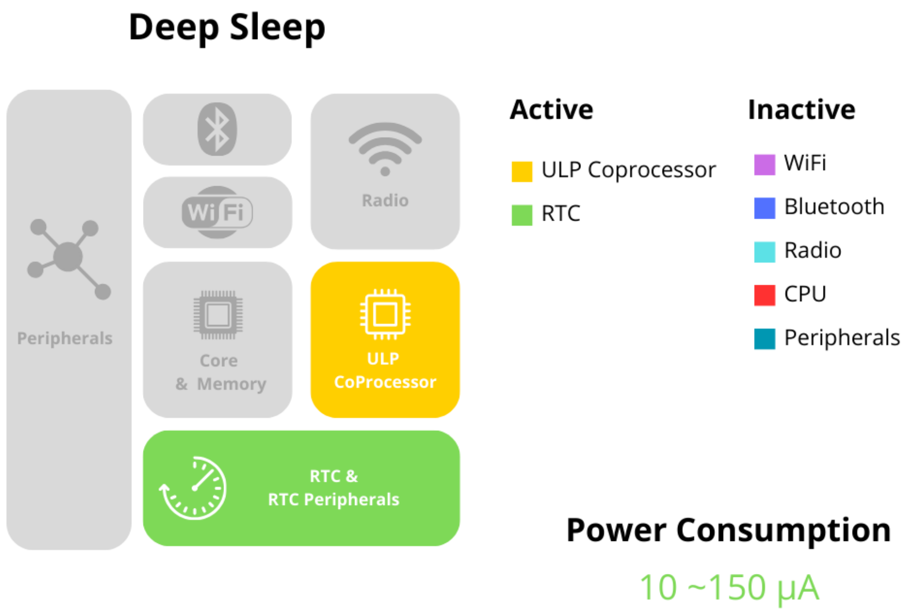


Figure 3.17: Deep sleep.

In deep sleep mode (Fig. 3.17), the CPU is switched off, while the Ultra-Low-Power (ULP) coprocessor remains active, monitoring sensor readings and initiating CPU activation when necessary. This mode is optimal for applications that necessitate CPU activation due to external stimuli, including timers, while maintaining minimal power consumption. The main memory of the chip is deactivated, resulting in the irretrievable loss of all stored data. Nevertheless, the real-time clock (RTC) memory remains operational, thereby ensuring the retention of data during deep sleep and its subsequent availability upon reactivation of the chip. Accordingly, the chip stores the data related to Wi-Fi and Bluetooth connections in the RTC memory prior to entering Deep Sleep mode. Upon reactivation from Deep Sleep, the chip performs a reset and initiates the program from its initial state.

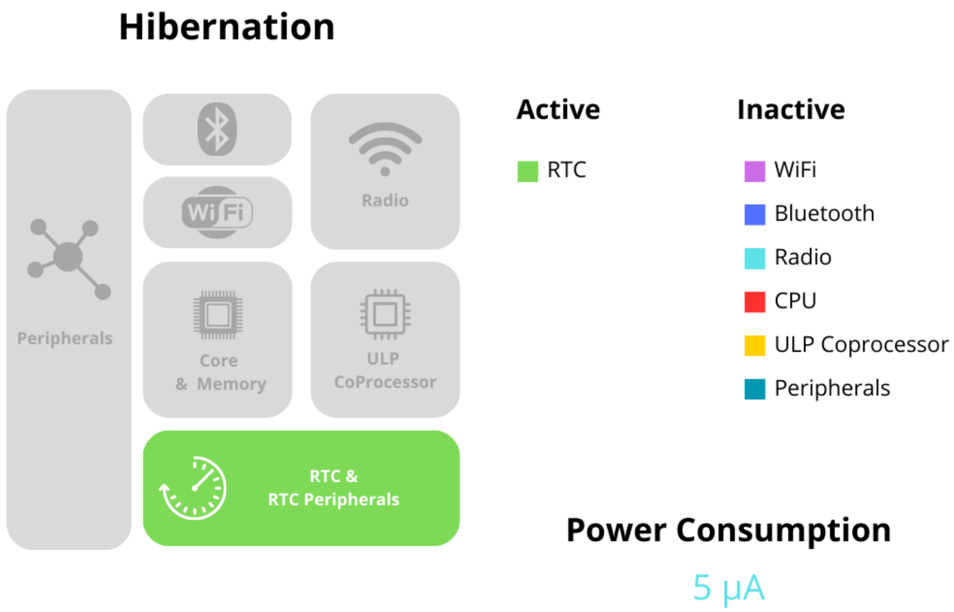


Figure 3.18: Hibernation.

Hibernate mode (Fig. 3.18) is comparable to Deep Sleep mode in terms of functionality. The sole distinction between the two modes is that, in hibernation mode, the chip disables the internal 8 MHz oscillator and ULP processor, leaving only an RTC timer (on slow clock) and a few GPIO RTC active to wake up the chip. Given that the RTC recovery memory is also disabled, it is not possible to save data in hibernation mode. As a consequence, the power consumption of the chip is further reduced.

An understanding of the various sleep modes is indispensable for the comprehension of the adopted solutions. By meticulous selection and configuration of these modes, we were able to prolong battery life while maintaining the requisite responsiveness and functionality for our application. The following section presents an analysis of the selected communication protocol, including an investigation of its implementation.

3.4.3 Communication protocol

The smart peephole is designed to interface with a device that currently only supports Bluetooth as its low-energy communication protocol. Consequently, the BLE protocol is used for communication during periods when video streaming is not active. Bluetooth Low Energy (BLE) is a wireless communication protocol designed for the transfer of data over short distances at low power, particularly within the domain of the Internet of Things (IoT) [51]. BLE was first introduced by the Bluetooth Special Interest Group (SIG) in version 4.0 of the Bluetooth standard, with subsequent enhancements in versions 4.2 and 5.x. These enhancements were designed to improve energy efficiency, range, and connectivity capabilities [102]. The protocol operates in the 2.4 GHz ISM band, utilizing a frequency-hopping spread spectrum to minimize interference and enhance communication reliability [23]. One of the primary advantages of BLE is its ultra-low power consumption, which allows devices to operate for extended periods on small batteries. This characteristic renders BLE particularly well-suited to applications in wearable technology, healthcare monitoring, and smart home devices [49]. BLE devices are capable of remaining in a low-power sleep mode for the majority of the time, waking only to transmit or receive data, which significantly extends battery life [109]. The protocol supports different data transfer modes, including connection-oriented and connectionless communications, allowing for flexibility in application design.

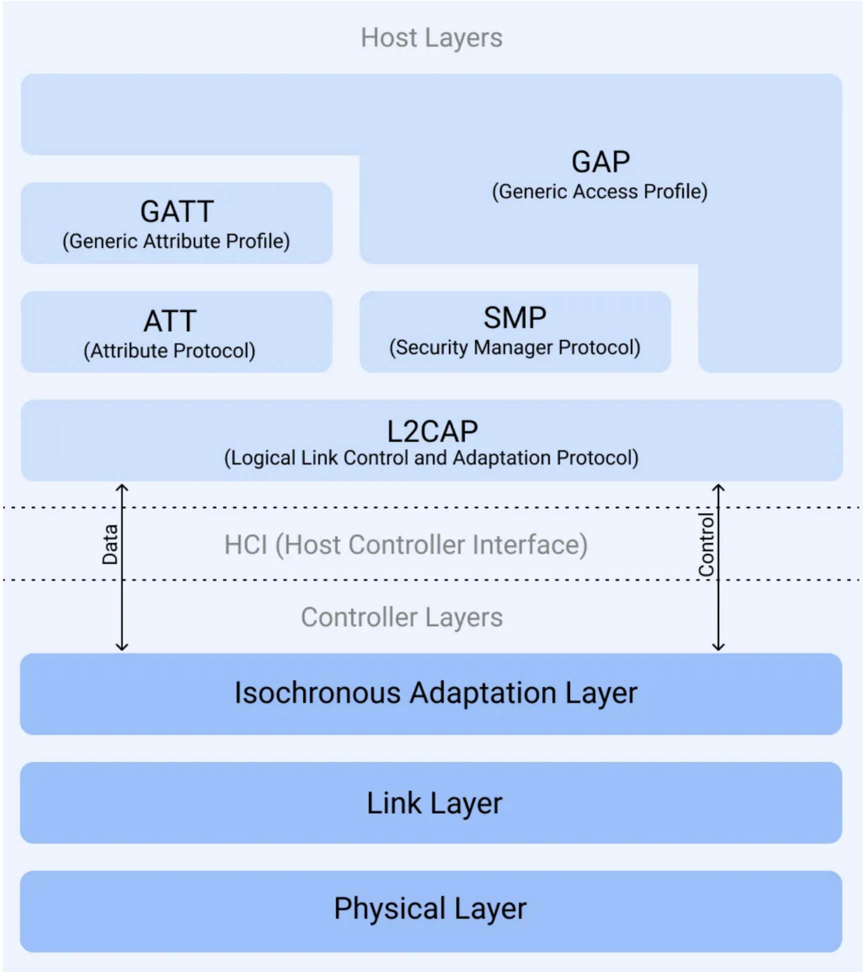


Figure 3.19: Bluetooth Low Energy stack.

Source [21].

The Bluetooth Low Energy (BLE) stack consists of several layers (Fig. 3.19), each responsible for different aspects of communication [22]:

- **Physical layer (PHY).** The physical layer (PHY) represents the lowest layer of the Bluetooth Low Energy (BLE) protocol stack. It is responsible for the physical method used to transmit information, which in this case is the utilisation of radio frequency (RF) waves for the transmission and reception of raw bits over the air.

- **Link layer.** It is an integration of hardware and software, and is responsible for the management of advertising, scanning and the establishment and maintenance of connections. It provides the initial level of control and data structure over the raw radio operations, such as packet structure.

- **L2CAP (Logical link control and adaptation protocol).** This layer is responsible for data integrity, it handles data framing, multiplexing, and segmentation/reassembly of data packets.

- **SMP (Security mananger protocol).** It provides services to other layers for the purpose of establishing a secure connection and ensuring the secure exchange of data between two BLE devices.

- **GAP (Generic access profile).** It is responsible for defining the roles of devices in BLE communication. The GAP defines four device roles: broadcaster, observer, peripheral and central. In a connectionless communication (broadcasting), where the exchange of data between devices occurs without first establishing a connection, the Bluetooth LE devices involved assume the roles of Broadcaster and Observer. A Broadcaster is expected to share the information, and an Observer is expected to be listening for such information. In a connection-oriented communication, where data is only exchanged after a connection has been established between two devices, the Bluetooth LE devices involved assume the roles of Peripheral and Central.

- **ATT (Attribute protocol).** The Attributes Protocol is used to store Profiles, Services, Characteristics and Attributes in a simple lookup table. The protocol employs 16-bit identifiers for each entry, facilitating a straightforward and efficient storage system. The protocol enables client-server communication through the reading and writing of attributes, which constitute the basic data units in BLE.

- **GATT (Generic attribute profile).** GATT is built on top of ATT and is responsible for defining the manner in which data is organised and accessed on a BLE device. It specifies the procedures for service discovery, reading, writing, and notifications. GATT organizes data into services and characteristics. A service is defined as a collection of related characteristics. To illustrate, a heart rate service may comprise characteristics such as heart rate measurement and body sensor location. A characteristic, on the other hand, is a data point or feature of a service. It contains a value and may also include descriptors that provide additional information about the characteristic, such as the format of the data or its unit of measurement.

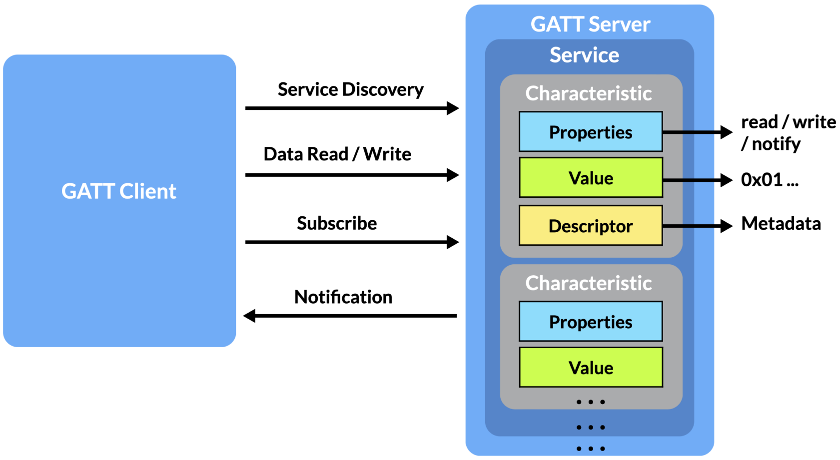


Figure 3.20: Communication between a GATT client and a GATT server.

The communication between a GATT client (typically a smartphone or computer) and a GATT server (the BLE device) adheres to a structured process [22].

- **Service discovery.** Upon establishing a connection with the server, the client initiates a service discovery process to identify the available services and characteristics on the server;
- **Read/write operations.** The client is able to read the values of the characteristics or, alternatively, to write new values to them. To illustrate, the client may read a temperature value from a characteristic or write a command to initiate an action on the server;
- **Subscribe.** The client is able to subscribe to specific characteristics, thereby receiving updates when the values of those characteristics change;
- **Notifications.** In the event of a change to a characteristic subscribed to by the client, the server transmits a notification to the client. This enables the client to receive updates in an asynchronous manner, obviating the necessity for continual polling of the server.

Having delineated the characteristics of both the sleep modes and the communication protocol, we may now embark upon an examination of the proposed solution. This solution employs a combination of these elements to achieve optimal power consumption while maintaining reliable and responsive communication.

3.4.4 Proposed solution

This section presents a review of the initial approach taken in developing the energy optimisation system, along with an analysis of the challenges and limitations encountered. The analysis facilitate an understanding of the methods employed to overcome these issues, leading to the development of the final optimised solution.

As a battery-powered device, the Smart Peephole must adopt strategies to conserve energy for the majority of the time. The most energy-efficient mode is the deep sleep mode, which is activated when the device is not in use. Nevertheless, it is imperative that the device remains in an active listening state and is capable of rapid activation upon receiving a command to initiate an RTSP audio/video stream. Potential inputs for initiating transmission include:

1. A physical button located on the exterior of the Smart Peephole;
2. A virtual button accessible via the Video Intercom (VI) graphical user interface (GUI);
3. An external doorbell.

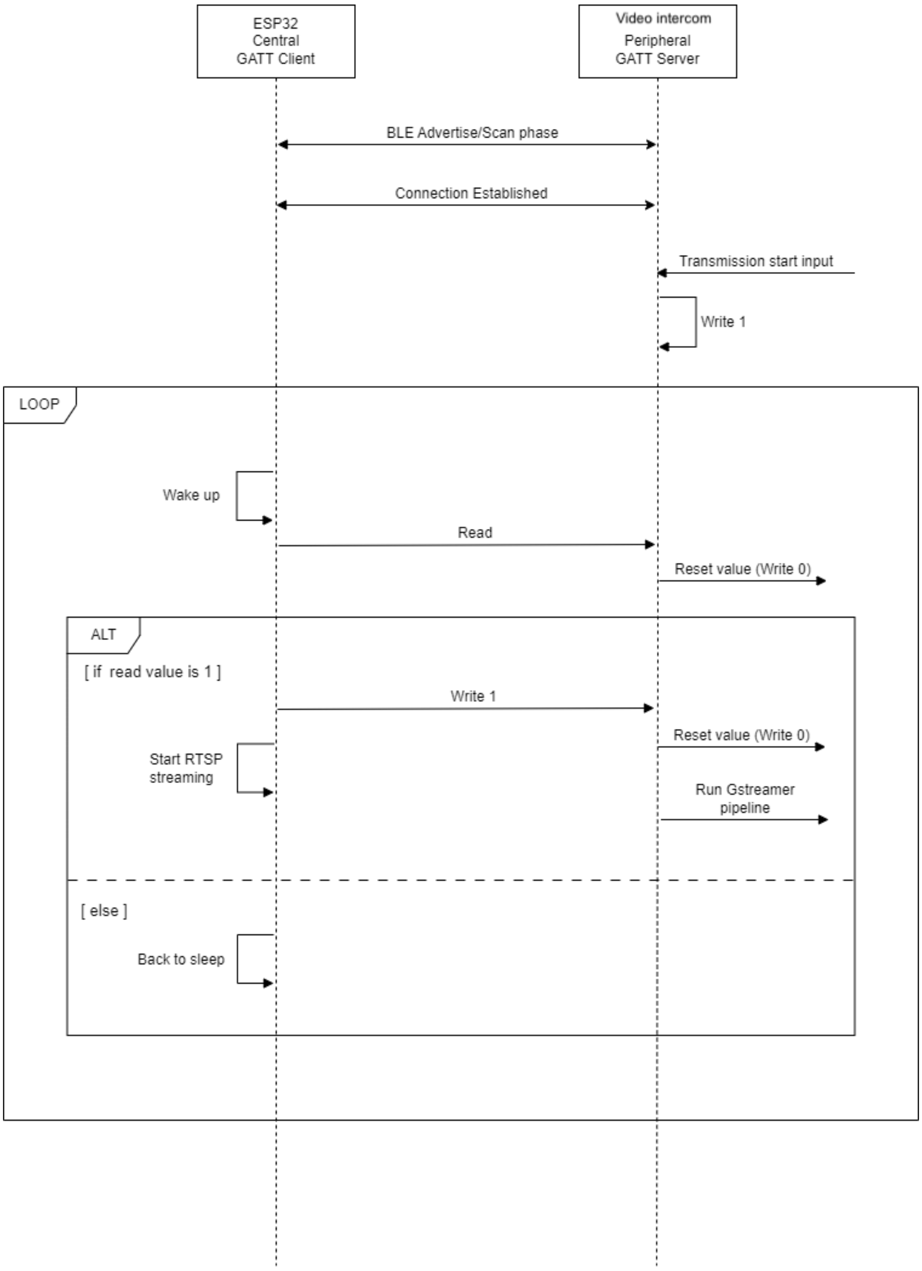


Figure 3.21: Flowchart of polling method.

Polling. In order to provide an explanation of the functioning of polling mode, it is necessary to consider the potential occurrences.

1. The physical button located on the exterior of the Smart Peephole has been activated. In this situation, the peephole will become active, transmit the specified data, and then initiate the RTSP streaming process. Upon receipt of the write command, the VI will execute the GStreamer pipeline in order to display the video streamed by the Peephole.
2. One of two scenarios has transpired: either the doorbell has been pressed, or a request has been made to initiate the transmission from the VI's interface. This event is illustrated in the flowchart above (Fig. 3.21). Cyclop is designed to wake up every few seconds to check for new requests. If it has received requests, it will start the Wi-Fi and the RTSP streaming; otherwise, it will return to a sleep state. The proposed methodology entails the implementation of a timer-based awakening mechanism.

For enhanced readability (Fig. 3.21), the phase of reconnection subsequent to the ESP wake-up call has been excluded. Additionally, the Reset value command (a Write 0) is executed in response to the onRead event, thus ensuring that the outcome of the reading remains unaffected.

It is evident that this option is not a sustainable solution, despite the fact that the deepest sleep mode is utilised, which results in the lowest power consumption when idle. When the device enters deep sleep, the BLE connection to the video intercom is lost. Consequently, the wake-up times (500 ms) in conjunction with the reconnection times (2/2.5 seconds) result in a total of 2.5 to 3 seconds of awake time. This equates to 45 seconds of wakefulness in a single minute, which is a considerable amount of energy consumption. In light of the identified issue with the reconnection time, we considered utilising a lighter sleep mode to maintain an always-active connection with the VI.

In order to gain a complete understanding of the proposed solution, it is first necessary to provide a brief explanation of the manner in which communication occurs between two devices that are connected via BLE.

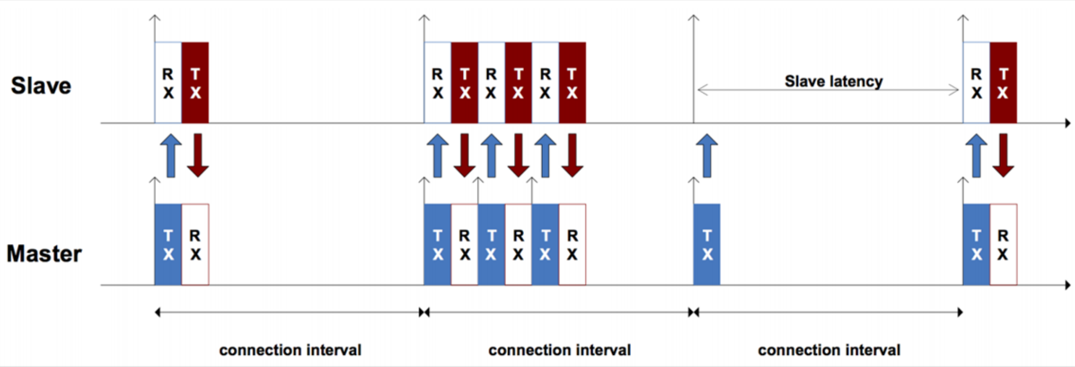
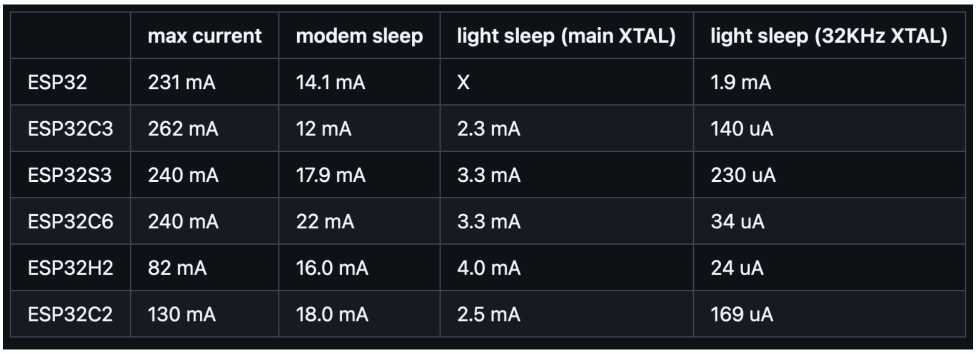


Figure 3.22: BLE connection interval.

Source [**site:ble_conn_itvl**].

In Bluetooth Low Energy, the connection interval (Fig. 3.22) represents a central parameter that determines the frequency with which two connected devices exchange data. It defines the temporal interval between consecutive data packet exchanges, which are referred to as "connection events." Once two BLE devices are connected, they establish a connection interval, which typically ranges from 7.5 milliseconds to 4 seconds. During each connection event, the devices are capable of sending and receiving data packets. A reduction in the connection interval will result in a greater frequency of communication between the devices, which may allow for faster data transfer and lower latency. However, this will also lead to an increase in power consumption. Conversely, a longer connection interval results in a reduction in the frequency of communication, which conserves battery life but may result in an increase in latency.

Automatic light sleep. The solution in question entails the autonomous switching of the chip's operating modes. This occurs when the processor is idle for a minimum of three ticks, which is the default value and corresponds to 30 ms if the tick rate is 100 Hz. With a connection interval of 1000 ms, the ESP32 will remain in light sleep mode for approximately one second. Thereafter, it will wake up, transmit the connection event to maintain the connection (or send/receive data), and finally return to sleep mode. In addition to the automatic light sleep function, this solution enables the dynamic variation of the processor frequency, allowing for a reduction in power consumption during sleep periods through the utilisation of a lower frequency. Throughout the sleep phase, the frequency can be lowered to 20MHz, maintaining connectivity.



	max current	modem sleep	light sleep (main XTAL)	light sleep (32KHz XTAL)
ESP32	231 mA	14.1 mA	X	1.9 mA
ESP32C3	262 mA	12 mA	2.3 mA	140 uA
ESP32S3	240 mA	17.9 mA	3.3 mA	230 uA
ESP32C6	240 mA	22 mA	3.3 mA	34 uA
ESP32H2	82 mA	16.0 mA	4.0 mA	24 uA
ESP32C2	130 mA	18.0 mA	2.5 mA	169 uA

Figure 3.23: Current consumption declared by Espressif in automatic light sleep mode with frequency management enabled.

Source [31].

This mode has specific requirements. In the case of the ESP32, an external quartz crystal is necessary to maintain the wake-up timer. Conversely, this component is not required for the S3 variant, although its utilisation results in markedly reduced power consumption (Fig. 3.23).

3.4.5 Results

This section presents the results of tests conducted to measure energy consumption. In order to accurately assess the energy consumption of the device, it was essential to quantify the current drawn by the development kit in each of its operational states, both during periods of inactivity and during audio-video transmission. The data obtained from these measurements enabled an accurate estimation of the energy required under different operational conditions, providing valuable insights for enhancing the device's efficiency. To ensure accurate current monitoring, specific equipment was employed to perform these analyses, which are described in detail below.

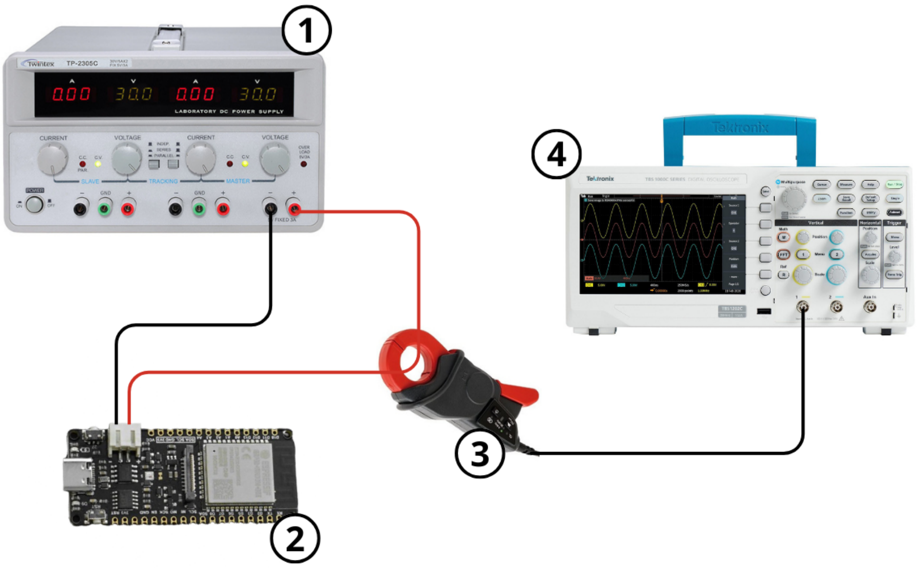


Figure 3.24: Equipment used to measure energy consumption.

As illustrated in the above figure, the components of the system include:

1. **Laboratory DC power supply.** It is defined as a device that provides a stable, adjustable source of direct current (DC) voltage and current. It is frequently employed in the testing and development of electronic devices, enabling the user to power circuits or components within a controlled environment. The adjustable voltage and current limits prevent damage to sensitive devices and facilitate accurate measurements for performance evaluation;
2. **FireBeetle2 ESP32-E.** The devkit which has been designed for use in low-power applications, as previously mentioned 3.4.1. The device is powered via the battery socket, which is supplied by the DC power supply;
3. **Current probe.** It is a device that is used to measure the electric current that is flowing through a conductor without there being a direct connection to the circuit. The device is secured around the wire or conductor and employs magnetic induction to accurately detect the current flow. Current probes are a common tool in electrical testing and diagnostics, facilitating the safe measurement of alternating current (AC) or direct current (DC) without interrupting the circuit;

4. **Digital oscilloscope.** It is an electronic test instrument that is used to visualise electrical signals over time, displaying voltage waveforms on a screen. It allows engineers to observe signal behaviour, such as amplitude, frequency, and any irregularities. When used in conjunction with a current probe, the oscilloscope displays the current waveform passing through a conductor. This combination is useful for the real-time analysis of current flow, the measurement and monitoring of power consumption, the detection of faults, or the observation of how current changes during the different operational states of a device.

In this configuration, the DC power supply serves to power the ESP32 via the battery socket, thereby providing a stable and controlled voltage source. The current probe is situated between the ESP32 and the power supply, enabling the measurement of the current entering the device. The data gathered by the current probe is subsequently displayed on the oscilloscope, facilitating the real-time visualisation of the current waveform. Utilising the oscilloscope, the mean power consumption values were determined and now be discussed in detail.

In order to obtain an overall estimate of the energy consumption of the Smart Peephole, a combination of two boards was employed: the ESP32-S3-Korvo2 was utilised for the RTSP streaming phase, while the FireBeetle2 ESP32-E was employed during the sleep periods. The rationale behind the utilisation of two distinct boards can be attributed to the Korvo2's excessive power consumption during its sleep phases, as elucidated in Section 3.4.1.
To determine the battery life of the Smart Peephole, an average usage of five audio/video transmission events per day, each lasting 30 seconds, was estimated to be in excess of the actual usage.

In particular, the objective is to determine the consumption based on a 24-hour period of the following events:

A. Five activation events of the WiFi mode and five preparation events for RTSP streaming;

B. Five transmission events, each lasting 30 seconds, of RTSP;

C. The average daily consumption in Automatic light sleep mode.

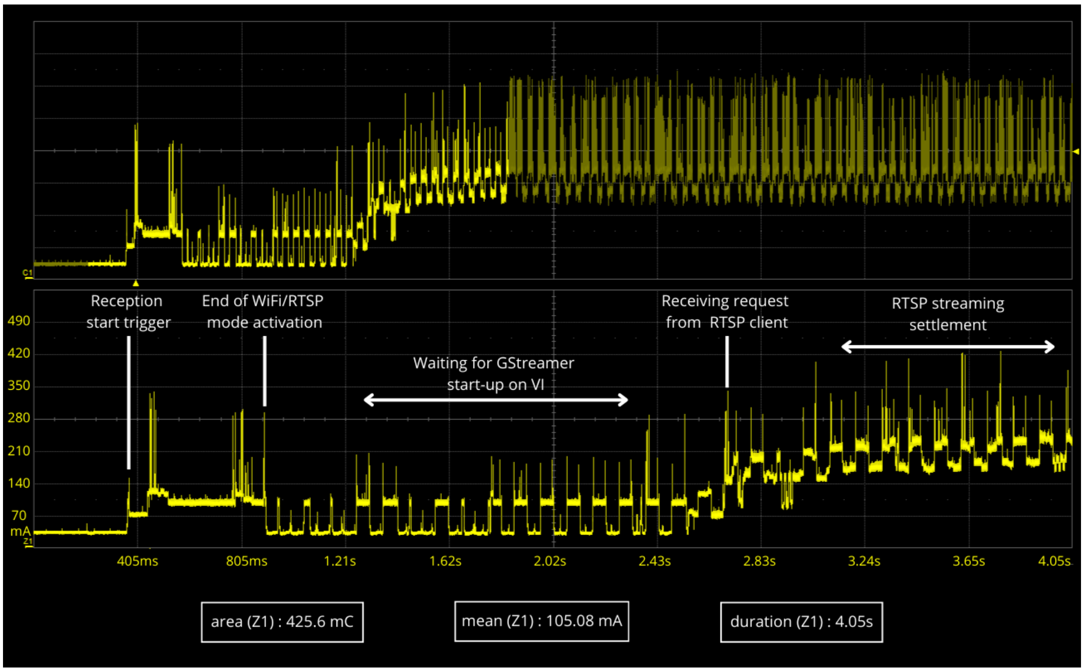


Figure 3.25: ESP32-S3-Korvo2 measurement WiFi/RTSP activation event.

A event. As illustrated in the diagram above (Fig. 3.25), the WiFi/RTSP mode start event is comprised of a series of sub-events: Upon receipt of the start trigger, the following sequence of events is initiated [1]:

1. Receiving the start trigger, either the doorbell has been pressed, or a request has been made to initiate the transmission from the VI's interface.;
2. Initialisation of the WiFi and RTSP components:
   - Average duration, from reception of the start trigger: **500ms**.
3. Waiting for GStreamer pipeline start on VI client:
   - Average duration: **3 seconds**.
4. Receipt of streaming start request from the client:
   - Time until display of first frames: about **500ms**.
5. Start audio/video transmission and settling:
   - Average duration **800ms**.

The mean duration of the event was 4.5 seconds, with an average energy consumption of 105 mA. In order to determine the number of milliampere-hours (mAh) associated with this event, it is necessary to calculate the electrical charge:

---

[1] The averages presented are based on a number of measurements and are not limited to the figure shown above.

$$1\,\mathrm{mC} = 1\,\mathrm{A} \times 1\,\mathrm{ms} \tag{3.1}$$

hence:

$$\textbf{Electric charge} \ = 0.105\,\mathrm{mA} \times 4500\,\mathrm{ms} = 472.5\,\mathrm{mC} = 0.1312\,\mathrm{mAh} \tag{3.2}$$

It can thus be concluded that the average power consumption of event A is 0.13mAh.

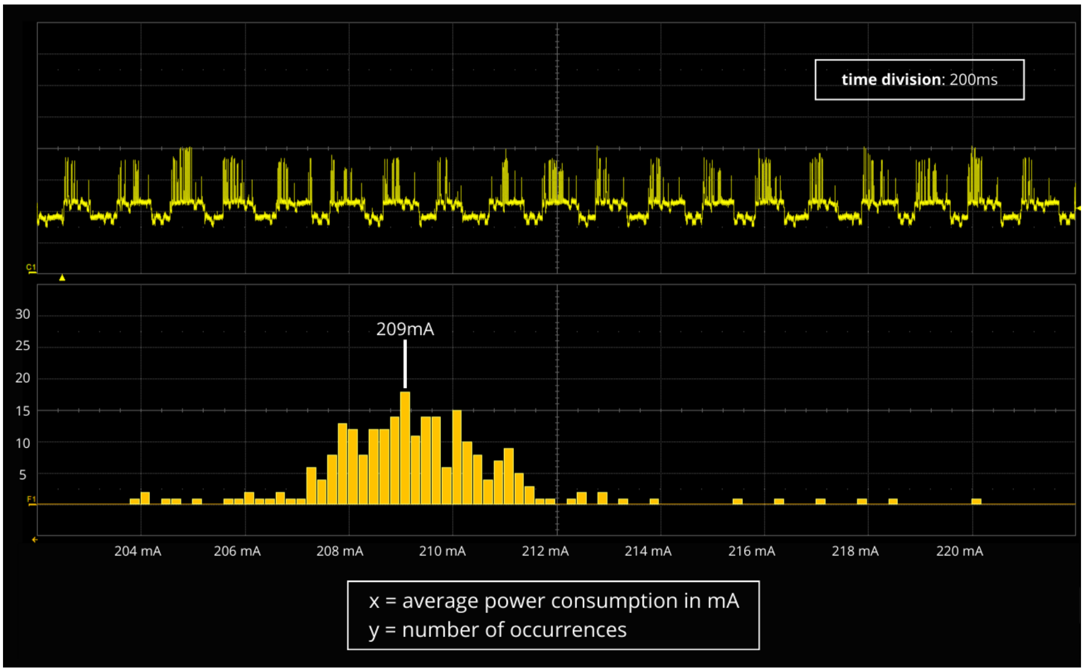


Figure 3.26: ESP32-S3-Korvo2 measurement of RTSP audio/video streaming event.

B event. The figure above is divided into two distinct graphs. The first graph depicts the energy consumption of RTSP audio/video streaming, with units of 200ms on the abscissa and 100mA on the ordinate. The second graph presents a collection of 250 average energy consumption measurements, each calculated over the time interval of the first graph. These measurements are then organised according to their frequency of occurrence, effectively calculating the mode.
The principal insight to be derived from this graph is the mean energy consumption of 209mA of RTSP streaming. Given an average streaming duration of 30 seconds:

$$\textbf{Electric charge} = 0.209\,\mathrm{A} \times 30000\,\mathrm{ms} = 6270\,\mathrm{mC} = 1.74\,\mathrm{mAh} \tag{3.3}$$

It can thus be posited that event B has an average power consumption of 1.74 mAh.

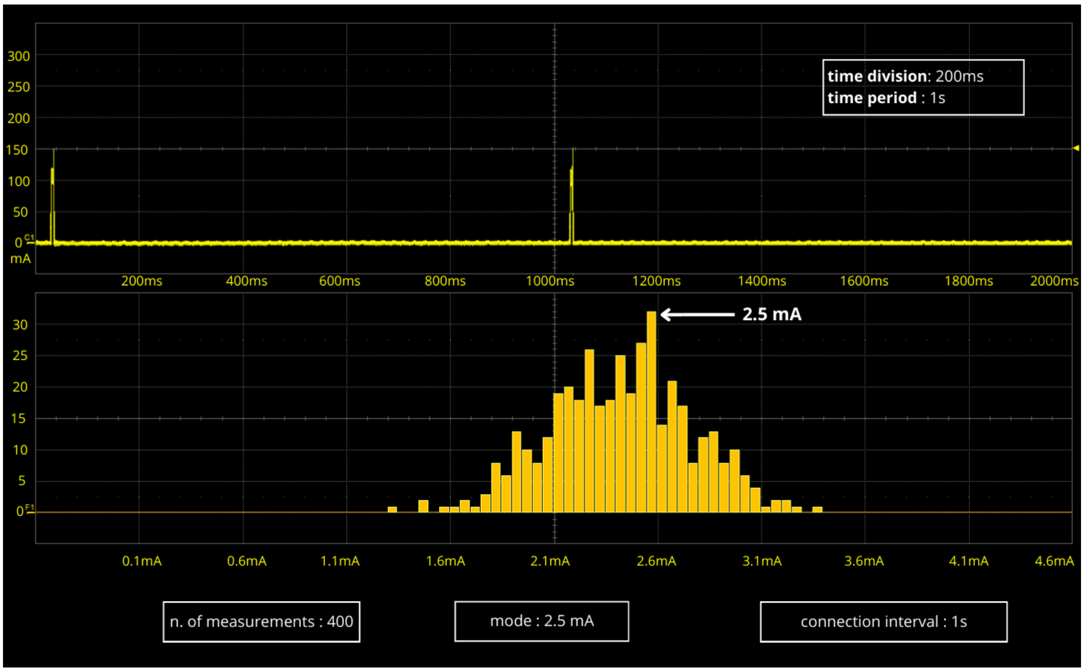


Figure 3.27: FireBeetle2 ESP32-E measurement of Automatic light sleep event.

C event. As with the preceding event, two distinct graphs can be identified in the figure above. The first graph represents the energy consumption of the Aytomatic light sleep mode. Specifically, two connection events can be identified, occurring one second apart (connection interval = 1s). The second graph collects 400 average energy consumption measurements, each calculated over the time interval of the first graph. These measurements are then organised according to their frequency of appearance, effectively calculating the mode.
The principal insight that can be derived from this graph is the average consumption of 2.5mA at rest.
Given that the device will remain at rest for the majority of the day, with the exception of events A and B, it can be calculated that the smart peephole will be in this mode for a total of 86,400 seconds (24 hours), 20 seconds (for five instances of event A) and 150 seconds (for five instances of event B). This equates to a total of 86,230 seconds.

$$\textbf{Electric charge} = 0.0025\,\text{A} \times 86230000\,\text{ms} = 215575\,\text{mC} = 59.8\,\text{mAh} \tag{3.4}$$

It can thus be posited that event C has a total (24h) power consumption of 59.8mAh.

The findings are presented in tabular form below.

Event	**Frequency**	**Duration (sec)**	**Energy consumption (Single event)**	**Energy consumption (Total)**
A	5	4	0.13 mAh	0.6 mAh
B	5	30	1.74 mAh	8.7 mAh
C	1	86230	59.8mAh	59.8mAh

Table 3.1: Daily energy consumption

In consideration of the presented data, an estimation of the daily energy consumption of the smart peephole can be made.

$$\textbf{Total daily consumption} = 0.6\,\text{mAh} + 8.7\,\text{mAh} + 59.8\,\text{mAh} = 69.1\,\text{mAh} \tag{3.5}$$

However in estimating the duration of the battery life before recharging is required, it is essential to consider not only the total daily energy consumption across various operational modes, but also several additional parameters. These include:

- the discharge safety margin;
- the battery's effective capacity;
- the configuration of the connection interval in automatic light sleep mode.

By incorporating these factors into the analysis, it is possible to make a more accurate determination of the device's practicality and longevity in real-world use.

Discharge Safety. The concept of 'discharge safety' is fundamental to the optimal management of batteries, particularly in the case of lithium batteries. The establishment of an appropriate 'discharge safety' value has the potential to significantly extend battery life by reducing the risk of damage due to deep discharge cycles [132]. This parameter, frequently expressed as depth of discharge (DoD), define the extent of energy that can be utilised prior to recharging the battery. A low DoD indicates that only a limited proportion of the battery's total capacity is utilised, which may be advantageous for battery longevity but could potentially restrict the device's autonomy. Conversely, a high DoD permits the utilisation of the majority of the battery's total capacity, although this may accelerate battery deterioration.

The graph (Fig. 3.28) below illustrates the impact of selecting an appropriate DoD on the life cycle of a Li-ion battery.

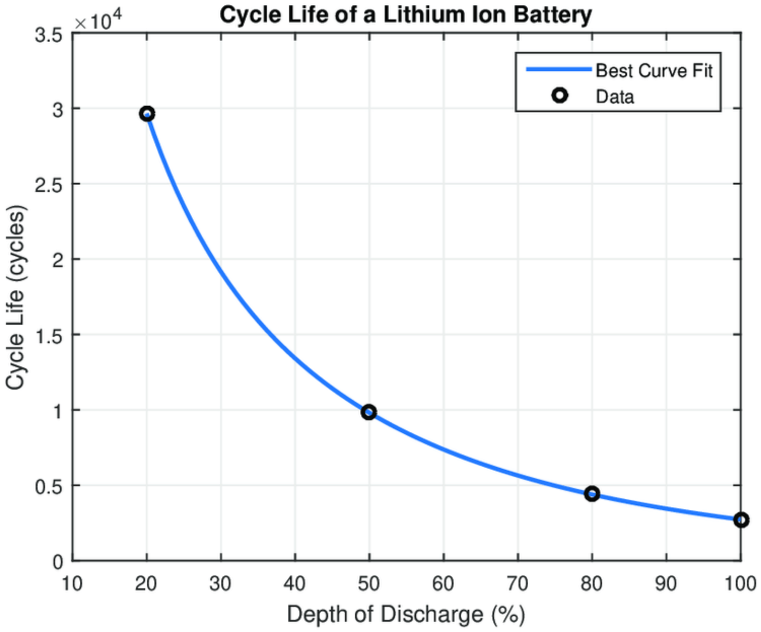


Figure 3.28: Depth of discharge versus cycle life of the lithium-ion battery.

Source [76].

In the case of lithium-ion batteries, the recommended DoD is 80%, which corresponds to a discharge safety value of 20% [103].

Battery capacity. In selecting a battery for the Smart Peephole, it is essential to strike a balance between the physical dimensions and the charging capacity. An illustrative example is the 6000mAh rechargeable lithium battery utilised in the Amazon Door View Cam [18], which offers a high level of autonomy with minimal power consumption, while maintaining a compact form factor. It is estimated that the Amazon Door View Cam will provide approximately two months of autonomy with this battery, which represents an excellent option for ensuring continuity of service without frequent interruptions.

Automatic light sleep configuration. The graph presented (Fig. 3.29) illustrates data collected during our experiments on the device's energy consumption on automatic light sleep mode, demonstrating the impact of varying connection intervals on power usage. By adjusting this interval, it is possible to observe changes in energy consumption and battery life.

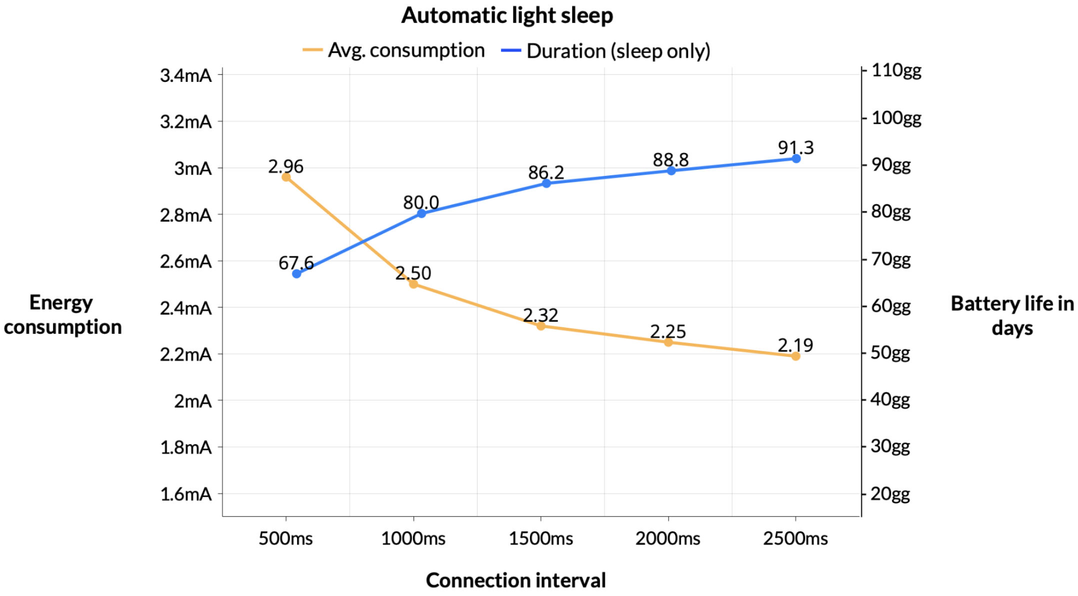


Figure 3.29: Connection interval impact on energy consumption.

The graph illustrates a clear correlation between an increase in the connection interval and a reduction in energy consumption, which in turn leads to an enhancement in battery life. This is due to the fact that longer intervals result in a reduction in the frequency of communication events, thereby saving power. In order to achieve an optimal balance between responsiveness and battery longevity, a connection interval of one second has been selected for the application in question. This configuration guarantees that the device will remain adequately responsive while optimising energy efficiency.

In light of the above considerations, it is now possible to make a more precise and realistic estimation of the device's battery life. In accordance with the specifications of a 6000mAh battery with a discharge safety of 20%, the battery life can be calculated as follows:

$$\textbf{Battery life} = \frac{6000 - (0.20 \times 6000)}{69.1} = \frac{4800}{69.1} \approx \textbf{70 days} \tag{3.6}$$

A battery life of 70 days represents a reasonable compromise for a smart peephole device, offering an extended period of use between recharges. This duration represents an optimal compromise between the convenience for the user and the device's power consumption, thereby minimising the need for frequent recharging while ensuring reliable, continuous operation.

Chapter 4

Conclusions

4.1 Retrospective

This thesis examined the challenges and solutions in the field of real-time communications and energy management for battery-powered connected devices, based on a low-cost microcontroller, with particular reference to the development of a smart peephole. From the beginning, the objective was to achieve a balance between the need for high-quality, real-time audio-video transmission and the energy efficiency required by battery-powered devices, with the aim of ensuring optimal performance without compromising integration with a broader technology ecosystem.

Smart home. The concept of smart home technologies was introduced as a rapidly expanding sector within the broader field of home security. A smart home necessitates the presence of devices that are not only capable of communicating and interacting with one another in real time, but are also energy efficient. This is due to the fact that a considerable number of these devices operate without a direct power connection and are therefore required to rely on batteries for extended periods. This work has responded to these issues by developing a tangible solution that addresses the needs of modern users who require security, reliability and ease of use.

Microcontroller. The selection of the ESP32 microcontroller proved to be a determining factor in the achievement of the project's success. This microcontroller is distinctive for its versatility and power, while offering wireless communication capabilities via Wi-Fi and Bluetooth, and a range of power-saving modes. The presence of these integrated technologies rendered the ESP32 the optimal solution for ensuring uninterrupted and seamless communication between the device and other components within the smart home ecosystem, while simultaneously minimising power consumption. The utilisation of communication protocols such as Bluetooth Low Energy (BLE) contributed to a reduction in power consumption when the device was in a state of idleness, while simultaneously ensuring the rapid reactivation and efficient transmission when required.

State of charge. As has been demonstrated, the longevity of the battery is a significant issue for battery-powered devices, particularly those that require constant, real-time communication. A detailed investigation was conducted into power management techniques, with a particular focus on the accurate estimation of the battery's state of charge (SoC). The accurate monitoring of the state of charge (SoC) is currently an open field of research with a variety of applications, from electric cars to wireless sensor networks, and is indispensable for ensuring that devices remain operational for as long as possible. The state-of-charge estimation method selected in this thesis, based on the use of the ampere-hour method enhanced with a composite correction factor, allows for a highly accurate estimation of the state of charge in comparison to traditional methods, with an estimated error below 2%. This not only optimise battery life, but also improve the overall reliability of the device, ensuring that the smart peephole would continue to function even under prolonged and intensive use. A significant aspect of this methodology is its capacity to enhance energy management without increasing the computational load on the device. This is a decisive factor in resource-limited devices, such as the smart peephole. Due to time constraints, it was not possible to implement this method in the final demo products. However, it has all the prerequisites to be the optimal choice for the project.

Real-time video streaming protocol. A further area of investigation for this project was the selection and implementation of real-time communication protocols for video transmission. In a security device such as the smart peephole, the rapid and faultless transmission of video and audio data enables the user to monitor the exterior of their residence at all times. The RTSP protocol was chosen for its capacity to efficiently manage the transmission of multimedia content, adapting to different network conditions and ensuring that video quality remained high even with limited bandwidth. This resulted in a smooth and seamless user experience, thus rendering the smart peephole a reliable home security solution. Furthermore, the integration of Wi-Fi with Bluetooth Low Energy enabled the device to intelligently alternate between the two communication modes, utilising BLE during periods of inactivity to minimise power consumption and switching to Wi-Fi solely for data transmission when necessary. This hybrid approach further enhanced the device's energy efficiency while maintaining high performance in data transmission.

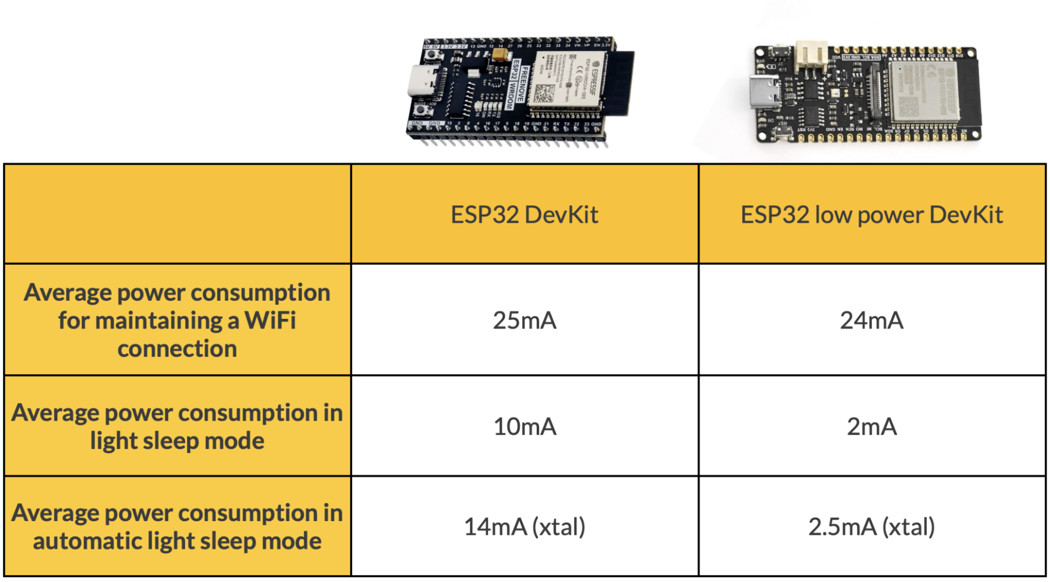
Case study. The development of the two main demo products represented the realisation of the theoretical and design solutions that had been elaborated in the preliminary phase of the project. The case study centred on the creation of two smart devices that would be incorporated into the existing Vimar ecosystem. The products developed, namely the "Cyclop" smart peephole and the audio device for two-way communication, serve to exemplify these objectives and enhance interaction with the home environment. The first product designated, Cyclop, was developed with the objective of providing a solution for the real-time transmission of video and audio data, in substitution of the door peephole. This was achieved through the utilisation of a Real-Time Streaming Protocol (RTSP), which facilitates a continuous and reliable flow of multimedia data. This proved to be of significant importance in guaranteeing that users are able to observe individuals outside their residences without any delays or disruptions, thereby providing a practical solution for real-time remote control. Furthermore, the ability to be integrated into an existing ecosystem has made Cyclop a competitive device in the smart home security solutions landscape. In addition to the development of Cyclop, a second product was developed: an audio device that has been specifically designed to offer two-way audio communication. This device, also based on the ESP32 platform, has been designed to ensure clear and immediate voice communication in real time. However, during the development of this device, some technical difficulties emerged, related to the integration of Espressif's Session Initiation Protocol (SIP) with the company's ecosystem. While this resulted in the cessation of product development, it highlighted some shortcomings of the ADF development kit. Specifically, the implementation of the SIP protocol was still in its early stages and not yet suitable for the production of a final product. The power management system of the Cyclop smart peephole ensures the operational efficiency of the device, as it is required to remain functional for extended periods without the need for frequent recharging. The development of power-saving strategies was oriented towards two main objectives: firstly, to optimise power consumption during periods of inactivity and, secondly, to enable a rapid response when the device was activated for real-time audio/video transmission. One of the initial steps was to conduct a rigorous examination of the device's hardware, with a particular focus on the components that directly influenced power consumption. It was discovered that the voltage regulator represented a potential point of inefficiency. To address this issue, the FireBeetle 2 ESP32-E, a development board purposely designed for low-power IoT applications, was employed. In terms of software optimisation, the option of utilising light sleep modes was investigated. By employing a technique known as automatic light sleep, it was feasible to sustain a persistent BLE connection with the video intercom, thereby guaranteeing minimal response latency and low power consumption. The integration of sophisticated sleep modes and the adoption of low-power components enabled the creation of a device that effectively strikes a balance between performance and autonomy, ensuring real-time transmission solely when necessary, without placing an excessive burden on battery life.

4.2 Future works

Following the development and validation of the demo products presented in this project, there are several potential areas for future improvement that could enhance the performance of the system and optimise its efficiency. These enhancements can be realised through the deployment of customised solutions, the formation of collaborative relationships with technology partners and the creation of dedicated hardware that is designed to align with the specific requirements of the device.

Development of a customised SIP protocol or collaboration with Espressif. One of the principal issues that manifested during the development of the audio device concerned the utilisation of the Session Initiation Protocol (SIP) implemented by Espressif. To address these issues, one potential solution is the development of a customised SIP protocol, optimised for specific company's requirements. An alternative option would be to collaborate with Espressif, the manufacturer of the ESP32 microcontroller, with a view to adapting or improving the existing communication protocol included in the Espressif Audio Development Framework (ESP-ADF). A collaboration with Espressif could result in a more complete solution than the current one, as well as facilitate integration with other devices within the company's smart home ecosystem.

Voltage regulator. A further evolutionary stage is the optimisation of the device's power consumption through the utilisation of a more efficient voltage regulator.

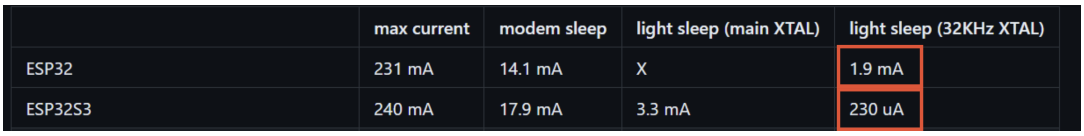


	ESP32 DevKit	ESP32 low power DevKit
Average power consumption for maintaining a WiFi connection	25mA	24mA
Average power consumption in light sleep mode	10mA	2mA
Average power consumption in automatic light sleep mode	14mA (xtal)	2.5mA (xtal)

Figure 4.1: Comparison of devkit with optimized vs. non-optimized voltage regulator.

During the development of the prototype, it became evident that the voltage regulator has a considerable impact on the overall power consumption, particularly during periods of device inactivity. This is clearly evident when the data collected above on two devkits using the same ESP32 microcontroller, but with two different voltage regulators, is compared (Fig. 4.1). The selection of a low-power voltage regulator, would result in a further reduction in consumption during light sleep mode.

Custom board. A final aspect of future work of the project involves the creation of a custom board based on the ESP32-S3, combined with an external quartz crystal and an optimised voltage regulator. Two distinct boards were employed to conduct the battery consumption tests: the Korvo2 for the audio/video streaming phase and the Firebeetle2 ESP32-E for the idle phase. In order to utilise the automatic light sleep mode, it was necessary to solder a quartz crystal on the Firebeetle2, to facilitate time scanning. According to estimates provided by Espressif (Fig. 4.2), in this mode, the S3 version of the ESP32, in combination with the quartz crystal, is more efficient than the basic ESP32 counterpart mounted on the Firebeetle that was tested.



	max current	modem sleep	light sleep (main XTAL)	light sleep (32KHz XTAL)
ESP32	231 mA	14.1 mA	X	1.9 mA
ESP32S3	240 mA	17.9 mA	3.3 mA	230 uA

Figure 4.2: Current consumption declared by Espressif in automatic light sleep mode with frequency management enabled.

Source [31].

The power consumption measured using the Firebeetle with ESP32, 2.5mA on average (see fig. 3.27), differs by a mere 600uA from that stated by Espressif, indicating that this data is a reliable representation of the actual power consumption of this microcontroller. In consideration of an inherent margin of uncertainty, an estimation of approximately 1/1.5mA, at worst, can be made with regard to the consumption in automatic light sleep mode with ESP32-S3 and external quartz crystal. The resulting values imply that the smart peephole would have a total battery life of **144/107 days**.

In conclusion, the future improvements represent the subsequent logical stages in the refinement of the performance of the developed devices, in particular the audio device and Cyclop. They will enable the continued development of the former and significantly enhance the battery life of the latter.

Appendix A

Needs and Requirements

A.1 Needs

Need ID	Description
ND01-010	The smart peephole must be built on a low-cost platform.
ND01-020	The smart peephole must integrate into the existing video intercom (VI) ecosystem.
ND01-030	The VI must display the peephole video when someone rings the doorbell.
ND01-040	It has to be possible for the VI to display the peephole video on demand.
ND01-050	The smart peephole shall be battery operated with an acceptable charging frequency.
ND01-060	The battery shall be capable of being conveniently recharged.
ND01-070	The smart peephole shall signal a transmission in progress.
ND01-080	The smart peephole must signal when the battery is low.

Table A.1: Needs group 1.

Need ID	Description
ND02-010	The audio device (AD) must be based on a low-cost platform.
ND02-020	The AD must integrate with the company's audio ecosystem.
ND02-030	The AD must be able to receive and make audio calls to and from a pre-configured device.
ND02-040	The audio stream must be bi-directional.
ND02-050	It must be possible to perform actions on the pre-configured device by pressing a button on the AD.

Table A.2: Needs group 2.

Need ID	Description
ND03-010	Whenever the smart peephole transmits the video stream, it must also provide audio communication.

Table A.3: Needs group 3.

A.2 Requirements

Requirement ID	Description	Need/s ID
REQ01-010	A comparative analysis of the low-cost boards with camera support on the market is necessary.	ND01-010
REQ01-020	The peephole will appear as a system camera.	ND01-020
REQ01-030	For video integration, a comparative analysis of existing video protocols supported by the existing ecosystem is necessary.	ND01-020
REQ01-040	For integration and communication with the video intercom (VI), an analysis is needed to understand whether and which communications will take place via Bluetooth and/or WiFi.	ND01-020
REQ01-050	The configuration required for integration (e.g. SSID and WiFi password) will have to be managed through a web page offered by a Web Server exposed by the smart peephole.	ND01-020
REQ01-060	The peephole will also be usable from a mobile phone, with the possibility of viewing the video stream, on demand, or when receiving a distributed out of door doorbell (ODD) notification (nice to have).	ND01-020
REQ01-070	When the bell button on the peephole body is pressed, the VI should display the video stream from the smart peephole, in a manner to be defined by analysis.	ND01-030
REQ01-080	When the OD button connected to the VI is pressed, the latter should show the video stream from the peephole, in the manner to be defined through an analysis.	ND01-030
REQ01-090	It should be possible to display the video stream of the smart peephole on demand, using the appropriate function keys in the VI interface.	ND01-040
REQ01-100	The configuration of the smart peephole camera must be managed via a web page offered by a web server exposed by the peephole itself	ND01-030, ND01-040
REQ01-110	The smart peephole should run on battery power and the charge should last for at least one month with an average use of 5 calls per day.	ND01-050
REQ01-120	An analysis of the device's consumption by adopting different technical solutions (BT/WiFi choice, deep-sleep mode, wake-up procedure, etc.) is necessary to quantify the actual recharging frequency required.	ND01-050
REQ01-130	The battery will have to be removable from the casing inside the port in order to be recharged.	ND01-060
REQ01-140	The battery should be able to be recharged via USB-C port.	ND01-060
REQ01-150	There should be an LED on the inside of the door that signals a transmission in progress.	ND01-070
REQ01-160	There should be an LED on the inside of the door to signal the low battery condition.	ND01-080
REQ01-170	The VI should show the battery status of the peephole on the screen, when in transmission with it.	ND01-080
REQ01-180	If equipped with an SD card, the peephole can save a frame of the video stream on it, whenever someone rings the doorbell.	

Table A.4: Requirements group 1.

Requirement ID	Description	Need/s ID
REQ02-010	A comparative analysis of the low-cost boards with audio support on the market is necessary.	ND02-010
REQ02-020	The audio device (AD) will integrate into the ecosystem via SIP protocol.	ND02-020
REQ02-030	The AD will implement a SIP client that will register to the SIP proxy of the pre-configured target device.	ND02-020
REQ02-040	The SIP configuration shall be modifiable through a web page offered by a Web Server exposed by the AD. The web interface shall allow the following fields to be edited: (see SPE02060).	ND02-020
REQ02-050	Integration with the company's centralised configuration system should be considered. (Further analysis)	ND02-020
REQ02-060	The SIP ID of the target device to be contacted can be configured via a web page offered by a webserver exposed by the AD.	ND02-030
REQ02-070	The PoC will gloss over audio reentry issues (which would require the introduction of echo cancellers, noise suppressors and other devices).	ND02-040
REQ02-080	The AD will have to enable the activation of specific functions on the target device (at the push of a button) by sending a SIP message.	ND02-050

Table A.5: Requirements group 2.

Requirement ID	Description	Need/s ID
REQ03-010	A comparative analysis of low-cost boards with audio and camera support on the market is required. Please refer to the analyses in requirements REQ01-010 and REQ02-010.	ND01-010, ND03-010
REQ03-020	The smart peephole will appear as system camera, communicating via SIP protocol.	ND01-020, ND03-010
REQ03-030	For integration and communication with video intercom (VI), an analysis is needed to understand whether and which communications will take place via Bluetooth and/or WiFi.	ND03-020
REQ03-040	The AD will implement a SIP client that will register to the SIP proxy of the pre-configured target device.	ND01-020, ND03-010
REQ03-050	The SIP configuration shall be modifiable through a web page offered by a Web Server exposed by the smart peephole. The web interface shall allow the following fields to be edited: (see SPE0).	ND01-020, ND03-010
REQ03-060	The peephole will also be usable from a mobile phone, with the possibility of viewing the video stream, on demand, or when receiving a distributed out of door doorbell notification (nice to have).	ND01-020

Table A.6: Requirements group 3 (Part 1).

Requirement ID	Description	Need/s ID
REQ03-070	When the bell button on the peephole body is pressed, the VI should display the audio/video stream from the smart peephole, in a manner to be defined by analysis.	ND01-030, ND03-010
REQ03-080	When the OD button connected to the VI is pressed, the latter should show the audio/video stream from the peephole, in the manner to be defined through an analysis.	ND01-030, ND03-010
REQ03-090	It should be possible to display the video stream of the smart peephole on demand, using the appropriate function keys in the VI interface.	ND01-030, ND03-010
REQ03-100	The configuration of the smart peephole camera must be managed via a web page offered by a web server exposed by the peephole itself	ND01-030, ND01-040, ND03-010
REQ03-110	The smart peephole should run on battery power and the charge should last for at least one month with an average use of 5 calls per day.	ND01-050, ND03-010
REQ03-120	An analysis of the device's consumption by adopting different technical solutions (BT/WiFi choice, deep-sleep mode, wake-up procedure, etc.) is necessary to quantify the actual recharging frequency required.	ND01-050, ND03-010
REQ03-130	The battery will have to be removable from the casing inside the port in order to be recharged.	ND01-060
REQ03-140	The battery should be able to be recharged via USB-C port.	ND01-060
REQ03-150	There should be an LED on the inside of the door that signals a transmission in progress.	ND01-070
REQ03-160	There should be an LED on the inside of the door to signal the low battery condition.	ND01-080
REQ03-170	The VI should show the battery status of the peephole on the screen, when in transmission with it.	ND01-080
REQ03-180	If equipped with an SD card, the peephole can save a frame of the video stream on it, whenever someone rings the doorbell.	

Table A.7: Requirements group 3 (Part 2).

Bibliography

Articles

[1] J. . et al. “A real-time automated port entry and exit management framework for fishing vessels using yolov8”. In: *International Journal of Latest Engineering Research and Applications (IJLERA)* 9 (3 2024), pp. 32–36. DOI: `10.56581/ijlera.9.3.32-36` (cit. on p. 31).

[2] *ADF block diagram source.* URL: `https://docs.espressif.com/projects/esp-adf/en/latest/about.html` (cit. on p. 40).

[3] *ADF RTSP library.* URL: `https://github.com/espressif/esp-adf-libs/blob/master/esp_media_protocols/include/esp_rtsp.h` (cit. on p. 44).

[4] *ADF SIP library.* URL: `https://github.com/espressif/esp-adf/tree/master/examples/protocols/voip` (cit. on p. 43).

[5] M. Aguayo et al. “Dash adaptation algorithm based on adaptive forgetting factor estimation”. In: *IEEE Transactions on Multimedia* 20 (5 2018), pp. 1224–1232. DOI: `10.1109/tmm.2017.2764325` (cit. on p. 31).

[6] Farid Uddin Ahmed et al. “A Brief Overview of On-Chip Voltage Regulation in High-Performance and High-Density Integrated Circuits”. In: *IEEE Access* PP (Dec. 2020). DOI: `10.1109/ACCESS.2020.3047347` (cit. on p. 49).

[7] G. Ali and S. Lee. “Multipath dynamic adaptive streaming over http using scalable video coding in software defined networking”. In: *Applied Sciences* 10 (21 2020), p. 7691. DOI: `10.3390/app10217691` (cit. on p. 31).

[8] *Amazon Ring source.* URL: `https://it-it.ring.com/products/door-view-cam-75mm` (cit. on p. 6).

[9] *AMS1117 Datasheet.* URL: `http://www.advanced-monolithic.com/pdf/ds1117.pdf` (cit. on p. 50).

[10] *AMS1117 source.* URL: `https://www.amazon.in/AMS1117-3-3V-Voltage-Regulator-SOT-223/dp/B08VF159MH` (cit. on p. 50).

[11] H. An, W. Park, and S. Park. “Wireless spy camera spotter system with real-time traffic similarity analysis and wifi signal tracing”. In: *IEEE Access* 12 (2024), pp. 4459–4470. DOI: `10.1109/access.2024.3350175` (cit. on p. 31).

[12] Anania Aron, Gabriel Girban, and Stefan Kilyeni. “A geometric approach of a battery mathematical model for on-line energy monitoring”. In: *2011 IEEE EUROCON - International Conference on Computer as a Tool.* 2011, pp. 1–4. DOI: 10.1109/EUROCON.2011.5929286 (cit. on p. 26).

[13] Uniza Asad and Summia Taj. “Interoperability in IOT based smart home: A review”. In: *Review of Computer Engineering Studies* VOL 5 (Sept. 2018), pp. 50–55. DOI: 10.18280/rces.050302.

[14] Furqan Asghar et al. “Simulation Study on Battery State of Charge Estimation Using Kalman Filter”. In: *Journal of Advanced Computational Intelligence and Intelligent Informatics* 20 (Nov. 2016), pp. 861–866. DOI: 10.20965/jaciii.2016.p0861 (cit. on p. 28).

[15] M. Bacco et al. “Performance analysis of webrtc-based video streaming over power constrained platforms”. In: *2018 IEEE Global Communications Conference (GLOBECOM)* (2018). DOI: 10.1109/glocom.2018.8647375 (cit. on p. 31).

[16] U. Banerjee et al. “Eedtls: energy-efficient datagram transport layer security for the internet of things”. In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference* (2017). DOI: 10.1109/glocom.2017.8255053 (cit. on p. 31).

[17] J. P. R. Barrera, N. Muñóz-Galeano, and H. Maldonado. “Soc estimation for lithium-ion batteries: review and future challenges”. In: *Electronics* 6 (4 2017), p. 102. DOI: 10.3390/electronics6040102 (cit. on p. 25).

[18] *Battery Video Doorbell specifications.* URL: https://it-it.ring.com/products/battery-doorbell (cit. on p. 69).

[19] N. Bhardwaj and K. T. Joseph. “Real-time video streaming applications in military 5g networks: a technical review”. In: *International Journal for Research in Applied Science and Engineering Technology* 12 (5 2024), pp. 3100–3103. DOI: 10.22214/ijraset.2024.62313 (cit. on p. 31).

[20] Md Multan Biswas et al. “Towards Implementation of Smart Grid: An Updated Review on Electrical Energy Storage Systems”. In: *Smart Grid and Renewable Energy* 4 (Feb. 2013), pp. 122–132. DOI: 10.4236/sgre.2013.41015 (cit. on p. 24).

[21] *BLE stack image source.* URL: https://medium.com/@thomsmed/core-bluetooth-and-the-ble-stack-95ddc29c8671 (cit. on p. 57).

[22] *Bluetooth Core specification 5.0.* URL: https://www.mouser.com/pdfdocs/bluetooth-Core-v50.pdf (cit. on pp. 57, 59).

[23] *Bluetooth low energy.* URL: https://www.bluetooth.com/bluetooth-technology/radio-versions (cit. on p. 57).

[24] Vincent Boly et al. “From descriptive customer data to need definition: a formalised approach”. In: *International Journal of Product Development* 21 (2016), p. 369. DOI: 10.1504/IJPD.2016.10004476 (cit. on p. 35).

[25] L. Buttyán and A. Grilo. "A secure distributed transport protocol for wireless sensor networks". In: *2011 IEEE International Conference on Communications (ICC)* (2011). DOI: 10.1109/icc.2011.5962646 (cit. on p. 32).

[26] Gilles Callebaut et al. "The Art of Designing Remote IoT Devices—Technologies and Strategies for a Long Battery Life". In: *Sensors* 21 (Jan. 2021), p. 913. DOI: 10.3390/s21030913 (cit. on p. 4).

[27] *Camera/MicroSD issue.* URL: https://esp32.com/viewtopic.php?f=13&t=38388&p=128119&hilit=adf+camera+microsd#p128119 (cit. on p. 46).

[28] W. E. Castellanos, J. C. Guerri, and P. Arce. "A qos-aware routing protocol with adaptive feedback scheme for video streaming for mobile networks". In: *Computer Communications* 77 (2016), pp. 10–25. DOI: 10.1016/j.comcom.2015.08.012 (cit. on p. 30).

[29] Y. Chen et al. "A time-efficient and accurate open circuit voltage estimation method for lithium-ion batteries". In: *Energies* 12 (9 2019), p. 1803. DOI: 10.3390/en12091803 (cit. on p. 27).

[30] Ze Cheng et al. "Estimation of State of Charge for Lithium-Ion Battery Based on Finite Difference Extended Kalman Filter". In: *Journal of Applied Mathematics* 2014 (Apr. 2014). DOI: 10.1155/2014/348537 (cit. on p. 26).

[31] *Delcared ESP32 current consumption with management enabled.* URL: https://github.com/espressif/esp-idf/tree/master/examples/bluetooth/nimble/power_save#typical-current-consumption-with-management-enabled (cit. on pp. 62, 75).

[32] Luca Di Nunzio et al. "Energy Consumption Saving in Embedded Microprocessors Using Hardware Accelerators". In: *TELKOMNIKA Indonesian Journal of Electrical Engineering* 16 (June 2018). DOI: 10.12928/telkomnika.v16i3.9387 (cit. on p. 13).

[33] P. Dias et al. "Optimized printed cathode electrodes for high performance batteries". In: *Energy Technology* 9 (1 2020). DOI: 10.1002/ente.202000805 (cit. on p. 24).

[34] G. Diraco et al. "Review on human action recognition in smart living: sensing technology, multimodality, real-time processing, interoperability, and resource-constrained processing". In: *Sensors* 23 (11 2023), p. 5281. DOI: 10.3390/s23115281.

[35] K. Du et al. "Server-driven video streaming for deep learning inference". In: *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies,* (2020). DOI: 10.1145/3387514.3405887 (cit. on p. 30).

[36] J. Eckhardt, A. Vogelsang, and D. Méndez. "Are "non-functional" requirements really non-functional?" In: *Proceedings of the 38th International Conference on Software Engineering* (2016). DOI: 10.1145/2884781.2884788 (cit. on p. 36).

[37] Albert Einstein, Boris Podolsky, and Nathan Rosen. "Can Quantum-Mechanical Description of Physical Reality be Considered Complete?" In: *Physical Review* 47.10 (1935), pp. 777–780. DOI: `10.1103/PhysRev.47.777`.

[38] *ESP32 Datasheet*. URL: `https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf` (cit. on pp. 50–52).

[39] *ESP32 LyraT Board*. URL: `https://espressif-docs.readthedocs-hosted.com/projects/esp-adf/en/latest/design-guide/dev-boards/get-started-esp32-lyrat.html` (cit. on p. 43).

[40] *ESP32-S3 Korvo-2*. URL: `https://espressif-docs.readthedocs-hosted.com/projects/esp-adf/en/latest/design-guide/dev-boards/user-guide-esp32-s3-korvo-2.html` (cit. on p. 44).

[41] *Espressif IDF*. URL: `https://www.espressif.com/en/products/sdks/esp-idf` (cit. on p. 40).

[42] *Espressif SIP API*. URL: `https://github.com/espressif/esp-adf-libs/blob/master/esp_media_protocols/include/esp_rtc.h` (cit. on p. 46).

[43] *EZVIZ CP4 image source*. URL: `https://www.amazon.it/EZVIZ-Spioncino-Campanello-Rilevamento-Bidirezionale/dp/B09PDFWFJJ` (cit. on p. 7).

[44] *EZVIZ CP4 specifications*. URL: `https://www.ezviz.com/it/product/CP4/58063#Spec` (cit. on p. 7).

[45] *FireBeetle 2 ESP32-E Datasheet*. URL: `https://wiki.dfrobot.com/FireBeetle_Board_ESP32_E_SKU_DFR0654` (cit. on p. 50).

[46] Matthias Fré et al. "Scalable MDC-Based Volumetric Video Delivery for Real-Time One-to-Many WebRTC Conferencing". In: Apr. 2024, pp. 121–131. DOI: `10.1145/3625468.3647617` (cit. on p. 31).

[47] *Freenove ESP32-WROVER Board*. URL: `https://store.freenove.com/products/fnk0060` (cit. on p. 42).

[48] J. Gao and H. D. Abruña. "Key parameters governing the energy density of rechargeable li/s batteries". In: *The Journal of Physical Chemistry Letters* 5 (5 2014), pp. 882–885. DOI: `10.1021/jz5001819` (cit. on p. 23).

[49] M. R. Ghori, T. Wan, and G. C. Sodhy. "Bluetooth low energy mesh networks: survey of communication and security protocols". In: *Sensors* 20 (12 2020), p. 3590. DOI: `10.3390/s20123590` (cit. on p. 57).

[50] Sonika Krishna Gogineni, Theresa Riedelsheimer, and Rainer Stark. "Systematic product development methodology for customizable IoT devices". In: *Procedia CIRP* (2019), pp. 393–399. DOI: `https://doi.org/10.1016/j.procir.2019.04.287`. URL: `https://www.sciencedirect.com/science/article/pii/S2212827119309357` (cit. on p. 34).

[51] Carles Gomez, J. Oller, and J. Aspas. “Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology”. In: *Sensors (Basel, Switzerland)* 12 (2012), pp. 11734–11753. DOI: 10.3390/s120911734 (cit. on p. 56).

[52] Yılmaz Güven et al. “Understanding the Concept of Microcontroller Based Systems To Choose The Best Hardware For Applications”. In: *Research Inventy: International Journal of Engineering And Science* 7 (Dec. 2017), p. 38 (cit. on p. 15).

[53] Maryam Hailan, Baraa Albaker, and Muwafaq Alwan. “Transformation to a smart factory using NodeMCU with Blynk platform”. In: *Indonesian Journal of Electrical Engineering and Computer Science* Vol 30 (Apr. 2023), pp. 237–245. DOI: 10.11591/ijeecs.v30.i1.pp237-245 (cit. on p. 13).

[54] Irsyad Nashirul Haq et al. “Development of battery management system for cell monitoring and protection”. In: *2014 International Conference on Electrical Engineering and Computer Science (ICEECS)*. 2014, pp. 203–208. DOI: 10.1109/ICEECS.2014.7045246 (cit. on p. 26).

[55] W. Hu, Z. Wang, and L. Sun. “Characterizing tcp performance for chunk delivery in dash”. In: *Lecture Notes in Computer Science* (2016), pp. 509–519. DOI: 10.1007/978-3-319-48890-5_50 (cit. on p. 31).

[56] Z. Huang et al. “Convolutional gated recurrent unit–recurrent neural network for state-of-charge estimation of lithium-ion batteries”. In: *IEEE Access* 7 (2019), pp. 93139–93149. DOI: 10.1109/access.2019.2928037 (cit. on p. 27).

[57] *IDF image source*. URL: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/about.html (cit. on p. 39).

[58] Y. Irawan et al. “Design of library noise detection tools based on voice pressure parameters”. In: *ILKOM Jurnal Ilmiah* 14 (3 2022), pp. 237–244. DOI: 10.33096/ilkom.v14i3.1191.237-244 (cit. on p. 13).

[59] Hadi Jameel Hadi and Husam Kareem. “Low-Cost Energy-Efficient Smart Monitoring System Using Open-Source Microcontrollers”. In: *International Review of Automatic Control* 9 (Nov. 2016), pp. 423–428. DOI: 10.15866/ireaco.v9i6.10315 (cit. on p. 13).

[61] T. Kaarlela et al. “Towards metaverse: utilizing extended reality and digital twins to control robotic systems”. In: *Actuators* 12 (6 2023), p. 219. DOI: 10.3390/act12060219 (cit. on p. 31).

[62] A. Kaknjo et al. “Real-time video latency measurement between a robot and its remote control station: causes and mitigation”. In: *Wireless Communications and Mobile Computing* 2018 (1 2018). DOI: 10.1155/2018/8638019 (cit. on p. 30).

[63] S. B. Kamatchi et al. “Enhancing microcomputer edge computing for autonomous iot motion control”. In: *International Journal on Recent and Innovation Trends in Computing and Communication* 11 (3 2023), pp. 58–67. DOI: 10.17762/ijritcc.v11i3.6202 (cit. on p. 13).

[64] M. Khan. "Innovations in battery technology: enabling the revolution in electric vehicles and energy storage". In: *British Journal of Multidisciplinary and Advanced Studies* 5 (1 2024), pp. 23–41. DOI: 10.37745/bjmas.2022.0414 (cit. on p. 24).

[65] Dahoo Kim et al. "Reducing Power and Energy Consumption of Nonvolatile Microcontrollers with Transparent On-Chip Instruction Cache". In: *Circuits and Systems* 5 (Oct. 2014), pp. 253–364. DOI: 10.4236/cs.2014.511027 (cit. on p. 13).

[66] Do-Hyun Kim et al. "Efficient Management of Fast Charging Systems Based on a Real-Time Monitoring System". In: *Electronics* 11 (Feb. 2022), p. 520. DOI: 10.3390/electronics11040520 (cit. on p. 26).

[67] M. Kim et al. "Facile fabrication of solution-processed solid-electrolytes for high-energy-density all-solid-state-batteries by enhanced interfacial contact". In: *Scientific Reports* 10 (1 2020). DOI: 10.1038/s41598-020-68885-4.

[68] P. Kumar et al. "Lightweight and secure session-key establishment scheme in smart home environments". In: *IEEE Sensors Journal* 16 (1 2016), pp. 254–264. DOI: 10.1109/jsen.2015.2475298.

[69] Douglas Lautner et al. "Power Efficient Scheduling Algorithms for Real-time Tasks on Multi-mode Microcontrollers". In: *Procedia Computer Science* 130 (Jan. 2018), pp. 557–566. DOI: 10.1016/j.procs.2018.04.099 (cit. on p. 14).

[70] In Lee. "The Internet of Things for enterprises: An ecosystem, architecture, and IoT service business model". In: *Internet Things* 7 (2019). DOI: 10.1016/J.IOT.2019.100078 (cit. on p. 4).

[71] J. Li et al. "An adaptively biased ldo regulator with 11na quiescent current and 50ma available load". In: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)* (2021). DOI: 10.1109/iscas51556.2021.9401687 (cit. on p. 49).

[72] S. Li et al. "Fast charging anode materials for lithium-ion batteries: current status and perspectives". In: *Advanced Functional Materials* 32 (23 2022). DOI: 10.1002/adfm.202200796 (cit. on p. 23).

[73] Xinyue Liu et al. "Advances in the Study of Techniques to Determine the Lithium-Ion Battery's State of Charge". In: *Energies* 17.7 (2024). ISSN: 1996-1073. DOI: 10.3390/en17071643. URL: https://www.mdpi.com/1996-1073/17/7/1643 (cit. on p. 26).

[74] Y. Liu et al. "State of charge estimation for li-ion batteries based on an unscented h-infinity filter". In: *Journal of Electrical Engineering Amp; Technology* 15 (6 2020), pp. 2529–2538. DOI: 10.1007/s42835-020-00544-0 (cit. on p. 27).

[75] X. Luo et al. "A lightweight privacy-preserving communication protocol for heterogeneous iot environment". In: *IEEE Access* 8 (2020), pp. 67192–67204. DOI: 10.1109/access.2020.2978525.

[76] Kevin Mallon, Francis Assadian, and Bo Fu. "Analysis of On-Board Photovoltaics for a Battery Electric Bus and Their Impact on Battery Lifespan". In: *Energies* 10 (July 2017), p. 943. DOI: 10.3390/en10070943 (cit. on p. 69).

[77] Premkumar Manoharan et al. "SoC Estimation and Monitoring of Li-ion Cell using Kalman-Filter Algorithm". In: *International Journal on Electrical Engineering and Informatics* 6 (Dec. 2018), pp. 418–427. DOI: 10.11591/ijeei.v6i1.548 (cit. on p. 26).

[78] S. Mare et al. "Consumer smart homes". In: *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications* (2019). DOI: 10.1145/3301293.3302371.

[80] Mbadfar, Ratna Babu Chinnam, and Murat Yildirim. "Enhancing Lithium-Ion Battery State-of-Charge Estimation Across Battery Types via Unsupervised Domain Adaptation". In: *Proceedings of the European Conference of the Prognostics and Health Management Society (PHME)* (2024). DOI: 10.36001/phme.2024.v8i1.4012 (cit. on p. 26).

[81] *Micro-RTSP github*. URL: https://github.com/brainrecall/Micro-RTSP/tree/master/examples/esp-idf (cit. on p. 42).

[82] "Microprocessor design using copras method". In: *3* 2 (3 2024), pp. 22–34. DOI: 10.46632/jeae/2/3/3 (cit. on p. 13).

[83] Konstantin Mikhaylov and Jouni Tervonen. "Evaluation of Power Efficiency for Digital Serial Interfaces of Microcontrollers". In: May 2012. DOI: 10.1109/NTMS.2012.6208716 (cit. on p. 14).

[84] K. C. Ndeche and S. O. Ezeonu. "Implementation of coulomb counting method for estimating the state of charge of lithium-ion battery". In: *Physical Science International Journal* (2021), pp. 1–8. DOI: 10.9734/psij/2021/v25i330244 (cit. on p. 25).

[85] T. Nguyen et al. "From lab on a chip to point of care devices: the role of open source microcontrollers". In: *Micromachines* 9 (8 2018), p. 403. DOI: 10.3390/mi9080403 (cit. on p. 15).

[86] J. Oh et al. "A secure and lightweight authentication protocol for iot-based smart homes". In: *Sensors* 21 (4 2021), p. 1488. DOI: 10.3390/s21041488.

[87] K. Ohno et al. "Security mechanism for dash agent framework". In: *2016 IEEE 5th Global Conference on Consumer Electronics* (2016). DOI: 10.1109/gcce.2016.7800411 (cit. on p. 31).

[88] J. Oikarinen et al. "Quality-of-service management in asema system for unicast mpeg4 video transmissions". In: *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO 2007)* (2007). DOI: 10.1109/euromicro.2007.51 (cit. on p. 31).

[89] *Osservatorio internet of Things*. URL: https://www.osservatori.net/it/ricerche/comunicati-stampa/smart-home-mercato-italiano (cit. on pp. 4, 5).

[90] P. Papadimitriou, V. Tsaoussidis, and C. Zhang. “End-to-end loss differentiation for video streaming with wireless link errors”. In: *Telecommunication Systems* 43 (3-4 2009), pp. 295–312. DOI: 10.1007/s11235-009-9203-8 (cit. on p. 30).

[91] Iyyanar Perumal, M. Chitra, and P. Sabarinath. “Effective and Secure Scheme for Video Streaming Using SRTP”. In: *International Journal of Machine Learning and Computing* (Jan. 2012), pp. 855–859. DOI: 10.7763/IJMLC.2012.V2.252.

[92] D. R et al. “Anti theft security system”. In: *International Journal of Innovative Research in Information Security* 9 (02 2023), pp. 15–20. DOI: 10.26562/ijiris.2023.v0902.03 (cit. on p. 19).

[93] O. Rezaei, R. Habibifar, and Z. Wang. “A robust kalman filter-based approach for soc estimation of lithium-ion batteries in smart homes”. In: *Energies* 15 (10 2022), p. 3768. DOI: 10.3390/en15103768 (cit. on p. 24).

[94] G. Ross et al. “Best practices and current implementation of emerging smartphone-based (bio)sensors – part 1: data handling and ethics”. In: *TrAC Trends in Analytical Chemistry* 158 (2023), p. 116863. DOI: 10.1016/j.trac.2022.116863 (cit. on p. 37).

[95] *RT9080 Datasheet.* URL: https://www.richtek.com/assets/product_file/RT9080/DS9080-08.pdf.

[96] H.S. Ryu et al. “Self-discharge characteristics of lithium/sulfur batteries using TEGDME liquid electrolyte”. In: *Electrochimica Acta* 52.4 (2006). ELECTROCHEMICAL ENERGY CONVERSION AND STORAGE - ISE MEETING 2005, pp. 1563–1566. ISSN: 0013-4686. DOI: https://doi.org/10.1016/j.electacta.2006.01.086. URL: https://www.sciencedirect.com/science/article/pii/S001346860600466X (cit. on p. 23).

[97] Chandra S et al. “IoT-Powered UPS Battery Monitoring: Ensuring High availability and reliability for Critical Systems”. In: *E3S Web of Conferences* 399 (July 2023). DOI: 10.1051/e3sconf/202339904007 (cit. on p. 26).

[98] S. S. Saha, S. S. Sandha, and M. Srivastava. “Machine learning for microcontroller-class hardware: a review”. In: *IEEE Sensors Journal* 22 (22 2022), pp. 21362–21390. DOI: 10.1109/jsen.2022.3210773 (cit. on p. 15).

[99] S. H. Sakdiyah, N. Eltivia, and A. Afandi. “Root cause analysis using fishbone diagram: company management decision making”. In: *Journal of Applied Business, Taxation and Economics Research* 1 (6 2022), pp. 566–576. DOI: 10.54408/jabter.v1i6.103 (cit. on p. 3).

[100] W. A. Salah and B. A. Zneid. “Evolution of microcontroller-based remote monitoring system applications”. In: *International Journal of Electrical and Computer Engineering (IJECE)* 9 (4 2019), p. 2354. DOI: 10.11591/ijece.v9i4.pp2354-2364.

[101] *SDK definition by Amazon.* URL: `https://aws.amazon.com/it/what-is/sdk/` (cit. on p. 38).

[102] M. A. Al-Shareeda et al. “Bluetooth low energy for internet of things: review, challenges, and open issues”. In: *Indonesian Journal of Electrical Engineering and Computer Science* 31 (2 2023), p. 1182. DOI: `10.11591/ijeecs.v31.i2.pp1182-1189` (cit. on p. 56).

[103] J. Shen, S. Düşmez, and A. Khaligh. “An advanced electro-thermal cycle-lifetime estimation model for lifepolt;infgt;4lt;/infgt; batteries”. In: *2013 IEEE Transportation Electrification Conference and Expo (ITEC)* (2013). DOI: `10.1109/itec.2013.6574494` (cit. on p. 69).

[104] M. Simić et al. “A novel method for in-situ extracting bio-impedance model parameters optimized for embedded hardware”. In: *Scientific Reports* 13 (1 2023). DOI: `10.1038/s41598-023-31860-w` (cit. on p. 19).

[105] *Smart Home statista.* URL: `https://www.statista.com/outlook/cmo/smart-home/italy` (cit. on pp. 4, 5).

[106] J. Song and H. Meng. “Research and teaching practice on experiment course of stm32-embedded microcontroller”. In: *Proceedings of the 2019 3rd International Conference on Education, Economics and Management Research (ICEEMR 2019)* (2019). DOI: `10.2991/assehr.k.191221.081` (cit. on p. 20).

[107] *SPIFFS documentation.* URL: `https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/storage/spiffs.html` (cit. on p. 45).

[108] Riza Sulaiman. “The process of quality assurance under open source software development”. In: *ISCI 2011 - 2011 IEEE Symposium on Computers and Informatics* (Mar. 2011). DOI: `10.1109/ISCI.2011.5958975` (cit. on p. 18).

[109] A. K. Sultania, C. Delgado, and J. Famaey. “Enabling low-latency bluetooth low energy on energy harvesting batteryless devices using wake-up radios”. In: *Sensors* 20 (18 2020), p. 5196. DOI: `10.3390/s20185196` (cit. on p. 57).

[110] J. D. Summers, S. Joshi, and B. Morkos. “Requirements evolution: relating functional and non-functional requirement change on student project success”. In: *Volume 3: 16th International Conference on Advanced Vehicle Technologies; 11th International Conference on Design Education; 7t* (2014). DOI: `10.1115/detc2014-35023` (cit. on p. 36).

[111] A. Swart. “Analyzing the application of two main microcontrollers in engineering education – a case study of three ieee conferences focusing on education”. In: *Advances in Science, Technology and Engineering Systems Journal* 6 (3 2021), pp. 339–346. DOI: `10.25046/aj060339` (cit. on p. 14).

[112] Y. W. Syaifudin et al. “Study of performance of real time streaming protocol (rtsp) in learning systems”. In: *International Journal of Engineering Amp; Technology* 7 (4.44 2018), p. 216. DOI: 10.14419/ijet.v7i4.44.26994 (cit. on p. 31).

[113] Madhavan Thothadri. “An Analysis on Clock Speeds in Raspberry Pi Pico and Arduino Uno Microcontrollers”. In: *American Journal of Engineering and Technology Management* 6 (June 2021), pp. 41–46. DOI: 10.11648/j.ajetm.20210603.13 (cit. on p. 17).

[114] I. Udousoro. “Effective requirement engineering process model in software engineering”. In: *Software Engineering* 8 (1 2020), p. 1. DOI: 10.11648/j.se.20200801.11.

[115] J. Uhomoibhi, C. Onime, and H. Wang. “A study of developments and applications of mixed reality cubicles and their impact on learning”. In: *The International Journal of Information and Learning Technology* 37 (1-2 2019), pp. 15–31. DOI: 10.1108/ijilt-02-2019-0026.

[116] C. Vidal et al. “Machine learning applied to electrified vehicle battery state of charge and state of health estimation: state-of-the-art”. In: *IEEE Access* 8 (2020), pp. 52796–52814. DOI: 10.1109/access.2020.2980961 (cit. on p. 27).

[118] H. Wang et al. “Novel estimation solution on lithium-ion battery state of charge with current-free detection algorithm”. In: *IET Circuits, Devices Amp; Systems* 13 (2 2019), pp. 245–249. DOI: 10.1049/iet-cds.2018.5406 (cit. on p. 25).

[119] P. Wang, F. Ye, and X. Chen. “A smart home gateway platform for data collection and awareness”. In: *IEEE Communications Magazine* 56 (9 2018), pp. 87–93. DOI: 10.1109/mcom.2018.1701217 (cit. on p. 1).

[120] X. Wang, C. Gao, and M. Sun. “Probabilistic prediction algorithm for cycle life of energy storage in lithium battery”. In: *World Electric Vehicle Journal* 10 (1 2019), p. 7. DOI: 10.3390/wevj10010007 (cit. on p. 23).

[121] Feng Xia et al. “Control-theoretic dynamic voltage scaling for embedded controllers”. In: *IET Comput. Digit. Tech.* 2 (2008), pp. 377–385. URL: https://api.semanticscholar.org/CorpusID:16233802 (cit. on p. 14).

[122] *Xiaomi Smart Doorbell 3 image source*. URL: https://www.amazon.it/Xiaomi-Risoluzione-Riconoscimento-Segnalatore-Bidirezionale/dp/B0B6F9W6KM (cit. on p. 7).

[123] *Xiaomi Smart Doorbell 3 specifications*. URL: https://www.mi.com/it/product/xiaomi-smart-doorbell-3/ (cit. on p. 8).

[124] J. Xie et al. “State of charge estimation of lithium-ion battery based on extended kalman filter algorithm”. In: *Frontiers in Energy Research* 11 (2023). DOI: 10.3389/fenrg.2023.1180881 (cit. on p. 27).

[125] Xin Xiong et al. "A novel practical state of charge estimation method: an adaptive improved ampere-hour method based on composite correction factor". In: *International Journal of Energy Research* 44 (Aug. 2020). DOI: `10.1002/er.5758` (cit. on pp. 27–29).

[126] B. Yang, Y. Wang, and H. Gao. "State-of-charge estimation of lithium-ion batteries based on pso-bp neural network". In: *International Journal of Energy and Power Engineering* 10 (6 2021), p. 115. DOI: `10.11648/j.ijepe.20211006.13` (cit. on p. 27).

[127] Y. Yu and S. Lee. "Remote driving control with real-time video streaming over wireless networks: design and evaluation". In: *IEEE Access* 10 (2022), pp. 64920–64932. DOI: `10.1109/access.2022.3183758` (cit. on p. 30).

[128] A. Zakhil et al. "Vehicular flow characterization: an internet of video things-based solution". In: *Pakistan Journal of Engineering and Technology* 6 (1 2023), pp. 13–22. DOI: `10.51846/vol6iss1pp13-22`.

[129] L. Zhang et al. "A sparse learning machine for real-time soc estimation of li-ion batteries". In: *IEEE Access* 8 (2020), pp. 156165–156176. DOI: `10.1109/access.2020.3017774` (cit. on p. 27).

[130] Y. Zhang, S. Wang, and W. Xu. "An improved smoothing factor-extended kalman filtering method for accurate online state-of-charge estimation of lithium-ion battery". In: *Journal of Physics: Conference Series* 2232 (1 2022), p. 012011. DOI: `10.1088/1742-6596/2232/1/012011` (cit. on p. 27).

[131] F. Zhao, Y. Guo, and B. Chen. "A review of lithium-ion battery state of charge estimation methods based on machine learning". In: *World Electric Vehicle Journal* 15 (4 2024), p. 131. DOI: `10.3390/wevj15040131` (cit. on p. 27).

[132] C. Zhou et al. "Modeling of the cost of ev battery wear due to v2g application in power systems". In: *IEEE Transactions on Energy Conversion* 26 (4 2011), pp. 1041–1050. DOI: `10.1109/tec.2011.2159977` (cit. on p. 68).

Books

[2] *ADF block diagram source.* URL: `https://docs.espressif.com/projects/esp-adf/en/latest/about.html` (cit. on p. 40).

[3] *ADF RTSP library.* URL: `https://github.com/espressif/esp-adf-libs/blob/master/esp_media_protocols/include/esp_rtsp.h` (cit. on p. 44).

[4] *ADF SIP library.* URL: `https://github.com/espressif/esp-adf/tree/master/examples/protocols/voip` (cit. on p. 43).

[8] *Amazon Ring source.* URL: `https://it-it.ring.com/products/door-view-cam-75mm` (cit. on p. 6).

[9] *AMS1117 Datasheet.* URL: http://www.advanced-monolithic.com/pdf/ds1117.pdf (cit. on p. 50).

[10] *AMS1117 source.* URL: https://www.amazon.in/AMS1117-3-3V-Voltage-Regulator-SOT-223/dp/B08VF159MH (cit. on p. 50).

[18] *Battery Video Doorbell specifications.* URL: https://it-it.ring.com/products/battery-doorbell (cit. on p. 69).

[21] *BLE stack image source.* URL: https://medium.com/@thomsmed/core-bluetooth-and-the-ble-stack-95ddc29c8671 (cit. on p. 57).

[22] *Bluetooth Core specification 5.0.* URL: https://www.mouser.com/pdfdocs/bluetooth-Core-v50.pdf (cit. on pp. 57, 59).

[23] *Bluetooth low energy.* URL: https://www.bluetooth.com/bluetooth-technology/radio-versions (cit. on p. 57).

[27] *Camera/MicroSD issue.* URL: https://esp32.com/viewtopic.php?f=13&t=38388&p=128119&hilit=adf+camera+microsd#p128119 (cit. on p. 46).

[31] *Delcared ESP32 current consumption with management enabled.* URL: https://github.com/espressif/esp-idf/tree/master/examples/bluetooth/nimble/power_save#typical-current-consumption-with-management-enabled (cit. on pp. 62, 75).

[38] *ESP32 Datasheet.* URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (cit. on pp. 50–52).

[39] *ESP32 LyraT Board.* URL: https://espressif-docs.readthedocs-hosted.com/projects/esp-adf/en/latest/design-guide/dev-boards/get-started-esp32-lyrat.html (cit. on p. 43).

[40] *ESP32-S3 Korvo-2.* URL: https://espressif-docs.readthedocs-hosted.com/projects/esp-adf/en/latest/design-guide/dev-boards/user-guide-esp32-s3-korvo-2.html (cit. on p. 44).

[41] *Espressif IDF.* URL: https://www.espressif.com/en/products/sdks/esp-idf (cit. on p. 40).

[42] *Espressif SIP API.* URL: https://github.com/espressif/esp-adf-libs/blob/master/esp_media_protocols/include/esp_rtc.h (cit. on p. 46).

[43] *EZVIZ CP4 image source.* URL: https://www.amazon.it/EZVIZ-Spioncino-Campanello-Rilevamento-Bidirezionale/dp/B09PDFWFJJ (cit. on p. 7).

[44] *EZVIZ CP4 specifications.* URL: https://www.ezviz.com/it/product/CP4/58063#Spec (cit. on p. 7).

[45] *FireBeetle 2 ESP32-E Datasheet.* URL: https://wiki.dfrobot.com/FireBeetle_Board_ESP32_E_SKU_DFR0654 (cit. on p. 50).

[47] *Freenove ESP32-WROVER Board.* URL: https://store.freenove.com/products/fnk0060 (cit. on p. 42).

[57] *IDF image source.* URL: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/about.html (cit. on p. 39).

[60] Daniel T. Jones James P. Womack. *Lean Thinking, Second Editon.* Simon & Schuster, Inc., 2010.

[79] Surendra Martha and Sadananda Muduli. *Thermodynamic Studies on Energy Density of Batteries.* Jan. 2021. ISBN: 9780124095489. DOI: 10.1016/B978-0-12-819723-3.00136-0 (cit. on p. 24).

[81] *Micro-RTSP github.* URL: https://github.com/brainrecall/Micro-RTSP/tree/master/examples/esp-idf (cit. on p. 42).

[89] *Osservatorio internet of Things.* URL: https://www.osservatori.net/it/ricerche/comunicati-stampa/smart-home-mercato-italiano (cit. on pp. 4, 5).

[95] *RT9080 Datasheet.* URL: https://www.richtek.com/assets/product_file/RT9080/DS9080-08.pdf.

[101] *SDK definition by Amazon.* URL: https://aws.amazon.com/it/what-is/sdk/ (cit. on p. 38).

[105] *Smart Home statista.* URL: https://www.statista.com/outlook/cmo/smart-home/italy (cit. on pp. 4, 5).

[107] *SPIFFS documentation.* URL: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/storage/spiffs.html (cit. on p. 45).

[117] W. Waag and D.U. Sauer. *SECONDARY BATTERIES – LEAD– ACID SYSTEMS | State-of-Charge/Health.* Ed. by Jürgen Garche. Amsterdam: Elsevier, 2009, pp. 793–804. ISBN: 978-0-444-52745-5. DOI: https://doi.org/10.1016/B978-044452745-5.00149-0. URL: https://www.sciencedirect.com/science/article/pii/B9780444527455001490 (cit. on p. 25).

[122] *Xiaomi Smart Doorbell 3 image source.* URL: https://www.amazon.it/Xiaomi-Risoluzione-Riconoscimento-Segnalatore-Bidirezionale/dp/B0B6F9W6KM (cit. on p. 7).

[123] *Xiaomi Smart Doorbell 3 specifications.* URL: https://www.mi.com/it/product/xiaomi-smart-doorbell-3/ (cit. on p. 8).

Websites

[2] *ADF block diagram source.* URL: https://docs.espressif.com/projects/esp-adf/en/latest/about.html (cit. on p. 40).

[3] *ADF RTSP library.* URL: https://github.com/espressif/esp-adf-libs/blob/master/esp_media_protocols/include/esp_rtsp.h (cit. on p. 44).

[4] *ADF SIP library.* URL: https://github.com/espressif/esp-adf/tree/master/examples/protocols/voip (cit. on p. 43).

[8] *Amazon Ring source.* URL: https://it-it.ring.com/products/door-view-cam-75mm (cit. on p. 6).

[9] *AMS1117 Datasheet.* URL: http://www.advanced-monolithic.com/pdf/ds1117.pdf (cit. on p. 50).

[10] *AMS1117 source.* URL: https://www.amazon.in/AMS1117-3-3V-Voltage-Regulator-SOT-223/dp/B08VF159MH (cit. on p. 50).

[18] *Battery Video Doorbell specifications.* URL: https://it-it.ring.com/products/battery-doorbell (cit. on p. 69).

[21] *BLE stack image source.* URL: https://medium.com/@thomsmed/core-bluetooth-and-the-ble-stack-95ddc29c8671 (cit. on p. 57).

[22] *Bluetooth Core specification 5.0.* URL: https://www.mouser.com/pdfdocs/bluetooth-Core-v50.pdf (cit. on pp. 57, 59).

[23] *Bluetooth low energy.* URL: https://www.bluetooth.com/bluetooth-technology/radio-versions (cit. on p. 57).

[27] *Camera/MicroSD issue.* URL: https://esp32.com/viewtopic.php?f=13&t=38388&p=128119&hilit=adf+camera+microsd#p128119 (cit. on p. 46).

[31] *Delcared ESP32 current consumption with management enabled.* URL: https://github.com/espressif/esp-idf/tree/master/examples/bluetooth/nimble/power_save#typical-current-consumption-with-management-enabled (cit. on pp. 62, 75).

[38] *ESP32 Datasheet.* URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (cit. on pp. 50–52).

[39] *ESP32 LyraT Board.* URL: https://espressif-docs.readthedocs-hosted.com/projects/esp-adf/en/latest/design-guide/dev-boards/get-started-esp32-lyrat.html (cit. on p. 43).

[40] *ESP32-S3 Korvo-2.* URL: https://espressif-docs.readthedocs-hosted.com/projects/esp-adf/en/latest/design-guide/dev-boards/user-guide-esp32-s3-korvo-2.html (cit. on p. 44).

[41] *Espressif IDF.* URL: https://www.espressif.com/en/products/sdks/esp-idf (cit. on p. 40).

[42] *Espressif SIP API.* URL: https://github.com/espressif/esp-adf-libs/blob/master/esp_media_protocols/include/esp_rtc.h (cit. on p. 46).

[43] *EZVIZ CP4 image source.* URL: https://www.amazon.it/EZVIZ-Spioncino-Campanello-Rilevamento-Bidirezionale/dp/B09PDFWFJJ (cit. on p. 7).

[44] *EZVIZ CP4 specifications.* URL: https://www.ezviz.com/it/product/CP4/58063#Spec (cit. on p. 7).

[45] *FireBeetle 2 ESP32-E Datasheet.* URL: https://wiki.dfrobot.com/FireBeetle_Board_ESP32_E_SKU_DFR0654 (cit. on p. 50).

[47] *Freenove ESP32-WROVER Board.* URL: https://store.freenove.com/products/fnk0060 (cit. on p. 42).

[57] *IDF image source.* URL: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/about.html (cit. on p. 39).

[81] *Micro-RTSP github.* URL: https://github.com/brainrecall/Micro-RTSP/tree/master/examples/esp-idf (cit. on p. 42).

[89] *Osservatorio internet of Things.* URL: https://www.osservatori.net/it/ricerche/comunicati-stampa/smart-home-mercato-italiano (cit. on pp. 4, 5).

[95] *RT9080 Datasheet.* URL: https://www.richtek.com/assets/product_file/RT9080/DS9080-08.pdf.

[101] *SDK definition by Amazon.* URL: https://aws.amazon.com/it/what-is/sdk/ (cit. on p. 38).

[105] *Smart Home statista.* URL: https://www.statista.com/outlook/cmo/smart-home/italy (cit. on pp. 4, 5).

[107] *SPIFFS documentation.* URL: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/storage/spiffs.html (cit. on p. 45).

[122] *Xiaomi Smart Doorbell 3 image source.* URL: https://www.amazon.it/Xiaomi-Risoluzione-Riconoscimento-Segnalatore-Bidirezionale/dp/B0B6F9W6KM (cit. on p. 7).

[123] *Xiaomi Smart Doorbell 3 specifications.* URL: https://www.mi.com/it/product/xiaomi-smart-doorbell-3/ (cit. on p. 8).