POLITECNICO
MILANO 1863

ENHANCING EMBEDDED TECHNOLOGY DESIGN PROCESSES
An overview and proposal

Tesi di Laurea Magistrale in Interaction Design

Studente:
MARCELLO AMMENDOLIA

Docente:
Prof. PAOLO PEREGO

ANNO ACCADEMICO 2022-2023

INTRODUCTION

In the realm of embedded technologies, the balance between sophisticated toolsets and user experience has long been a delicate dance. While the field has witnessed remarkable technological strides, the journey for those navigating design platforms has remained, at times, convoluted. This thesis was birthed from an observation that, despite the advancements in embedded technologies, the designers and engeneers process journey was often riddled with complexities that detracted from the core design objectives.

Embedded technologies, by their very nature, are intricate and demand a level of precision in their design. However, the traditional design journey often complicates this further, entailing extensive research, consultations, and an overwhelming plethora of delicate choices to make. The inherent challenge and our driving question became clear: how could we revolutionize this journey to be both user-centric and attuned to the specific nuances of designing embedded technologies?

With this ambition, our research embarked on the quest to develop an enabling platform that transcends typical design boundaries. Our aim was to create an environment where users could intuitively design, personalize, and visualize their embedded technology projects, all within a unified and integrated platform. This would not just be another tool, it would be a holistic experience tailored to the unique requirements and aspirations of those diving deep into the world of embedded technologies.

This thesis offers a deep dive into our developmental narrative, tracing our steps from the early realization of existing gaps to the culmination in our platform for developing embedded technologies projects faster and easier. Along this journey, we grapple with challenges, celebrate milestones, and maintain an unwavering commitment to elevating the user experience in the domain of embedded technology design.

CHAPTER 0
THESIS METHODOLOGY:

This thesis is structured into 4 main sections: Research (A), Concept (B), Elaborated Interface (C), and Development Process (D). In the research section, an initial attempt is made to provide a brief informative overview about what embedded technologies are, their diverse range of applications, and the market that characterizes them (A0). The first chapter (A1) delves into different approaches that can be used in the field of Embedded Technology design, observing their pros and cons. It ranges from the commonly used component-based approach to the newer declarative and AI-based methods.

Next, the discussion shifts to the challenges that define this kind of design, making it difficult. However, these challenges arise from the context and are not inherent to the design process itself (A2). The fourth research chapter provides an overview of hardware solutions that have emerged in the last decade to assist users in these processes. It critically examines the pros and limitations of using boards like the popular Arduino and many similar ones (A3).

The fifth chapter focuses on software solutions commonly employed by designers and engineers to achieve their goals in such projects. It analyzes various software products available in the market for users of all expertise levels, concentrating on each software's strengths and weaknesses and comparing them to provide a comprehensive view (A4).

In the concept section (B), the core principles upon which the entire solution is based will be illustrated, outlining the key vision and direction taken in the design and development process.

In the elaborated interface section (C), the appearance and functionality of the proposed interface are detailed. It begins with the presentation of the platform, offering a preview of how the interface appears to users. The characteristics of the interface (C1) delve into key functionalities and design decisions. The final interface pages (C2) provide a comprehensive picture of how users will see and interact with the interface.

In the development process section (D), the introduction (D1) sets the context and goals. Chapter two delves deeper into user analysis. In this thesis, we present personas and User Journey Maps to fully understand user needs, their pain points, the emotions they typically experience when engaging with this kind of work, differences arising from varied backgrounds, and many other insightful reflections that represent the initial evaluation phases with users, pinpointing challenges and opportunities. (D2). The development of the first information architecture (D3) describes the initial structuring of information and interactions, with details on user flows and tree tests. During the wireframe development phase (D4), various iterations of the interface are showcased, highlighting the changes and adjustments made. The final tests (D5) represent the concluding phase of interface verification, encompassing task-based tests, a questionnaire, and heatmap analysis.

The thesis concludes with reflections on the development process and on the work in general (E), summarizing lessons learned and future prospects.

SECTION A
RESEARCH PHASE

SECTION A – CHAPTER 1
OVERVIEW OF EMBEDDED TECHNOLOGIES PRODUCT DESIGN

Embedded systems have become ubiquitous in our daily lives, appearing in everything from smartphones and household appliances to industrial machines and healthcare devices. As miniaturized computing units designed to perform specific functions or tasks within a larger system, they are the invisible workhorses powering the smart, connected world we live in today.[1] The rapid advancements in semiconductor technology, coupled with the decreasing cost of electronic components, have fueled an exponential growth in the deployment of embedded systems. They are no longer confined to niche applications but have proliferated across various sectors, affecting how we live, work, and interact with technology. [2]. However, this prevalence is not without its challenges. The rise in complexity, performance requirements, and functionalities demands a multi-disciplinary approach for design and development. It necessitates the amalgamation of software engineering, electronic circuit design knowledge, user interface development, and even mechanical engineering, among other disciplines. In today's interconnected world, the proliferation of embedded technologies has dramatically transformed the landscape of product design, making it far more complex yet endlessly more capable. [3] This evolution has permeated the very fabric of our lives, encapsulating objects and making them become "smart." Take for instance the home appliances we rely on daily. A washing machine from decades ago was essentially a mechanical product,

---

[1] Tripathi, S. L. and Dwivedi, S. (2022) Electronic Devices and Circuit Design. 1st edn. Apple Academic Press.
[2] Rafiquzzaman M. Microcontroller Theory and Applications with the PIC18F (1st. ed.), Wiley Publishing (2011)
[3] Güven Y., Coşgun E., Kocaoğlu S., Gezici H., Yilmazlar E.Understanding the concept of microcontroller based systems to choose the best hardware for applications Res. Inventy Int. J. Eng. Sci., 7 (38) (2017)

designed primarily with gears and motors in mind. Contrast this with today's smart washing machines, brimming with sensors and sophisticated timing algorithms. These modern appliances are no longer mere machines but connected entities, capable of being controlled remotely and adaptable to a variety of washing conditions.

Similarly, the humble lighting fixture has undergone a luminous transformation. Once an object purely of aesthetic and functional design, modern lighting has evolved into intelligent systems. [4] With capabilities ranging from remote control to dynamic brightness and color adjustments, the contemporary light fixture is a hive of embedded technologies that require expertise well beyond traditional design principles.

Wearables present another compelling example. What began as simple devices for tracking physical activity have now evolved into highly sophisticated pieces of technology that can monitor a variety of health metrics, make payments, and even provide real-time translation services. [5] The expertise required to develop wearables has expanded correspondingly, encompassing everything from biomechanics to data security.

In the exploration of embedded device development across various sectors, Raad's (2020) "Fundamentals of IoT and Wearable Technology Design" delineates a broad spectrum of applications. [6] The realms of health care and fitness are seen to substantially benefit from these advancements, offering enhanced monitoring and analytical capabilities. Similarly, the sports sector is leveraging embedded technology to augment performance analytics and foster a deeper understanding of biomechanics. In the realm of entertainment and gaming, embedded devices enhance user experience through interactive and immersive technologies. The domain of pet care is also evolving, with embedded devices aiding in monitoring and ensuring the

[4] Kiruthika, J. and Dhandapani, A., 2016. Making Washing Machines Smart through IoT. International Journal of Modern Trends in Engineering and Science, 3, pp.39-41

[5] Farion, C. (2022) The Ultimate Guide to Informed Wearable Technology. 1st edn. Packt Publishing.

[6] Raad, H. (2020) Fundamentals of IoT and Wearable Technology Design. 1st edn. Wiley.

well-being of animals. Public safety is another critical area where embedded technology plays a pivotal role, facilitating better monitoring and emergency response mechanisms. The travel and tourism sectors are employing embedded devices to streamline operations and enrich the user experience, whether through navigational aids or interactive information systems. Aerospace is a sector where the stakes are high, and the integration of embedded systems is instrumental in enhancing safety, navigation, and communication. The educational landscape is being reshaped with the infusion of embedded technologies that foster interactive learning and access to resources. Lastly, the fashion industry is at the cusp of a revolution with the advent of wearable technology, blending aesthetics with functional value, reflecting a seamless amalgamation of style and substance. Through these myriad applications, the pervasive impact of embedded device technology across diverse sectors underscores a transformative trajectory, propelling each domain towards heightened efficacy and user engagement.

This seismic shift in the realm of design and engineering is not merely an academic or technical curiosity; it has considerable economic implications as well. According to SNS Insider Research, The Embedded Systems Market size was valued at US$ 97.35 Bn in 2022, and is Projected to reach US$ 159.9 Bn by 2030, with growing healthy CAGR of 6.4% Over the Forecast Period 2023-2030. [7] This even considering the impact of Covid19, War in Ukraine and Global Recession. [8]

[7] Global Market Insights, 2023. Embedded System Market Share | Global Report, 2023-2032. Available at: www.gminsights.com

[8] ResearchGate, 2023. Embedded Systems Market Size, Share & Growth Report 2023. Available at: www.researchgate.net

SECTION A – CHAPTER 2
AN ANALYSIS OF PROCESSES AND METHODOLOGIES FOR DEVELOPING EMBEDDED PRODUCTS

2.1 Introduction to processes and methodologies

Let's take a look at the approaches adopted by designers and engineers when creating a design product containing embedded components. In this chapter, the main hypothetical approaches are discussed, from the most common one, which is the waterfall approach based on component choice, to other hybrid approaches shared with other branches of engineering, such as Computer Science.

It should be noted that the choice of approach is often influenced by one's background. Engineers, who often have more technical training and a deeper understanding of the components and their functions, might find it easier and more intuitive to use a system that assumes significant expertise in the field of components. [9] They can quickly select appropriate components and understand how to integrate them into a larger system. On the other hand, designers, who may be more focused on user experience, aesthetics, or general functionalities rather than technical details, might find the "component-based" approach less straightforward. For them, the need to know specific components might represent a barrier to entry or a distraction from their main goal, which is to create a product or system that meets certain functional or aesthetic requirements. In this context, design tools more oriented towards "system" or "workflow", which allow focusing on functional objectives without worrying too much about technical details, might be more suitable for designers.

[9] Zhang, J. (2018) An Ebd Approach To Embedded Product Design. 1st edn. LAP LAMBERT Academic Publishing.

2.2 The Logistical and Financial Implications of Embedded Designs

The challenges of working with embedded designs start from the prototyping phase and continue throughout the product development cycle. Up to now, bottleneck management research has concentrated on manufacturing processes, while neglecting product design and engineering processes.[10] There are multiple steps in the workflow, each fraught with its own challenges:

1) The first step in circuit design is not designing circuits. The first step is to establish, identify, and harmonize the specifications and standards the design is expected to satisfy. This is a nontrivial exercise because it is not unusual to find some requirements in stark conflict with others. Resolution of such conflicts is itself a high practice of the engineering arts. [11] Pries, K. and Quigley, J, talk about K characteristics of the product, that must be firstly identified and addressed because the rest of the system will be really dependent on it. [12]
2) Component Selection: The next step involves selecting the appropriate components based on technical specifications, availability, and cost. With many options available, this process can be time-consuming and requires a deep understanding of component functionalities and compatibilities. This step is usually done using the "component approach" but also other methodologies can be applied, as we will see further on.
2) Sourcing: Once chosen, the components must be obtained from suppliers, often dealing with multiple vendors, minimum order quantities, and extended lead times.
3) Shipping and Waiting: After purchase, there's the inevitable wait for components to arrive. Delays can occur for various reasons such as customs, shipping mistakes, or backorders.

---

[10] Johannes Hinckeldeyn, Rob Dekkers, Nils Altfeld, Jochen Kreutzfeldt, Expanding bottleneck management from manufacturing to product design and engineering processes, Computers & Industrial Engineering, Volume 76, 2014, Pages 415-428, ISSN 0360-8352

[11] Fowler, K. (2014) Developing and Managing Embedded Systems and Products. Elsevier Science. Chapter 10 - Electronic Design

[12] Pries, K. and Quigley, J. (2008) Project Management of Complex and Embedded Systems. 1st edn. CRC Press.

4) Assembly: After arrival, the delicate assembly process starts. Given the intricate nature of electronic components, specialized equipment and expertise are essential.
5) Testing: After assembly, each part is tested under various conditions to guarantee functionality and dependability. Any issues at this stage might necessitate revisiting previous steps, further lengthening the development timeline.
6) Iterative Process: The complex nature of electronics design often requires multiple iterations, each demanding a repetition of the previously mentioned steps. Design Reviews Well-constructed and recurring reviews can have a positive influence on the design outcome. [13] When many eyes and brains review a project and the design details, the likelihood is that design errors reveal themselves. When a group of engineers with varied experience meet and critique the design, they uncover software or hardware problems of design. When the team practices design reviews rigorously, it can eliminate or reduce expensive testing by removing problems promptly.
While customer specifications usually require formal design reviews, they often lack specific direction and discipline in the design review requirement, resulting in an unstructured review process that fails to fulfill either of the following two main purposes of design review, which are to gain new knowledge and challenge satisfaction.[14]

Any modification or error in projects involving embedded components can significantly increase both financial and time costs. When a change occurs — whether to a component or a PCB layout — its effects ripple through the entire project. Inefficient manufacturing can lead to higher production costs, poor workflows, and ultimately a lot of scrapped parts or rework. A worst-case scenario is product recalls or high levels of warranty work. In today's competitive environment, efficiency in manufacturing is required. The new or adjusted design requires extensive testing for compatibility and another round of functional tests. [15] This not only extends the project timeline but also incurs additional costs. Delays can lead to

[13] Greenfield, Adam (2006). Everyware: The Dawning Age of Ubiquitous Computing. New Riders.
[14] Pries, K. and Quigley, J. (2008) Project Management of Complex and Embedded Systems. 1st edn. CRC Press.
[15] Tennant, D. (2022) Product Development. 1st edn. Wiley.

missed market opportunities and budget overruns. Added shipping costs for new component orders, especially from international suppliers, bring further delays and potential customs or import duties. Given the time-sensitive nature of tech innovations, being first to market can offer significant advantages.

2.3 Component Based Design Model

Component-based design is an approach that focuses on the use of predefined and modular components to build a system. This model is most commonly used in electrical engineering, software engineering, and other engineering fields. Although widely used, component-based design can become a logistical bottleneck, especially when dealing with a large number of different components. The need to maintain a detailed inventory, ensure compatibility between components, and manage versions and updates can become a significant burden. This is particularly true in large-scale projects or environments where rapid development is crucial. [16]

As already mentioned, Engineers, who often have more technical training and a deeper understanding of the components and their functions, might find it easier and more intuitive to use a system like this, thanks to which they can quickly select appropriate components and understand how to integrate them into a larger system. [17]

Tools that utilize this model are the famous and popular Thinkercad, KiCad, Altium Designer, Eagle, and so on. Without adequate knowledge, designers might choose components that aren't optimal for their needs, leading to inefficiencies, malfunctions, and in extreme cases, project failure. This method can also create barriers to innovation; if engineers or designers aren't updated on the latest components or available technologies, they might end up using outdated solutions that limit the project's innovative potential. Moreover, once the components have been selected and integrated, the design becomes rigid and less adaptable to changes

[16] Greenfield, Adam (2006). Everyware: The Dawning Age of Ubiquitous Computing. New Riders.

[17] Zhang, J. (2018) An Ebd Approach To Embedded Product Design. 1st edn. LAP LAMBERT Academic Publishing.

in specifications or needs, as making modifications can be laborious.[18] Using predefined components can also introduce interdependencies that make the system harder to modify or update in the future. However, there are also clear advantages. Using predefined components that are known to work well together can speed up the prototyping process, avoiding the need to design every part of the system from scratch. Another advantage is reuse: components can be reused across different projects, leading to greater efficiency and consistency across various systems. Lastly, the modular nature of component-based design allows for individual parts to be tested independently, facilitating the verification and validation of the overall system.

2.4 Dichiarative or Functional Model

The functional or declarative model is a design approach that focuses on goals or desired outcomes, rather than on implementation details. The functional or declarative model represents an emerging paradigm in the design of integrated circuits and hardware systems in general. Traditionally associated with functional programming languages and domains such as simulation and optimization, this model is also gaining ground in the field of electrical engineering. [19]

The reason for this shift is partly due to the advent of artificial intelligence (AI) technologies, which are revolutionizing the way we approach the design and optimization of complex systems. In this scenario, neural network classification, as delineated by Liu and Law (2021) in "Artificial Intelligence Hardware Design", becomes a pivotal technique. [20] AI acts as a catalyst, allowing designers to specify what they want a system to do, rather than how it does it. Machine learning and optimization algorithms can then explore the space of possible implementations to find the most efficient solution that meets functional requirements. This is particularly useful in scenarios where the system's complexity makes it difficult, if

[18] Skibińska, J., Saafi, S., Pascacio, P., ... & Lohan, E. S. (2021). A survey onwearable technology: History, state-of- the-art, and current challenges. Computer Networks, 193, 108074.
[19] Page, T. (2010) A Methodology for Decision-Support in Electronic Product Design. 1st edn. LAP LAMBERT Academic Publishing..
[20] Liu, A. C.-C. and Law, O. M. K. (2021) Artificial Intelligence Hardware Design. 1st edn. Wiley.

not impossible, for a human to correctly choose among thousands of available components, wasting time and resources.

Therefore analyzing this model, as pros we see how the functional model offers a high level of abstraction, allowing designers to focus on the functional objectives of the circuit rather than on the details at the transistor or logic gate level. This makes it easier to reason about the system's behavior. Advanced automation is another advantage, with AI capable of handling complexities that would be time-consuming or challenging for humans, such as the selection and optimization of components. Logic functions or modules can be easily reused and combined to create more complex functionalities, enhancing design efficiency. An optimization-based slotting approach to automated Integrated Circuit (IC) design is proposed in a parametric programming context. A paper discusses parametric design systems as a generative tool in architectural design, emphasizing the algorithmic basis of parametric tools which could be analogous to a layered model with parametric design at each layer. [21] This approach may hint at a layered or modular design approach, where different parameters or layers could be optimized independently [22] Testing becomes simpler since it's easier to test and verify individual functions, each designed for a specific and well-defined task. Maintenance and system updates may also be more straightforward, as changes to a specific function are less likely to impact other parts of the system. Some design tools using a functional or declarative approach can automatically optimize the implementation to meet certain constraints, like speed or energy consumption. There are already some papers about the building of such databases. In a paper by the two crucial problems in component-based software development are component identification and component selection. The main purpose of the paper is to provide a reference point for future research by

[21] Kalkan Okur, E., Yesevi Okur, F., and Altunişik, A.C., 2018. Applications and usability of parametric modeling. Journal of Contemporary Engineering and Management Innovations, [online] October.

[22] Stevek, J., Kvasnica, M., Fikar, M. and Gomola, A., 2017. A Parametric Programming Approach to Automated Integrated Circuit Design. IEEE Transactions on Control Systems Technology, [online] (99), pp.1-12..

categorizing and classifying different component identification and component selection methods and emphasizing their respective strengths and weaknesses. [23]

On the flip side, for those accustomed to thinking in terms of implementation and hardware details, the functional approach might initially seem counterintuitive or hard to grasp. Debugging is another challenge. Since the focus is on the "what" rather than the "how," identifying and resolving implementation-level issues can be more intricate. A system that uses textual prompts to determine implementation details can become a "black box", making it difficult for designer s to understand the decision-making process.

2.5 State / Events Design Model

Both the event-based model and the state-based model are used to describe a system's behavior, but each places emphasis on particular aspects. The state-based model emphasizes the different states a system can assume and the transitions between these states. State-based modeling is also used in simulating discrete-event systems, employing finite state machines (FSM) and timed automata mechanisms. [24] In contrast, the event-based model primarily focuses on external or internal events that trigger a response in the system. The advantages and disadvantages of the state-based model can be compared to those of the event-based model regarding synthesis. In embedded systems, state-based models can be utilized to develop statecharts aiding in event-based control and test systems, thus showing a level of compatibility and complement with event-based models[25]

The event-based model centers on the system's responses to specific events, both internal and external, making it especially useful for simulating and prototyping dynamic and reactive systems. This approach is excellent for capturing system

[23] Software component identification and selection: A research review. Shabnam Gholamshahi, Seyed Mohammad Hossein Hasheminejad, First published: 31 October 2018.

[24] Byoung Kyu Choi and Donghun Kang, 2013. State-Based Modeling and Simulation. In: Modeling and Simulation of Discrete-Event Systems, Wiley Online Library, Chapter 9.

[25] Shelley Gretlein, 2013. Software Modeling for Embedded Systems. In: Robert Oshana and Mark Kraeling, ed., Software Engineering for Embedded Systems, Newnes.

dynamics, such as state changes in response to external inputs or specific timings. Because of its modular nature, each event is generally treated as a separate unit, making it easier to add, remove, or modify without impacting the entire project. This separation also simplifies debugging and validation, as individual events can be easily tested. Furthermore, the model is well-suited for simulating system behavior over time and offers an intuitive visual representation, with events and transitions being easily mapped out. In the domain of embedded systems itself, and therefore not in the softwares used to make them, event-based software architecture has shown merits, especially in dynamic and reactive systems, where responses to internal and external events are pivotal. [26]

However, managing events in complex systems can become challenging due to their inherent complexity. Event-based simulation might also be slower if there are many events and interactions. There's a risk of overfitting, where a system could become too specific to certain events, limiting its flexibility. While individual events might be easy to test, complex interactions can present challenges, and not al l behaviors or interactions can be effectively represented.

2.6 Layered Model

The layered or levels model is a design approach that divides a system into different levels of abstraction or functionality, each of which is based on the level below. [27] These models then form the basis for all subsequent development stages. Creating models for your embedded design provides numerous advantages over the traditional design approach. Using this approach – combined with hardware prototyping – you reduce the risk of mistakes and shorten the development cycle by performing verification and validation testing throughout the development instead of only during the final testing stage. [28] The layered model offers several advantages. Modularity allows each level to be

[26] IMT, (n.d.). Event-based software architecture in embedded system development. [online] Available at: https://www.imt.ch

[27] Bobrow, D., 1984. A Layered Approach to Software Design. Interact Program Environmen.

[28] Shelley Gretlein, 2013. Software Modeling for Embedded Systems. In: Robert Oshana and Mark Kraeling, ed., Software Engineering for Embedded Systems, Newnes.

designed, tested, and optimized independently of the others. This design structure also promotes the reuse of components, as those designed for one level can be easily incorporated into other projects that operate at the same level of abstraction. Moreover, maintainability is enhanced since changes or corrections can be applied to a specific level without affecting the entire system. Furthermore, by breaking the system down into levels, it becomes easier to understand and document.

However, this approach is not without its challenges. Managing multiple levels can introduce complexity into the project. Additionally, each level of abstraction could bring with it some inefficiencies or overhead. One also has to be cautious of interdependencies; a change in one level might inadvertently affect other levels, complicating the implementation process. Lastly, the initial phase of design could be resource-intensive as time and effort are needed to establish and fine-tune each level.

2.7 Conclusions

From the discussions and insights presented in this chapter, various facets of embedded technology design, its challenges, and the transformative potential of emerging technologies have been analyzed. Drawing from these observations, the following conclusions are reached:

The process that characterizes the creation of a product with embedded technologies is rigid and immutable in some of its phases, such as material sourcing, system assembly, and the necessary testing, among others. On the other hand, phases that characterize the conceptual and creative development of the product are more flexible. We also noted how the vast majority of users trying to engage with such a project utilize the component-based system, which has historically established itself as a standard for the industry. In addition to its undeniable benefits, this method also brings numerous challenges, especially regarding the constant evolution of offerings and lack of adequate resources. This quickly leads to the obsolescence of hard-earned knowledge and the steep learning curve associated with this approach.

In every approach to a project that we analyzed, modifications and errors must be kept to the absolute minimum. They bear a significant cost and impact on the project, affecting it in terms of time, money, and frustration for those trying to work on it.

We saw how new technologies, such as A.I., pave the way for the implementation of methodologies that were previously inconceivable. For instance, the declarative approach would have previously required a programmer to write an endless series of "if" statements, but now it has become entirely feasible. Lastly, but not less importantly, we would like to highlight how other methodologies offer valuable alternatives to the component selection method wich is used by all the platforms.

SECTION A – CHAPTER 3
AN ANALYSIS OF THE CONTEXT AND LEARNING PROCESS

3.1 Introduction

In recent years, the field of interaction design and product development has witnessed a significant shift towards the integration of electronics and embedded systems in everyday products. However, as the technological landscape evolves, professionals face numerous challenges in adapting to new requirements.[29] This chapter aims to shed light on the reasons behind this knowledge gap, focusing on the slow and expensive learning process, limited resources, and the vast amount of information and skills required.

3.2 Expanding field with diverse technologies;

As we already said, the field of electronics and embedded systems is continuously evolving, with new technologies and components emerging regularly. From a technical standpoint, embedded systems involve the integration of software components, computer hardware, sensors, and actuators into larger mechanical and electronic systems. Embedded software is central to enabling critical features within these systems.[30] Staying up-to-date with advancements and understanding the various types of components and technologies, their datasheets, and specifications can be overwhelming. Designers and engineers often find themselves juggling multiple responsibilities, making it challenging to dedicate sufficient time to broaden their knowledge base. Working with discrete electronic components requires a holistic understanding of hardware and software integration. Due to this complexity,

[29] G. Karsai, F. Massacci, L. Osterweil and I. Schieferdecker, "Evolving Embedded Systems," in Computer, vol. 43, no. 5, pp. 34-40, May 2010, doi: 10.1109/MC.2010.135.

[30] Cracking the complexity code in embedded systems development, Johannes Deichmann March 25, 2022 | Article

designing products with correct PCBs traces and electronic components often requires close collaboration between interaction designers, product designers, and engineers. Adopting agile methods in the development process of embedded systems has been proposed as a means to address the growing complexity and evolving requirements in this domain. [31] However, bridging the gap between these disciplines and ensuring effective communication can be challenging. Designers may lack a deep understanding of electronics, while engineers may struggle to grasp user-centered design principles. A case study highlighted the level and nature of integration between design and engineering disciplines and aimed to understand the execution of interdisciplinary/transdisciplinary education.[32] Establishing a collaborative environment that fosters knowledge sharing and interdisciplinary teamwork is essential to overcome these challenges. Design education programs face challenges in keeping pace with these advancements. Outdated curricula may fail to incorporate the latest trends, leaving graduates unaware of the current best practices and cutting-edge techniques. As a result, professionals entering the industry may find themselves lacking the necessary knowledge and skills to work with modern electronic systems. limited resources, and the vast amount of information and skills required.

3.3 Many different users and needs

In many educational institutions, design and engineering disciplines are taught separately, with limited integration between the two. Science team's integrative capacity is pivotal in facilitating both social and cognitive integration processes, which are essential for interdisciplinary collaboration and knowledge sharing. [33] This separation often results in a lack of cross-disciplinary knowledge exchange. Interaction designers and product designers may receive minimal exposure to engineering principles, including electronics, while engineers may have limited

[31] Kaisti, M., Rantala, V., Mujunen, T. et al. Agile methods for embedded systems development - a literature review and a mapping study. J Embedded Systems 2013, 15 (2013).

[32] Klaassen, R.G., 2018. Interdisciplinary education: a case study. European Journal of Engineering Education, 43(6), pp.842-859.

[33] Salazar, M.R., Lant, T.K., Salas, E. et al., 2012. Facilitating Innovation in Diverse Science Teams Through Integrative Capacity. Small Group Research, 43(5).

understanding of user-centered design and aesthetics. This disconnect hinders the development of well-rounded professionals capable of effectively working to products composed mainly or partially with electronic components. While interaction designers and product designers may excel in user-centered design and aesthetics, they may lack the necessary programming and firmware development skills to effectively leverage SMD components. Similarly, engineers may face challenges in translating their software expertise into seamless hardware integration. An article published in a Sage journal emphasizes that interdisciplinary teams, with members holding different expertise, harbor a multitude of perspectives that potentially elevate innovation. However, it also notes that teams often falter in integrating their expertise, which hinders them from realizing their innovative potential. [34] Normally those fields are kept separate one from each other, but as we said, the need for blending technologies in everyday products is demanding designers to have new eyes. To take a new product from inception through to launch requires leaders who know how to put together budgets, schedules, motivate teams, and plan activities. [35]

Also, the rise of the Makers community has significantly impacted the landscape of embedded technologies and electronics design. This community, thanks to many factors ranging from the education available on the web to the miniaturization of electronic devices, is now composed of individuals from diverse backgrounds, ranging from artists and designers to software developers and mechanical engineers. While this diversity fosters innovation and collaboration[36], it also introduces challenges in knowledge sharing and standardization, especially in the context of working with embedded technologies. Often, members of the Makers community may have specialized expertise in a particular area but lack a comprehensive understanding of the complex ecosystem of modern electronics. For example, a software developer might be proficient in coding but have limited knowledge of hardware design, and vice versa for a mechanical engineer. This disparity becomes

[34] Vestal, A. and Mesmer-Magnus, J., 2020. Interdisciplinarity and Team Innovation: The Role of Team Experiential and Relational Resources. Small Group Research

[35] Tennant, D. (2022) Product Development. 1st edn. Wiley

[36] Tan, M., Yang, Y. & Yu, P., 2016. The Influence of the Maker Movement on Engineering and Technology Education.

particularly noticeable when dealing with the intricate requirements of SMD components, where a deeper, interdisciplinary understanding is essential for effective design and implementation. Moreover, the rapid growth of this community has led to the proliferation of DIY guides and tutorials, many of which may not meet industry standards or best practices. This further complicates the learning process for individuals who are new to the field and rely on such resources for self-education.

3.4 Lack of educational support

One of the glaring challenges facing designers and engineers in the realm of Surface Mount Device (SMD) technology above Arduino is the paucity of clear educational resources tailored to this specialized area. The generalization in educational materials on Surface Mount Device (SMD) technology and its impact on those seeking a deeper understanding is indeed a pertinent issue. A notable reference discussing the adaptation of modern SMD prototyping practice to learning is highlighted in a feasibility study on ResearchGate.[37] While the internet and bookstores are awash with materials focused on basic electronics and beginner-friendly platforms, these resources often only scratch the surface of what is required to be proficient in design and fabrication. Interaction designers, product designers, and engineers typically receive their training in design schools or engineering programs. However, these curricula often lack sufficient emphasis on electronics and working with SMD components.[38] Consequently, professionals entering these fields may possess limited knowledge about electronic circuits, PCB design, and soldering techniques, causing a steep learning curve when transitioning to projects involving SMD components. The difficulty begins with the nature of electronic components themselves, which are considerably more complex and varied than basic electronic elements. The diversity of SMD components, each with its own specifications, applications, and limitations, makes a one-size-fits-all approach to education and training impractical. And yet, most existing educational materials often opt for a simplified, generalized approach,

[37] Papanikolaou, V., 2015. Surface-mount device prototyping in education - A feasible alternative to conventional through-hole practice.
[38] Papanikolaou, V., 2015. Surface-mount device prototyping in education - A feasible alternative to conventional through-hole practice.

leaving those interested in diving deeper into SMD technology in a lurch. Even online tutorials or courses that purport to delve into advanced electronics topics often bypass the nuanced intricacies of SMD component selection, circuit design, and assembly techniques. Given the precise and highly specialized skills needed—ranging from understanding complex datasheets to mastering delicate soldering methods—the lack of focused educational content creates a significant hurdle. Those who seek to self-educate find themselves piecing together information from disparate sources, which is not only time-consuming but can also lead to gaps or inconsistencies in understanding. Moreover, the rapid pace at which electronic technology evolves only exacerbates the issue. Even when educational materials for SMD technology do exist, they can quickly become outdated, leading to a vicious cycle where aspiring designers and engineers are perpetually playing catch-up, armed with incomplete or obsolete information.

This absence of comprehensive and up-to-date educational materials restricts the accessibility of this type of technology, effectively creating a knowledge gap that many find hard to bridge. The lack of structured learning resources puts the onus on the individual to piece together a coherent understanding, which can be a daunting, confusing, and often inefficient process.

3.5 Conclusions

As we reflect upon the analysis of the context and learning processes surrounding embedded product development, it becomes evident how intertwined economic growth, technological evolution, and educational challenges are shaping the industry's trajectory. The development process for embedded products, undoubtedly influenced by the massive economic growth rates that have marked the sector, can reach levels of extreme logistical complexity even for simple projects and require significant coordination from a large group of individuals. Due to the high number of technologies that emerge daily, users struggle to stay up-to-date and increasingly rely heavily on the internet and the opinions of highly specialized experts. For this reason, the teams working on these types of projects need to be larger and more resource-intensive in economic terms.

The recent market expansion, coupled with the development of the making community, has allowed access to these types of technologies to actors from very different backgrounds. However, this represents a challenge from an educational standpoint since the documentation remains scarce, unsuitable to be understood by everyone, and schools and universities maintain a limited emphasis on electronics.

SECTION A – CHAPTER 4
OVERVIEW OF CURRENT HARDWARE SOLUTIONS TO HELP DESIGNERS (ARDUINO AND MODULES PROTOTIPATION)

4.1 Introduction

In the evolving landscape of embedded systems and electronics, due to the difficulties shown in the previous chapters, Arduino and its accompanying array of component modules stand out as an often-chosen path for designers and engineers alike for making prototypes.[39] This chapter aims to provide a comprehensive examination of this popular approach, dissecting the allure that draws professionals to Arduino while laying bare the complications and limitations that often accompany this route. As an enabling technology, Arduino has democratized access to the world of electronics, making it feasible for individuals with various skill levels and backgrounds to prototype and bring their ideas to fruition. [40] However, what begins as an expedient solution for prototyping can sometimes turn into a labyrinth of design challenges and technical constraints, impacting everything from ergonomics and aesthetics to manufacturing feasibility. In the pages that follows, after detailing the numerous benefits of Arduino and modular component prototyping, we will shift our focus towards unearthing the often-overlooked complications and limitations inherent in this approach. From design considerations like physical form factor and user experience to technical constraints like processing power and integration issues.

[39] McRoberts M. Beginning Arduino(second ed.), Apress, USA (2013)

[40] Oellermann, M., Jolles, J.W., Ortiz, D., Seabra, R., Wenzel, T., Wilson, H. and Tanner, R.L., 2022. Open Hardware in Science: The Benefits of Open Electronics. Integrative and Comparative Biology, 62(4), pp.1061-1075. https://doi.org/10.1093/icb/icac043. Published: 20 May 2022

4.2 Overview of benefits of Arduino and module prototipation

The many advantages of Arduino are a well-known and described topic. [41] [42] [43] When embarking on a journey into the realm of electronics and embedded systems, many designers and engineers gravitate toward the comforting embrace of Arduino and modular components. The preference for these over discrete components isn't arbitrary; it's rooted in several compelling reasons that make Arduino an attractive starting point for various projects. [44]

Firstly, let's consider the allure of accessibility and ease-of-use that Arduino offers. This open-source platform has democratized electronics and programming by making them remarkably accessible. [45] The user-friendly programming interface is designed to be intuitive, allowing even those with a minimal background in electronics to start tinkering. Moreover, the sprawling community of Arduino users provides a supportive environment replete with tutorials, libraries, 3d models and forums. In a domain as complex as electronics, where a knowledge gap can be intimidating, the ease with which one can get started on Arduino is undeniably a strong pull. [46]

Arduino prototipation is agevolated and comes with modular components. Here is where speed and efficiency come into play. Instead of navigating through the cumbersome processes of selecting, purchasing, and soldering individual discrete components, you can plug in pre-fabricated modules and hit the ground running. The

[41] D'Ausilio AlessandroArduino: A low-cost multipurpose lab equipment Behav. Res. Methods, 44 (2012), pp. 305-313

[42] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021). A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. Computer Science Review, 40.Behav. Res. Methods, 44 (2012)

[43] A. Hars, Working for free? Motivations of participating in open source projects, in: Proceedings of the 34th Hawaii International Conference on System Sciences, Maui, HI, USA, 2001, pp. 1–9.

[44] ABanzi M., Shiloh M.Make: Getting Started with Arduino the Open Source Electronics Prototyping Platform (3rd. ed.), Maker Media, Inc, Sebastopol, CA, USA (2014)

[45] D'Ausilio AlessandroArduino: A low-cost multipurpose lab equipment Behav. Res. Methods, 44 (2012), pp. 305-313

[46] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021). A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. Computer Science Review, 40.Behav. Res. Methods, 44 (2012), pp. 305-313

modular approach also limits the margin for human error. Since each module is pre-designed to perform a specific function, and often comes with built-in safety features, the odds of causing electrical mishaps through incorrect connections are significantly reduced. In an arena where mistakes can be costly, both in terms of time and resources, this is a notable advantage.

Arduino and its ecosystem of modular components are not rigid; they offer considerable flexibility. This means you can easily tweak, adapt, and expand your projects without having to start from scratch. In the iterative process of prototyping, where design changes are almost a given, the adaptability offered by Arduino and modular components is incredibly valuable. In terms of cost, Arduino provides an economically viable entry point. While the costs might escalate with the complexity of the project and the number of modules you end up using, the initial financial barrier is relatively low. This opens up the world of electronics to hobbyists and small development teams. [47]

Despite the multitude of advantages offered by Arduino and modular components, it's critical to acknowledge that they are not without their limitations. Often, the convenience and rapid prototyping capabilities come at the expense of other important factors. Many designers and engineers have found that, while Arduino-based prototypes are excellent for initial concept validation or small-scale projects, they may not be wholly representative of the final product.

For instance, while modular components are plug-and-play, they sometimes lack the specificity and optimization that discrete components can offer. This can be especially problematic when the final product demands highly specialized functionalities, stringent power requirements, or particular form factors that modular components simply can't provide.

Furthermore, it's not uncommon to encounter compatibility issues or limitations in the modules themselves, which can stifle innovation or add unexpected roadblocks in the development process. Sometimes what begins as a quick and easy Arduino

[47] Blum, J. (2019) *Exploring Arduino*. 2nd edn. Wiley.

project can end up requiring a transition to more complex electronics, incurring a cost in both time and resources.

In the next sections, we will delve into these constraints and challenges in greater detail, demonstrating that while Arduino and modular components serve as excellent entry points and educational tools, they are not always the most suitable options for prototyping advanced or commercial applications, not even in the first phases.

4.3 Limitations under a design point of view;

4.3.1 Phisical size and form factor

The physical size and form factor of an Arduino board can be both an advantage and a limitation, depending on the specific application or project requirements. The standard Arduino Uno board measures approximately 68.6 mm by 53.4 mm, making it quite compact. [48] This small size is usually advantageous for prototyping as it fits comfortably on a breadboard or inside a project enclosure. However, in projects where space is at a premium or where the board needs to be embedded into a very small or uniquely shaped device, the physical dimensions of the board can be a limiting factor. The form factor of the Arduino boards is another consideration. All the components and pins on an Arduino board are fixed in place, and while this standardization makes the board easy to use, it offers little flexibility. For example, if a project design requires that the USB port or the power jack be placed in a different location, or the pins to be arranged differently, the fixed layout of the Arduino board can be restrictive.[49] Moreover, it's also essential to remember that the physical form of the Arduino board includes components that may not be necessary in a finalized product, such as the USB interface for programming. These extra components take up space and can also consume additional power, both of which might be scarce in a finished product design.

[48] https://en.wikipedia.org/wiki/Arduino_Uno

[49] Bolanakis, D. (2021) Microcontroller Prototypes with Arduino and a 3D Printer. 1st edn. Wiley

4.3.2 Ergonomics and User Experience

Ergonomics and user experience form an integral part of the design process when transitioning from an Arduino prototype to a final product. These factors significantly influence how a user interacts with the product and their overall satisfaction with the product. Arduino-based prototypes often prioritize technical functionality over user ergonomics. They might use generic modules and components, have exposed wires and PCBs, or lack a user-friendly interface. [50] While this approach is often suitable for prototyping and testing technical functionalities, it is usually not acceptable for a final product. Final products need to be designed with the end-user in mind. Ergonomic considerations include how the product fits and feels in the user's hand, the positioning and operation of buttons or controls, the visibility and readability of displays, and even factors like weight and balance.[51] A positive user experience can greatly enhance a product's acceptability and success in the market.

4.3.3 Aesthetic Considerations

The transition from an Arduino prototype to a final product involves several design considerations, one of which is aesthetics. While aesthetics might not be the primary focus during the prototyping phase, it becomes a crucial aspect when designing a product for the market. Arduino prototypes, made with off-the-shelf boards and components, usually do not prioritize aesthetics.[52] As we said, they are designed for functionality and ease of use, which can result in a bulky and circuit-like appearance with exposed wires and PCBs. In contrast, final products often need to have a sleek and attractive appearance, as aesthetics significantly influence a product's market acceptance. The aesthetic design of a product encompasses several elements, such as

[50] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021). A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. Computer Science Review, 40.Behav. Res. Methods, 44 (2012), pp. 305-313

[51] Bruno, D.M. (2009) *Questione di metodo. Analisi, sintesi, teorie e casi di studio sulla cultura del progetto.* 1st edn. Aracne.

[52] Blum, J. (2019) *Exploring Arduino*. 2nd edn. Wiley.

color, shape, size, and materials. These factors not only need to be pleasing to the eye but also coherent with the product's branding and target audience. Additionally, the aesthetic design may need to comply with industry-specific design trends or regulatory requirements. Therefore, while aesthetics may not be a primary concern during the Arduino prototyping phase, it becomes a critical factor when designing a final product to be tested for the market. Understanding the importance of aesthetics and incorporating it early in the design process can help ensure a smooth transition from prototype to product.

4.3.4 Manufacturing feasibility

The move from an Arduino prototype to a final product introduces several manufacturing considerations. While Arduino allows for relatively easy prototyping with its modular design and broad ecosystem, this ease does not necessarily transfer over to the process of manufacturing a final product. [53]

Arduino-based prototypes are typically hand-assembled and involve off-the-shelf components, which are great for building and testing individual units but may not be suitable for mass production. In contrast, the manufacturing of a final product often involves large quantities and requires the design to be compatible with automated assembly and testing procedures.

Several factors that is impossible to fore-see using Arduino's can impact manufacturing feasibility. Therefore, the design should also take into account the manufacturing processes that will be used. This includes considerations for PCB layout for easier automated soldering, assembly-friendly design to minimize assembly errors, and test points for automated testing.

4.4 Limitations under an engineering point of view;

[53] Hari Kishan Kondaveeti, Nandeesh Kumar Kumaravelu, Sunny Dayal Vanambathina, Sudha Ellison Mathe, Suseela Vappangi, A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations, Computer Science Review,Volume 40.

4.4.1 *Limited Input and Output pins*

One of the defining features of Arduino boards is the number and variety of input/output (I/O) pins they offer for interfacing with other devices and components. Depending on the model, an Arduino board typically provides digital I/O pins, analog input pins, and specialized pins like PWM outputs or communication interfaces such as SPI, I2C, and UART. The standard Arduino Uno, for instance, has 14 digital I/O pins and 6 analog input pins. [54] While these are sufficient for simple projects, the limited number of pins can become restrictive when building more complex systems. The restriction applies not only to the number of devices you can connect but also to the variety of these devices due to specialized interfaces.

4.4.2 *Processing power and memory constraints*

Arduino microcontrollers are popular for their versatility and ease of use, but they do have inherent limitations when it comes to processing power and memory.[55] Most Arduino boards, such as the popular Arduino Uno, use an 8-bit AVR microcontroller, which operates at 16 MHz. This processing speed may suffice for simple tasks such as blinking LEDs or reading sensor data, but it can quickly become a bottleneck when dealing with complex calculations or data-intensive operations. The limitations become evident when attempting to perform complex tasks such as fast Fourier transforms, digital signal processing, or real-time image processing, which require high-speed and complex computations. Moreover, Arduino boards typically have very limited memory. For example, the Arduino Uno has 32KB of flash memory and 2KB of SRAM. This is another limitation that can affect application complexity. [56]

---

[54] Wikipedia, 2023. Arduino Uno. Available at: https://en.wikipedia.org/wiki/Arduino_Uno

[55] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021). A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. Computer Science Review, 40.

[56] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021). A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. Computer Science Review, 40.

4.4.3 Integration and Further Developement Challenges - Hardware:

Integration challenges form a significant part of the transition from an Arduino prototype to a final product. While Arduino boards are designed for easy integration of components in the prototyping stage, this simplicity might not translate seamlessly into a final product. During the prototyping stage, Arduino’s wide range of compatible modules, also known as shields, allow for the easy addition of various functionalities. You can stack these shields onto your Arduino board to add sensors, communication modules, motor controllers, displays, and more. [57] This plug-and-play nature is one of Arduino's strengths during the initial prototyping phase. However, when moving to the final product, such integration can pose challenges. One common challenge is the transition from Arduino shields to discrete components. Shields are usually not suitable for final products due to their size, cost, and often unnecessary extra features. Instead, developers use standalone components that need to be integrated onto a custom PCB. This step requires significant electronics design expertise and often additional considerations about component compatibility, electrical noise, and signal integrity. Software compatibility issues can also arise during integration. [58] Code written for an Arduino prototype might not work flawlessly on the final hardware due to differences in architecture, clock speed, or peripherals. Debugging these issues can be more complex compared to debugging on an Arduino board. Furthermore, while prototyping, developers might overlook certain component specifications that become critical in the final product, such as operating temperature or tolerance. These overlooked specifications can lead to integration challenges in the final product. To overcome these challenges, developers should plan their transition from prototype to final product carefully. In conclusion, while Arduino simplifies prototyping by easing component integration, developers must be prepared to address a range of integration challenges when moving to a final product. Recognizing these challenges ahead of time and planning for them can greatly smooth the transition from prototype to product.

[57] Arduino, Shields & Carriers. Available at: https://store.arduino.cc/collections/shields-carriers)

[58] Kondaveeti, H.K. and Mathe, S.E., 2021. A Systematic Literature Review on Prototyping with Arduino: Applications, Challenges, Advantages, and Limitations. Computer Science Review, 40(38)

4.4.4 Integration and Further Developement Challenges – Firmware:

When prototyping with Arduino, developers benefit from the user-friendly Arduino Integrated Development Environment (IDE). The Arduino IDE provides an easy-to-use platform for writing code, compiling it, and uploading it to an Arduino board. Its simplicity, however, can create a form of lock-in, making it difficult for developers to transition to other IDEs when switching to different microcontrollers for more advanced or specialized projects. Arduino's IDE abstracts away many of the complexities associated with programming microcontrollers. For example, it automates the process of setting up the build environment, compiling code, and transferring the compiled code to the microcontroller. While this simplification is great for beginners, it also means that developers may not get exposure to these important aspects of microcontroller programming. [59] Different IDEs often have varying interfaces, functionalities, and workflows, and they may also require explicit configuration of build settings, deeper understanding of the underlying hardware, or different programming languages. Consequently, when an Arduino developer attempts to switch to another IDE such as PlatformIO, Atmel Studio, or the ARM mbed platform, they may face a steep learning curve. [60] Moreover, Arduino's libraries and code examples, which are designed to be simple and easy to use, may not be directly compatible with other platforms or microcontrollers. If developers are heavily reliant on these resources, they may find it challenging to rewrite their code or find equivalent libraries for a different platform. Despite these challenges, it's important to note that gaining familiarity with other IDEs and microcontrollers can open up a broad range of possibilities that aren't available within the Arduino ecosystem. Other microcontrollers may offer higher performance, more specialized peripherals, lower power consumption, or better support for real-time or multitasking applications.

[59] Marzoli, I., Rizza, N., Saltarelli, A., & Sampaolesi, E., 2021. Arduino: From Physics to Robotics. In: D. Scaradozzi, L. Guasti, M. Di Stasio, B. Miotti, A. Monteriù, & P. Blikstein, eds. Makers at School, Educational Robotics and Innovative Learning Environments.

[60] P. O. Muller, C. Stich and C. Zeidler, "Components @ work: component technology for embedded systems," Proceedings 27th EUROMICRO Conference. 2001: A Net Odyssey, Warsaw, Poland, 2001.

4.4.5 Power Consumption and Battery Life

Power consumption and battery life are crucial factors in the development of most electronic devices, especially for portable or wireless devices. However, they often pose challenges when transitioning from an Arduino prototype to a final product. In prototyping stages, where the main focus is on functionality and proof-of-concept, power consumption is often a secondary concern. Arduino boards, for their ease of use and broad functionality, are not particularly power-efficient. They often include several on-board components such as voltage regulators, USB interfaces, and LED indicators, which continuously draw power, even when not in use. Moreover, the default settings for many Arduino boards and libraries do not prioritize power optimization. For instance, the processor may run at full speed even when it's not necessary, and power-saving modes may not be utilized. As a result, an Arduino prototype may consume significantly more power than a final product designed for power efficiency. For small systems, the best choice may be to have no OS at all. The embedded software may consist of a program that runs a control loop that polls devices for activity, or responds to flags being set by interrupt handlers. This solution is simple, inexpensive, and is often the best choice for small high-volume applications where minimizing cost is a key driver. [61] When transitioning to a final product, power consumption and battery life become critical considerations, especially for battery-powered or energy-harvesting devices. High power consumption not only drains the battery faster, reducing the device's operational time, but can also lead to excessive heat dissipation, which might require additional thermal management solutions.

To optimize power consumption and extend battery life, a number of strategies can be applied. These might include selecting low-power components, optimizing the software to use power-saving modes of the microcontroller, and designing the system to efficiently manage power. It may also be beneficial to replace the general-purpose microcontroller used in the Arduino with a more specialized, low-power microcontroller for the final product.

[61] Fowler, K. (2014) Developing and Managing Embedded Systems and Products. Elsevier Science. Chapter 10 - Electronic Design

In conclusion, while power consumption and battery life might not be a primary concern during the Arduino prototyping stage, they become critical factors in the design of a final product. Therefore, it's important to anticipate these factors in the early stages of design and include power management strategies in the transition from prototype to product.

4.5 Conclusions

As designers and engineers venture into prototyping, Arduino often emerges as a beacon of accessibility and functionality. Yet, as with any tool or approach, it brings its set of challenges.[62] Arduino's main benefits include its accessibility and ease-of-use, a supportive community, the reduced margin for error due to its modular design, flexibility in prototyping, and cost-effectiveness for both hobbyists and professionals. Very often, is the only way designer and engeneers can quickly and phisically see the result of their work. Adopting Arduino comes with limitations under a Design point of view. Arduino boards come with a set physical size, making their adaptability rigid for projects demanding specific dimensions. While Arduino excels in prototyping with its modular and user-friendly approach, it might not always cater to intricate design facets like ergonomics and aesthetics. Prototypes based on Arduino typically underscore technical functionality, potentially sidelining user experience and aesthetic considerations. Transitioning from such a prototype to a mass-produced product can be fraught with challenges. The inherent design and structural constraints of Arduino-based projects might not always be congruent with the prerequisites of a real production, even in small scales, necessitating design modifications for both manufacturability and market resonance.
There are several Engineering Limitations in the use of microcontrollers like Arduino. Similar boards are constrained by a fixed number of I/O pins, affecting multi-device interfacing. Predominantly operating on 8-bit AVR microcontrollers, their processing power struggles with complex operations, and the limited memory (e.g., Arduino Uno's 32KB flash and 2KB SRAM) restricts application

[62] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021). A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations. Computer Science Review, 40.

expansiveness. In today's demanding projects, power constraints are a significant drawback, often necessitating the replacement of the Arduino microcontroller after initial tests. Transitioning from prototyping with shield-based designs to discrete components in final products demands intricate electronics expertise, with considerations around component compatibility, signal integrity, and more. Developers relying on the user-friendly Arduino IDE may face challenges when transitioning to advanced development environments. Additionally, power efficiency is not a primary design consideration for Arduino, posing concerns for final battery-operated products.

SECTION A – CHAPTER 5
OVERVIEW OF CURRENT SOFTWARE SOLUTIONS TO HELP DESIGNERS

5.1 Introduction

After an overview of the Embedded System design state of the art and the lack of satisfing all the project needs by newer solutions like Arduino and similar microcontrollers and their hardware environment, we will now focus on analyzing the softwares and therefore digital solutions born to help the user in designing their embedded products and prototypes.

In the landscape of electronics and embedded systems design, software tools have emerged as essential companions for designers and engineers. These platforms provide the much-needed infrastructure for turning abstract ideas into workable designs. The role of these tools extends far beyond merely providing a space to sketch out ideas. They have the capability to intelligently manage the complexity inherent in modern electronic systems, where each component is not just a discrete entity but part of a networked ecosystem. By offering features such as automated routing, real-time error checking, and extensive libraries of predefined components, these platforms dramatically accelerate the design process.

Furthermore, these software tools are particularly valuable in breaking down the silos that traditionally separate different areas of expertise. Engineers and designers can now collaborate more effectively, working from a common platform that caters to both their needs. In an environment where interdisciplinary skills are becoming more crucial than ever, these platforms serve as a lingua franca, bridging gaps in terminology, workflow, and end goals. Investing in these advanced software tools is not merely an expenditure but a strategic move. It's an investment that pays off in terms of time saved, errors avoided, and the quality of the end product. Given the high costs associated with prototyping, manufacturing, and testing in the electronics industry, any reduction in the design cycle or error rate can result in substantial savings.

The state-of-the-art in software engineering for embedded systems, tho, is far behind other application areas. [63] Software for embedded systems is typically monolithic and platform-dependent. These systems are hard to maintain, upgrade and customize, and they are almost impossible to port to other platforms. Component-based software engineering would bring a number of advantages to the embedded systems world such as fast development times, the ability to secure investments through re-use of existing components, and the ability for domain experts to interactively compose sophisticated embedded systems software. [64]

5.2 Market players

These are the market players examined in our research, succinctly summarized by their primary distinguishing features:

Altium Designer stands out as the most comprehensive and complex software, ideal for engineers and professional projects. It is often used in industrial and research contexts. It boasts high interoperability and potential for AI integration but may require frequent consultation of its documentation and a solid background in the discipline.
Cadence Allegro and OrCAD are geared towards professional and engineering use. They are frequently employed in industrial settings where the complexity of projects is high. They offer strong simulation capabilities and interoperability but are not strongly community-driven and can be quite complex.

[63] P. O. Muller, C. Stich and C. Zeidler, "Components @ work: component technology for embedded systems," Proceedings 27th EUROMICRO Conference. 2001: A Net Odyssey, Warsaw, Poland, 2001
[64] P. O. Muller, C. Stich and C. Zeidler, "Components @ work: component technology for embedded systems," Proceedings 27th EUROMICRO Conference. 2001: A Net Odyssey, Warsaw, Poland, 2001

SolidWorks PCB is distinct for its focus on integration with mechanical design, making it suitable for projects that require both electronic and mechanical design.
Autodesk Eagle and KiCad represent more balanced options, suitable for small to medium businesses, freelancers, and advanced hobbyists. They have moderate complexity and provide a good number of online resources.
TinkerCAD, Fritzing, and EasyEDA are Ideal for beginners or less complex projects. They lean more towards education and hobbyists, with a focus on Arduino. These software applications are often used in educational contexts or for personal projects and tend to have a more intuitive user interface. CircuitMaker, DesignSpark PCB, and DipTrace are more geared towards the community, they provide a platform for emerging designers and hobbyists. They aim to bridge the gap between hobbyist software and professional tools, offering a set of intermediate features at an affordable price.

5.3 Commonly observed features

These are the market players examined in our research, succinctly summarized by their primary distinguishing features:

Component-based Approach: All these softwares utilize a component-based approach, accessing libraries of discrete components or component modules depending on their precision and complexity. All platforms suitable for entry-level do not use discrete components but modules that already present other often unspecified passive components within them.
Simulation Tools: The ability to simulate circuit behavior before the prototyping phase is offered by some platforms to save time and resources. In addition to simulation, advanced tools for analyzing energy consumption, signal integrity, and thermal performance are increasingly common.
Real-Time Error Verification and Control: Many of these programs perform real-time checks while the user is designing, highlighting potential issues such as short circuits or overloads. Layer Management: With the increasing complexity of circuits, layer management becomes crucial. These software allow to easily work with multi-layer circuits. Collaborative Interface: Some programs offer collaborative

functionalities, allowing teams to work simultaneously on the same project, often with integrated version control features. Export and Production: The ability to export designs in various formats suitable for production is another key feature. This can include schematics, layouts, and Bill of Materials (BOM). Hardware-Software Integration: Many modern tools offer some form of integration with software development environments, facilitating the transition from circuit design to firmware programming. Many of these softwares, especially those eyeing entry-level users, have community sections within them where users can communicate and share their projects.Divergences between offers

While these electronic design software platforms may share core functionalities, they are not a one-size-fits-all solution. The divergences between these tools often manifest in their targeting of specific user demographics, the complexity of the projects they can handle, and the ecosystems they integrate with. For instance, some software platforms focus on the needs of large enterprises and professional engineering teams. These often come packed with advanced features like complex simulation environments, high-speed design capabilities, and seamless integrations with other enterprise-level software solutions. These platforms typically require significant investments in terms of both cost and the time needed for mastery. They are designed to handle anything from consumer electronics to aerospace projects.

On the opposite end of the spectrum are tools geared toward hobbyists, freelancers, or small businesses. These platforms are generally more user-friendly and may even be free or relatively inexpensive. While they might lack the high-end features of their enterprise counterparts, they provide sufficient capabilities for less complex projects. They often prioritize ease of use, and some even offer community-shared libraries and designs, making them excellent starting points for those new to electronic design. Then, there are those platforms that aim for a middle ground, offering a balance between user-friendly interfaces and advanced functionalities. These tools might find favor among startups or medium-sized companies that need some level of sophistication without breaking the bank. It's also worth noting that some platforms lean more toward specific types of electronic design. For example, some tools excel in analog circuit design, others in digital or mixed-signal environments, and yet others focus on PCB layout more than circuit simulation.

Educational settings also have their specialized requirements, and certain platforms focus on this by offering simplified interfaces and educational discounts or even free licenses. These tools often prioritize ease of use and learning, offering educational pathways to help newcomers grasp the basics of electronic design. In the realm of collaborative work, software platforms are increasingly integrating features to support multi-disciplinary teams. While some are built from the ground up with collaboration in mind, others offer it as an add-on or a separate suite.

5.4 Conclusions

As we reach the culmination of our in-depth exploration of software solutions designed to aid users in the realm of embedded systems and electronics design, it's crucial to synthesize the diverse nuances and insights gathered. The digital platforms in focus are not just tools; they symbolize the broader trend of integration, intelligence, and interdisciplinary collaboration in the design space. From this investigation, the following important conclusions are drawn:

The research shows space for experiencing new methods, as all these software tools adopt a component-based approach, drawing from libraries of discrete components or component modules depending on their precision and complexity. This results in all the problems we have already analyzed in section A1.3 and as will be seen later in section A4.

There’s a lack of a Unified Platform, since from the analysis we have conducted, it is evident that there isn't a platform that allows for an easy user experience combined with the use of discrete components. In our analysis, it immediately became clear that component modules are the primary solution adopted to facilitate novice users. However, this doesn't allow them to achieve satisfactory results due to all the limitations inherent to this kind of module. (See the dedicated chapter...)

The platforms are constantly balancing the ease of use and the need for specialization. These platforms need to specialize for one user category or another, just as they also cannot offer a service that is useful for both the expert and the novice user.

There is a Rising Role of Online Communities. Communities are proving to be an increasingly adopted element by users. Within these communities, members learn a significant amount about what they need, foster collaboration, and share solutions and expertise.

SECTION A – CHAPTER 6
CONCLUSIONS FROM RESEARCH PHASE

6.1 Conclusions

The table that shows the issues that emerged from the research phase is shown in *Table 11, figure 2*

In analyzing the processes involved, it became evident that the prevalent utilization of the only component method, while common, is laden with a variety of disadvantages. This method's rigidity and high cost underscore the imperative for exploring alternative methodologies, which, although present, remain largely unexplored. The meticulous nature of these processes necessitates an error-averse approach from the inception stages to mitigate further complications down the line.

The context within which these processes operate is dynamically evolving with the market expansion, rendering it increasingly challenging for users of varied backgrounds to remain current. A significant shortfall identified is the dearth of recognized educational resources, which has inadvertently propelled the emergence of online communities as pivotal learning platforms. However, these communities alone are insufficient to bridge the knowledge gap, especially amidst the diversified user backgrounds.

Examining the hardware solutions, the adoption of Arduino, albeit practical, presents certain design limitations. It was found that hardware modules frequently fall short of the requisite standards for crafting realistic prototypes, amplifying the demand for a unified, all-encompassing platform. The absence of such a platform compels users to seek resources externally, in a bid to strike a judicious balance between ease of use and the requisite specialization in embedded technology. This prevalent tendency towards the only component method is mirrored in software utilization as well, highlighting a systemic issue.

The research on users further elucidated the exigency for real-world, physical simulation capabilities to foster a more intuitive and supportive design environment. Users often find themselves navigating a labyrinth with scant guidance, which exacerbates feelings of frustration and loss during the product design phase. The incessant search for external resources epitomizes the dire need for a more integrated, resource-rich platform that caters to both the novice and the seasoned designer. Through a thorough understanding of these identified problems, necessities, and the current landscape, there's a clear call to action for developing holistic solutions. These solutions should aim at obliterating the existing barriers, enriching the resource pool, and fostering a more collaborative, flexible, and user-centric ecosystem in both hardware and software domains.

SECTION B
CONCEPT ELABORATION

SECTION B – CHAPTER 1
SUMMARY OF NECESSITIES

Problem no.1, Difficulty in approaching a project with embedded technologies

The process of developing an integrated circuit for a product is in itself an experience that often proves to be problematic and challenging even for the most experienced Designers and Engineers. In particular, as demonstrated by user journey maps, due to the commonly applied component approach, that with modules instead of discrete components very often don't fit the purpose of creating realistic prototypes, users are forced into a long research phase with scarce documentation, long loops of re-iteration of the work done, difficult project portability, and a wide possibility of error. Additional elements such as the constant need for updates dictated by the rapidly evolving technological landscape or the increasing economic cost of hardly predictable errors, increase the difficulty in approaching these types of projects.

Arising necessity: Hence the need for a tool that allows for rapid and effective realistic prototyping is outlined, so that by optimally choosing the constituent elements in the first instance, the number of iterations that the project must undergo is drastically reduced and the bottleneck that characterizes that phase of the design is widened. It is also necessary that these are discrete components, for the sake of a better realism of the prototype and therefore a greater utility.

Problem No. 2: Software platforms provide insufficient assistance.

Our research highlights how on one hand, the development of a project with Arduino often turns out to be futile, while on the other hand, the software platforms normally used for the development of such projects do not sufficiently assist the users, especially those not very experienced in the field, since a vast amount of knowledge is required on their part. Particularly challenging and time-consuming is the fact that since these platforms operate by using the components as the "key units" to be imported and used in the project, the user, even if more interested in utilizing the technology rather than its components, is required to know them one by one, or consult very long online catalogs to choose the component that suits them, very often omitting to view many others, or online communities. This means that there are not even one-stop platforms for users, who in any case resort to 3D modeling to have a physical visualization.

Arising necessity: Hence, this issue leads to the need for a way to have a quicker learning curve, adopt different methods in support of the user. More documentation, examples, a more intuitive way of making and undoing projects, and all accessible from a single place, which even if virtual, still allows for a physical visualization.

SECTION B – CHAPTER 2
CONCEPT PROPOSED

One-Stop Platform that facilitates the design path leading to the development of a product with embedded components, making it more accurate and less costly in terms of time and resources, usable by Designers, Engineers, Technicians, and Hobbyists.

In particular, the solution in question adopts a more "functional" or "behavioral" approach to the design that occurs within it, overcoming the “object” model. Instead of focusing on the components and their properties and methods, at least initially, it focuses on the actions or behaviors that the system must perform. In this way, the study and preparation regarding the technology is "scaled down" to something carried out by the software. The user no longer has to think and choose the components, but only focus on what the goal of the system being created is, defining only Goals and Relationships between the elements, greatly accelerating the development process and avoiding endless re-iterations.

Lastly, since there are no similar ones, the tool, or simulator, should assist users in the development of projects that utilize SMD technology, helping to overcome the need to use an Arduino or similar even for virtual prototyping, as is the case for platforms like Tinkercad and similar.

SECTION C
INTERFACE ELABORATED

SECTION C – CHAPTER 1
FEATURES OF THE INTERFACE

1.1 Adopts a new Hybrid Approach.

The single most important point when developing a prototyping strategy is to define the purpose of the prototyping. [65] Within this platform, users are not required to know all the appropriate and best components to achieve their set goals. Instead, they only need to think about the actions the system must perform in terms of predicates, as presented in the project brief. Indeed, this platform employs a Hybrid Approach, encompassing both a layered and a declarative structure. This contrasts with the fact that all other analyzed solutions, whether hardware or software, utilize a component-based model, with all the associated problems. To illustrate with an example, if a user wanted to add audio feedback to their system, they could simply select "play audio" and let the system choose the most suitable components for the system's requirements, without having to know each one individually. The interface, therefore, also allows the less experienced user (and especially one without a strong basic electronic knowledge) to approach the design of an electronic product while achieving professional standards.

1.2 Allows for easily add details and specifications.

Thanks to the dichiarative approach, it is easy for the user to specify the needs and the requirements that he has for every action of the system. He only has to select them from the list that the system provides. For example, if he has to save data on a portable peripheral, he might just select "portability" with the needs associated with "save to memory".

---

[65] Elverum, C.W., Welo, T. and Tronvoll, S., 2016. Prototyping in New Product Development: Strategy Considerations. Procedia CIRP, 50, pp.117-122.

1.3 Is programmed to be a constantly re-adapting system.

Within this platform, is easy to add specifications that are general and valid for all the parts of the project (may this be regarding the way of powering the project, the maximum or minimum size...). This is possible thanks to a higher level of automation, that allows the software to choose elements and components remaining inside the bounds that are given by the user, automatically re-adapting the generated system if needed, so the user does not have to change all the components by himself if something is changed.

1.4 Provides constant help and guide for the user.

This feature of the platform is granted by the Step-by-Step process and the informative elements. The step-by-step pages in which the user is "guided" vertically are made to make the user at ease and for him to have a sense of control over the situation. The Informative Elements are the constant companion of the user in his journey. They assure the user does not exit the platform he is using and can always understand what he has in front of him.

1.5 Community Integration:

Recognizing the pivotal role of community in project development, our platform fosters designer curiosity and champions user growth and expertise accumulation.

1.6 Accurate Physical Simulation.

Another pivotal feature of our platform is its capability to offer an accurate physical simulation of the final product, significantly enhancing the workflow. This not only enables users to visualize the end design in 3D but also automatically provides them with an exhaustive bill of materials and the electrical circuit layout. Such comprehensive representation not only assures precision in the design process but also ensures users have all the resources they need at their fingertips.

1.7 Navigate between similar technologies and components.

This feature, combined with the general hybrid approach of the interface, avoids the iterative loop that, according to research findings, negatively marks the phase between choosing components for a project and their actual implementation. Once the system is generated, users can easily explore alternative technologies and components. Not only is this advantageous in terms of cost and resources for users, which is the primary objective, but it also relieves them from frequent feelings of frustration and stress.

1.8 Superior UX and UI.

Prioritizing user-centric design, our platform boasts a clear user interface, modular design, and intuitive interactions. It's designed to cater to varied project scales and ensures ease of navigation across sections, guaranteeing user satisfaction and productivity.

1.9 Adopts a new Market Positioning Perspective.

The product positions itself in the market gap that was identified during the research phase. Primarily, it facilitates the use of MCU and both passive and active SMD components on par with software like Altium. However, it maintains the ease and simplicity typical of programs like Thinkercad and similar, without resorting to Modules and Breakout Boards like Arduino, which lead to the issues already highlighted in the research phase. The product also showcases innovation in its use of AI-driven functions to streamline the process with a level of precision that was, until recently, thought to be unattainable.

SECTION C – CHAPTER 2
FINAL INTERFACE PAGES

The interface is shown in *Table 0, figure 1.*

2.1 Login Page [Table 1, figure 1]

2.1.1 Features:

Login, Register

2.1.2 Page description;

The Login Page offers a straightforward interface for user authentication, featuring fields for email and password entry. The page provides two primary actions: 'Sign In' for existing users and 'Register' for new users. While simple in design, this page serves as the initial point of interaction between the user and the system, setting the tone for the user experience that follows.

2.1.3 Interactable Items;

A - Login;
A1 -> User Inserts Email Address and Email
A2 -> Button that redirects to Home Page
A3 -> Button that redirects to Sign Up Page

2.2 Profile Page [Table 1, figure 2]

2.2.1 Features:

Ability to view and manage one's own user profile;
Ability to upload and modify profile pictures;
Ability to add a brief self-description or bio;

2.2.2 Page description;

The Profile Page is structured to function as a user's personal dashboard for identity and account management. It features several interactable items under the Profile section (A). These include the ability to insert contact information (A1), input a self-description or bio (A2), edit the profile picture via a designated button (A3), and save these edited details (A4). Beyond basic account management, the Profile Page also serves as a platform for users to articulate their personal and professional identity. This is facilitated through options for uploading personal images and adding brief biographies, allowing users to share information like their profession and interests with the community. In addition, the page includes a feature for displaying special badges earned for specific accomplishments and milestones within the software, further contributing to user engagement and community recognition.

2.2.3 Interactable items;

A - Profile;

A1 -> User Inserts Contacts informations

A2 -> User inserts self description A3 -> Button to edit profile picture

A4 -> User saves edited informations

2.3 Community Page [Table 1, figure 3]

2.3.1 Features:

Ability to view projects shared by the community

Ability to search for specific projects through an advanced search function based on the tags assigned to the individual project

Ability to view the date the project was shared

3.3.2 Page description;

The Community Page is the nexus where users of the platform come to share, explore, and engage with each other's projects. It offers several key features designed to enhance the user experience and facilitate a sense of connection within the community.

One of the most fundamental features is the ability to view projects that have been shared by other members of the community. This enables users to get a broad overview of what others are working on, drawing inspiration or perhaps even identifying potential collaborators.

Aiding in the process of discovery is an advanced search function. By using this, users can narrow down the list of projects based on specific tags assigned to each one. Whether they are looking for something very niche or something more general, the tagging system makes it easier to find projects that are most relevant to a user's interests or needs.

Another noteworthy feature is the ability to see the date when each project was shared. This helps users gauge the timeliness of a project, allowing them to distinguish between what's cutting-edge and what may be dated but still valuable. The interactive elements on the Community Page contribute to a smooth and intuitive user journey. There's a button that leads directly back to the Home Page for those who wish to pivot to different aspects of the platform. Another button redirects to the Profile Page, providing quick access to personal projects and settings. Users keen on finding specific projects will find the user input for search filters especially handy. Lastly, clickable elements exist that directly lead to projects by other users, thereby making the process of exploring the community's contributions more interactive and engaging.

3.3.3 Interactable items;

A - Functional Bar;

A1 -> Button that redirects to Home Page

A2 -> Button that redirects to Profile Page

B - Community Projects;

B1 -> User input for filters to be applied in the research

B2 -> User research filters

B3 -> Element that redirects to Project by another User

2.4 Home Page [Table 1, figure 4]

2.4.1 Features:

Ability to create a new project;

Ability to open one of your saved projects, presented in chronological order;

Ability to access Profile and community sections;

Possibility to see date of project creation;

Ability to see new projects selected by the community;

Ability to apply filters to see relevant saved projects;

2.4.2 Page description;

The Home Page serves as a critical hub for user activity and is thoughtfully organized to ease interaction with the software platform. It offers a multiplicity of features, each designed to cater to the unique requirements and curiosities of its diverse user base.

At its core, the Home Page enables the creation of a new project, a feature that sits prominently within the interface, inviting users to embark on their creative journey. For ongoing ventures, the page showcases saved projects, displayed in chronological order, thus providing quick and easy access. The ability to see the creation date of each project adds another layer of user convenience, helping to manage and prioritize ongoing or past efforts.

In addition to personal project management, the Home Page also offers gateways to other vital sections of the platform. Users can effortlessly navigate to their Profile Page or the Community Section, encouraging them to explore and interact beyond their isolated digital workspace.

The inclusion of new projects selected by the community serves a dual purpose. While it provides a snapshot of what's trending or highly regarded within the community, it also acts as a curiosity-sparking feature. By showcasing these projects, the platform encourages users to delve into the Community Page, thus creating a vibrant cycle of exploration and engagement.

The page further aids in project management through a feature that allows users to apply filters to view relevant saved projects. This function enhances the user experience by cutting through clutter, making it easier for users to find what they're looking for without endless scrolling or searching.

As for the interactable items, the Home Page is efficiently laid out to aid quick decision-making. The Functional Bar at the top contains buttons that redirect to the Community and Profile Pages. A separate section dedicated to new community projects not only allows users to click through to individual projects but also offers a button that leads to the broader Community Page. Finally, the section displaying projects by users incorporates search filters and clickable elements that lead to saved projects, enhancing the overall navigational experience.

2.4.3 Interactable items;

A - Functional Bar;

A1 -> Button that redirects to Home Page

A2 -> Button that redirects to the enxt page of the System Building

A3 -> Button that redirects to Settings

A4 -> Button that redirects to Profile Page

B1 -> Button that opens/closes information window (B)

C1 -> Search bar

C2 -> Information Icon, Opens up the information Menu (B) with the informations related to the action selected

C3 -> Button to select the action as an action performed by the system

2.5 System's building – Actions Page [Table 1, figure 5], [Table 2, figure 5.1]

2.5.1 Features:

Ability to select the actions to be performed by the system

Ability to obtain information about each of the proposed actions

Head to the Profile, Home, Settings pages, and move forward in the system building process

2.5.2 *Page description:*

The pages dedicated to system building represent the core of the innovation that the design platform developed introduces in the field of product development with embedded circuits. The development of these pages stems from the following considerations: In the field of hardware development with embedded circuits, defining the components used is the primary and fundamental element that requires the direct attention of the user or designer in charge of the development of the physical product. This is the foundation on which the rest of the system is built. Once this pre-definition is established, many if not all other technical aspects, from firmware processing to subjecting energy consumption to needs, can be managed through automated processes, following predefined standards and parametric logic.

By grouping the components according to the action they perform, it is possible to offer an "event-driven" approach to design, in sharp contrast to the approach based on a difficult and lengthy study of all the components available on the market per-se, which characterizes the other solutions available on the market. This change of paradigm offers a series of advantages, which are explored in the section...

This page incorporates a declarative approach where the user's declarative options and subsequently additional details and features that enrich them are presented through a list, provided by a tailor-made graphical interface (GUI), thus allowing for an optimized user experience, a decision dictated by reasons of simplicity and intuitiveness. This approach is inspired by the natural decision-making processes that occur in the human brain, making the interaction natural. The ease with which users can navigate and make selections within the interface is particularly advantageous in creative contexts, where the freedom to explore different configurations is fundamental. Users can organize and modify elements in an intuitive and non-linear way, exploring different possibilities with great ease and flexibility. Another crucial aspect is the scalability of the GUI-based approach. This system is capable of handling projects of varying sizes and complexities, without compromising clarity or ease of use. Moreover, the graphical interface provides a clear and immediate visual representation of possible actions and their relations, thus facilitating the understanding of the system as a whole.

2.5.3 *Interactable Items:*

A - Functional Bar;
A1 -> Button that redirects to Home Page
A2 -> Button that redirects to the enxt page of the System Building
A3 -> Button that redirects to Settings
A4 -> Button that redirects to Profile Page
B1 -> Button that opens/closes information window (B)
C1 -> Search bar
C2 -> Information Icon, Opens up the information Menu (B) with the informations related to the action selected
C3 -> Button to select the action as an action performed by the system

2.6 System's building – Details Pages [Table 2, figure 6, 6.1, 6.2, 6.3], [Table 3, figure 6.4, 6.5, 7]

2.6.1 *Features:*

Possibility to select important requirements for the actions that the system must perform and for the components it must use;
Possibility to obtain information about each of the proposed actions;
Head to the Profile, Home, Settings pages, and proceed with the system building process;

2.6.2 *Pages description:*

The first of these two pages further details the information provided on the previous page. While the previous page focuses on the actions the system must perform, here we highlight the requirements the system must meet. These requirements are presented in the form of clickable icons to emphasize their importance. For example, if "Portability" is selected in the "Store in Memory" slot, the system will design a system where memory can be extracted. An informative pop-up panel is always available to the user, providing clarifications when needed during their interaction with the system. The second page, “Additional requirements for the system” follows the same logic as the previous one, but here the user can add general informations not only related to a single action.

2.6.3 Interactable Items:

A - Functional Bar;
A1 -> Button that redirects to Home Page
A2 -> Button that redirects to the enxt page of the System Building
A3 -> Button that redirects to Settings
A4 -> Button that redirects to Profile Page
C1 -> Information Icon, Opens up the information Menu (B) with the informations related to the action selected
C2 -> Button to select the requirement associated with the action
C3 -> User Value Input

2.7 Phisical Layout Page [Table 3, figure 8], [Table 4, figure 8.1],

2.7.1 Features:

See the project in its 3 dimensions;
Highlight the different components and the groups of passive components that accompany them;
Rotate the model;
Display information on the components;
Provide an overview of other available technologies;
Allow an easy comparison between the components;
Download the 3D model (STL) of the conceived product;
Switch to the visualization of the electrical circuit and to that of the BOM (Bill of Materials).

2.7.2 Page description;

The Physical Layout Page is ingeniously designed to amalgamate a variety of advanced features, each purposed to streamline and enhance the circuit design and simulation process. Among the key features is the Technologies Overview (Section C), which springs into action when a user clicks on a specific component, unveiling an option to delve into related technologies. This ingenious feature does more than just listing these technologies; it unfurls informative explanations through a side menu (Section B), acting as a compass for users in the vast sea of technological choices, aiding them in making well-informed decisions without the customary need for external research.

Sailing further into the functionalities, we encounter the Compare Components Menu (Section E), a carefully crafted feature that houses a drop-down interface. This menu acts as a crucible where component details are juxtaposed, thus obviating the need for the external web-based searches that are a common staple on other platforms. The journey of discovery doesn’t end here; as we venture into the 3D Integrated Components Layout Space (Section D), we are greeted with a realistic 3D visualization. This feature, akin to a skilled architect, automates the positioning of components in a simulated, yet realistic environment, with a singular aim - to craft simulations that mirror physical realizations with a higher degree of fidelity.

As we meander through the platform, additional functionalities emerge from the backdrop, each with a unique role. The 3D project visualization, component highlighting, and model rotation are like the skilled artisans, each contributing to the creation of a coherent visual narrative. The options to download the 3D model in STL format is like a bridge, connecting the digital design to the physical realm, while the option to switch to electrical circuit and Bill of Materials (BOM) views are like seasoned guides, leading the way as we traverse through the different vistas of the project.

A notable distinction that sets this platform apart from others like Tinkercad, is the introduction of an auto-coupling feature. This feature, like a skilled matchmaker, automatically pairs passive components with their main counterparts. This eliminates the often tedious manual pairing, acting like a time-saving catalyst, and enhancing simulation accuracy, thus serving as a testimony to the platform's commitment to creating a user-centric, efficient, and accurate design and simulation environment.

2.7.3 Interactable Items:

A - Functional Bar;

A1 -> Button that redirects to Home Page

A2 -> Button to clone a component

A3 -> Button to remove a component

A4 -> Button to regenerate the system

A5 -> Button to download the 3D model "STL" to the PC

A6 -> Button that redirects to settings of the interface

B - General Information Menu;

B1 -> Button that folds the Information Menu

C - Technology Alternatives Side Menu;

C1 -> Button that redirects to Home Page

C2 -> Button to clone a component

C3 -> Button to remove a component

C4 -> Button to regenerate the system

D - Components Layout Space;

D1 -> Selected component, opens "C" menu

D2 -> Unselected component, clickable.

D3 -> Interactable element used to modify 3D rotation of the model

D4 -> Button that redirects to CIRCUIT LAYOUT PAGE

D5 -> Button that redirects to BOM page

E- Compare components Menu;

E1 -> Button that changes the component visible in page "D"

E2 -> Button that closes the "E" menu

E3 -> Button that closes the "E" menu

E4 -> Expandable menu containing details

E5 -> Expandable menu containingsummary

2.8 Bill of Materials Page [Table 4, figure 9]

2.8.1 Features:

Maintain traceability of components and their function;
Quick comparison of components;
View cost information of components;
Display additional information;
Possibility to download the BOM file (.CSV);

2.8.2 Page description;

The Bill of Materials (BOM) Page serves a multifunctional role in facilitating and improving project management in the context of circuit design and simulation. One primary utility is the tracking feature that offers detailed traceability of components used, thereby reducing the possibility of errors or oversights that could potentially disrupt the implementation phase. The interface also allows for quick component comparisons, making it easier for users to make informed decisions regarding component selection. Further enhancing project management, the BOM Page provides observable cost information for each component, aiding in accurate budget planning and management. An added functionality is the ability to download the BOM as a .CSV file, facilitating seamless data transfer and communication among project team members, as well as external collaborators, suppliers, or clients. A well-structured BOM not only serves as an internal tool for keeping track of component usage and costs but also acts as an effective communication instrument that ensures all stakeholders have a clear and detailed understanding of the project's scope and requirements.environment.

2.8.3 Interactable Items:

A - Functional Bar
A1 -> Button that redirects to Home Page
A2 -> Button to regenerate the system
A3 -> Button to download the BOM or the CSV file

A4 -> Button that redirects to settings of the interface

B - General Information Menu;

B1 -> Button that folds the Information Menu

C - Bill of Materials;

C1 -> Button that opens the Bill Of Materials Filters Menu (C)

C2 -> Buttons to add different filters to the B.O.M.

D - Mode Selecting Menu;

D4 -> Button that redirects to CIRCUIT PHISICAL LAYOUT

D5 -> Button that redirects to CIRCUIT LAYOUT PAGE

2.9 Electrical Circuit Page [Table 4, figure 10, 10.1]

2.9.1 Features:

Schematic/Diagram Visualization;
Visualization of component groups;
Information and help for understanding the circuit;
Details and focus on the passive components used for each peripheral;
Possibility to download the Gerber File;
Possibility to move towards other technical pages;

2.9.2 Page description;

The The Electrical Circuit Page is an intuitive interface designed to streamline the process of electrical circuit design. It consists of four major sections for ease of use: the Functional Bar for general navigation, the General Information Menu for context on selected components, the Electrical Circuit Space for schematic visualization and interaction, and the Component Details Menu for in-depth electrical information. Users can navigate to the home page, regenerate the system, or export a Gerber file through buttons in the Functional Bar. Within the Electrical Circuit Space, components can be selected or deselected, and users can seamlessly navigate to the Bill of Materials or Physical Layout pages. Additional insights into the electrical connections and related passive components are available through the Component Details Menu. The page offers the flexibility to download Gerber files and navigate to other technical pages, making it an indispensable tool for efficient circuit design.

2.9.3 Interactable Items:

A - Functional Bar

Used to go to homepage, download the electrical file, rigenerate the system and open settings.

A1 - Button that redirects to Home page

A2 - Button that rigenerates the system, sorting it in another way

A3 - Button to export the Gerber file

A4 - Buttons that opens the settings of the interface

B - General Information Menu

Gives information about the selected items

C - Electrical Circuit Space;

In this section, the electrical schematic is displayed. By clicking on a part of it, it's possible to highlight the component and the part of the circuit associated with it, along with commonly associated passive components. By clicking on the parts of the drawing in this section, access is granted to section D. It's possible to move towards other technical pages such as BOM and Physical Layout.

C1 - Unselected component

C2 - Selected component

C3 - Button that redirects to BOM

C4 - Button that redirects to Phisical Layout

D - Component Details Menu;

Gives additional informations and helps the user regarding the electrical (and therefore also the connections) aspects of the system. Is triggered by clicking by any part of the "C" drawing.

D1 - Image of the part of the circuit involving the component

D2 - Additional Electrical Details

SECTION D
DEVELOPING PROCESS

SECTION D – CHAPTER 1
INTRODUCTION TO THE SOLUTION'S DEVELOPEMENT

Section D focuses on the process of developing a user-friendly interface for embedded technology products. This process begins with understanding the user via personas and User Journey Maps (UJMaps) in Chapter 2, which helps in structuring the Information Architecture discussed in Chapter 3. The iterative design approach is then introduced through wireframe development in Chapter 4, with each iteration refined based on user feedback and analysis. This iterative process, employing standard methods at each step, extends to the final interface testing detailed in Chapter 5, where various methods are employed to ensure the interface meets both functional and user experience requirements. The user has been at the center of every part of the design process, validating each step through rigorous testing. The chapters in this section will provide a detailed overview of each phase in the development process, discuss challenges faced, and strategies used to address them, emphasizing the central role of user validation and iterative refinements to achieve the desired outcome.

SECTION D – CHAPTER 2
USERS JOURNEYS

2.1 Introduction to Personas and UJM

To better understand these challenges and identify opportunities for improvement, this section introduces three distinct user personas, each at different stages of their careers and with varying levels of expertise in the field. Accompanying each persona is a User Journey Map (UJM), delineating the emotional highs and lows, actions, and specific pain points they experience while using circuit design platforms such as the ones we descrived in the previous section.

1, Lorenzo, the Interaction Designer: A moderately experienced user who works at an IoT startup. His challenges revolve around the component-centric workflow, limitations in real-world simulation, inadequate support for SMD technologies, and a dependency on physical prototyping.

2, Emily, The 3D Designer Venturing into Circuit Design: This persona represents users who are experts in the realm of 3D design but are novices in the field of circuit design. They are likely to face issues like steep learning curves, lack of foundational knowledge in electronics, and perhaps even software interface complexities that are different from what they are used to.

3,. Raj, the Seasoned Engineer Struggling with Rapid Technological Changes: Highly experienced but facing challenges due to the rapid evolution of technology. He might struggle with integrating new features into his already established workflow or keeping up-to-date with the latest component libraries and simulation technologies.

Each persona provides a lens through which we can examine the capabilities and limitations of current circuit design platforms. Through their User Journey Maps, we can pinpoint specific moments of friction and identify opportunities for enhancing user experience and functionality. By catering to the needs and pain points of these personas, developers and interface designers can aim to create more intuitive, efficient, and inclusive platforms for circuit design.

2.2 Persona and UJM of a medium experienced user

Lorenzo

Profession:

Interaction Designer

Background:

- Age: 30
- Occupation: Product Designer at a tech startup focused on IoT solutions
- Location: Turin, Italy
- Education: Master's degree in Design Engineering

Technical proficiency:

- Hardware: Custom-built PC, breadboards, SMD soldering station
- Software: TinkerCAD Circuits, Altium Designer, LTSpice
- Programming Skills: Python and JavaScript for basic scripting

Goals:

- To design and prototype IoT products with embedded SMD components.
- To minimize the time and resource expenditure in the design process.
- To collaborate effectively with engineers and other designers.

Pain Points with Current Solutions:

1. Component-Centric Workflow: TinkerCAD and similar tools focus on individual components, requiring Lorenzo to spend significant time selecting and understanding each component's specifications.

2. Lack of Real-World Simulation: Existing platforms like TinkerCAD offer limited simulation capabilities, making it difficult to predict how a circuit will behave under real-world conditions.

3. Inadequate Support for SMD Technologies: Lorenzo finds it challenging to simulate and prototype circuits using SMD components, as platforms like TinkerCAD often rely on traditional through-hole components for simulation.

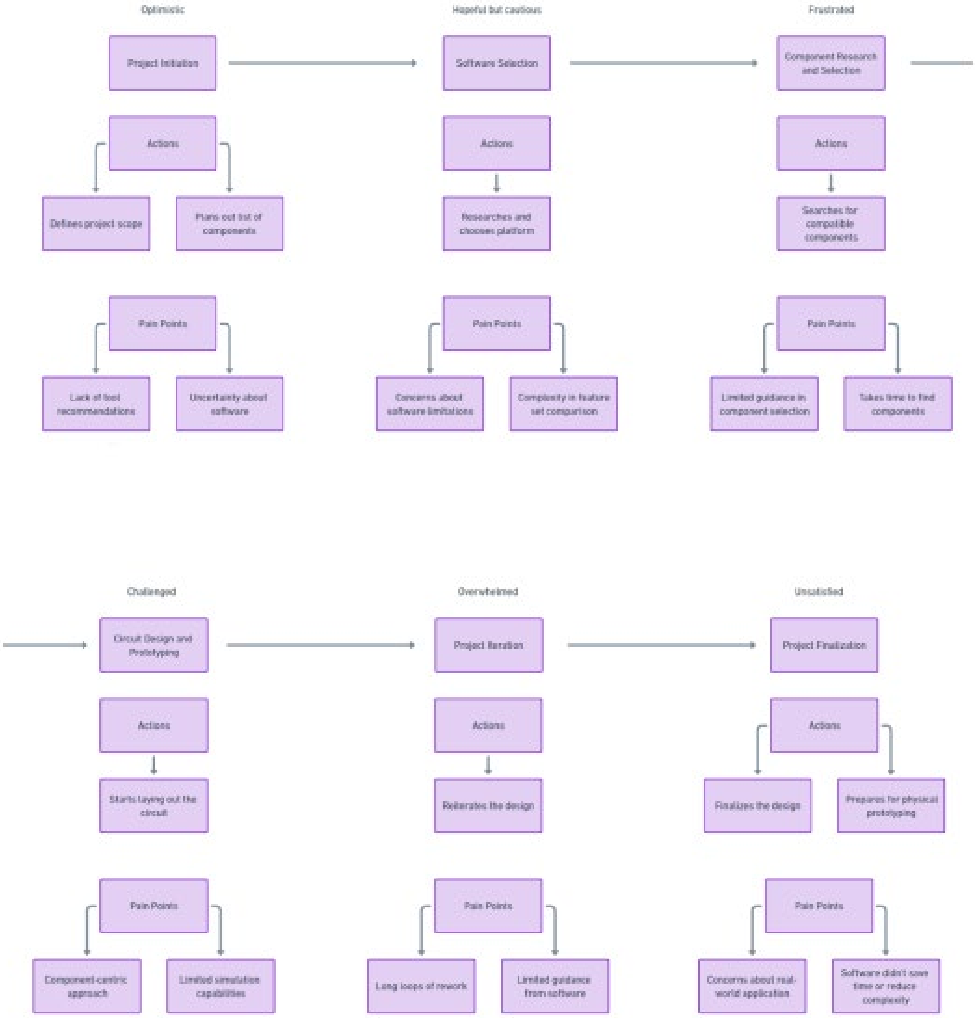
4. Dependency on Physical Prototyping: Even for initial validation, Lorenzo has to turn to Arduino or similar platforms, which delays the development process and increases costs.

Behavior:

Frequently cross-references between online component databases and TinkerCAD when selecting components and often redoes circuit designs multiple times due to poor component choices or inadequate simulations. Seeks advice from online forums when stuck, but usually finds the guidance insufficient for SMD projects.Takes a long time to select components on TinkerCAD due to the absence of an SMD-focused library, forcing him to consult external sources.

Finds it hard to collaborate with team members as the design keeps going through multiple iterations due to component changes. Abandons the virtual prototype at an early stage to shift to physical prototyping using Arduino, resulting in longer development times and higher costs.

Table 1 – *User Journey Map of Medium Experienced user*



2.3 Persona and UJM of a novice user

Emily

Profession:

3D Designer

Background:

- Age: 28
- Occupation: 3D Designer at a small design studio
- Location: New York, USA
- Education: Bachelor's degree in Industrial Design

Technical proficiency:

- Hardware: High-end Mac, 3D printer, basic breadboard for tinkering
- Software: Blender, SketchUp, limited experience with TinkerCAD Circuits
- Programming Skills: Limited to basic Arduino sketches

Goals:

1. To incorporate basic electronics into her 3D designs, making them more interactive.
2. To learn enough about circuit design to communicate effectively with electrical engineers on projects.
3. To find a design tool that offers a manageable learning curve for electronics.

Pain Points with Current Solutions:

1. Complexity and Overload: The component-centric nature of platforms like TinkerCAD is overwhelming for Emily, who has a limited understanding of electronics.

2. Limited Guidance and Support: Emily feels lost navigating through the plethora of components and functionalities on TinkerCAD, which assumes a significant level of user expertise.

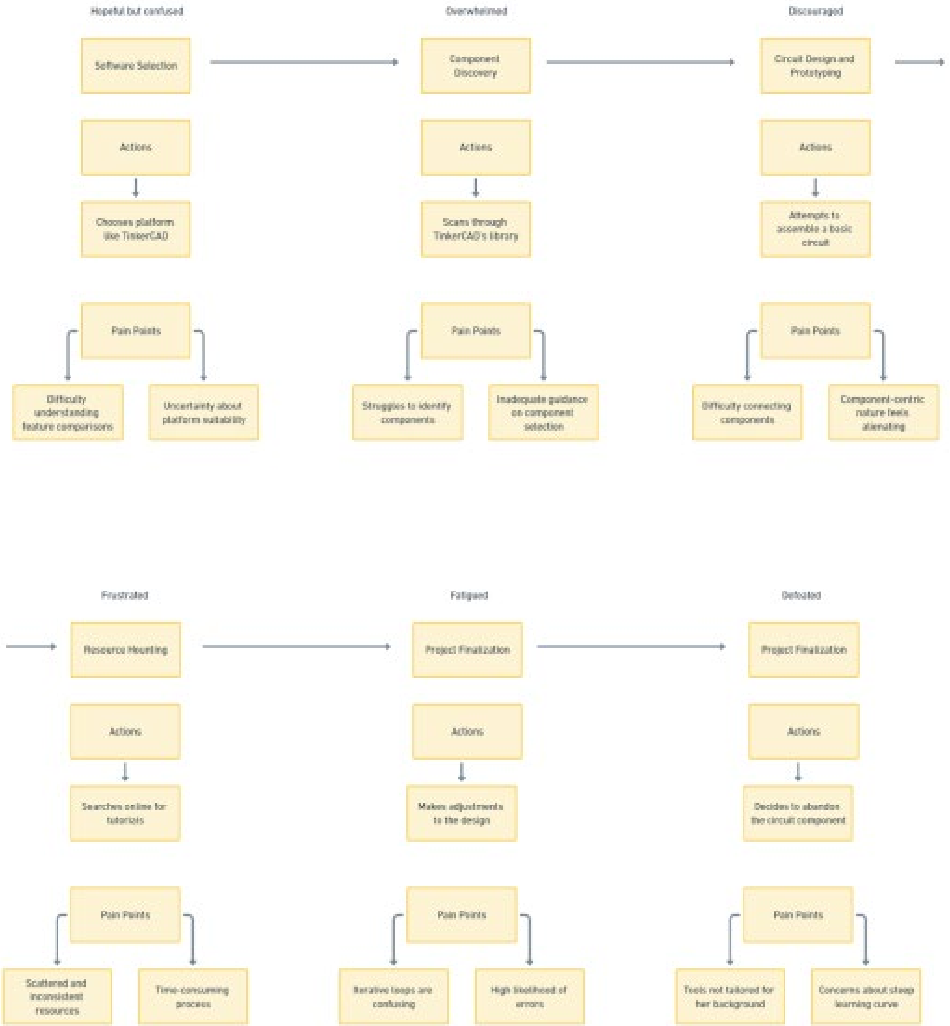
3. Poor Simulation for Beginners: The limited simulation capabilities of existing tools make it hard for her to understand if her designs will work in the real world.

4. Arduino Dependency: Her lack of knowledge makes even simple Arduino-based physical prototypes a challenge, delaying the design process.

Behaviour:

Often sticks to basic components in TinkerCAD due to a lack of understanding, limiting the complexity and functionality of her designs. Reads multiple tutorials and watches YouTube videos to make up for her lack of electronic design knowledge. Finds it easier to give up on adding electronics to her projects rather than dealing with the complexity of current design tools. Spends hours on tutorials to complete simple tasks like adding a basic LED circuit to her 3D designs. Gets frustrated when her circuit simulations fail and she can't diagnose the problems due to her limited understanding, until she gives up on the idea of virtual prototyping altogether, due to the steep learning curve, and sticks to her comfort zone of pure 3D design.

Table 2 – *User Journey Map of Novice user*



2.4 Persona and UJM of an experienced user

Raj

Profession:

Electronic engeneer

Background:

- Age: 28
- Occupation: Lead Electrical Engineer at an automotive tech company
- Location: Bangalore, India
- Education: Ph.D. in Electrical Engineering

Technical proficiency:

- Hardware: Fully-equipped electronics lab
- Software: Extensive experience with AutoCAD, Eagle, LTSpice, and TinkerCAD
- Programming Skills: C++, MATLAB, and embedded systems programming

Goals:

1. To streamline and modernize the design process in his department.
2. To keep abreast of emerging technologies in electronic design.
3. To mentor younger engineers in best practices for electronic design.

Pain Points with Current Solutions:

1 Keeping Up with New Components: Raj finds it difficult to keep up-to-date with the influx of new electronic components, which are not always readily available in platforms like TinkerCAD.

2. Limited Support for Emerging Tecnologies: He finds that tools like TinkerCAD are not quick to incorporate newer technologies like advanced SMD components, leaving him to resort to less efficient methods.

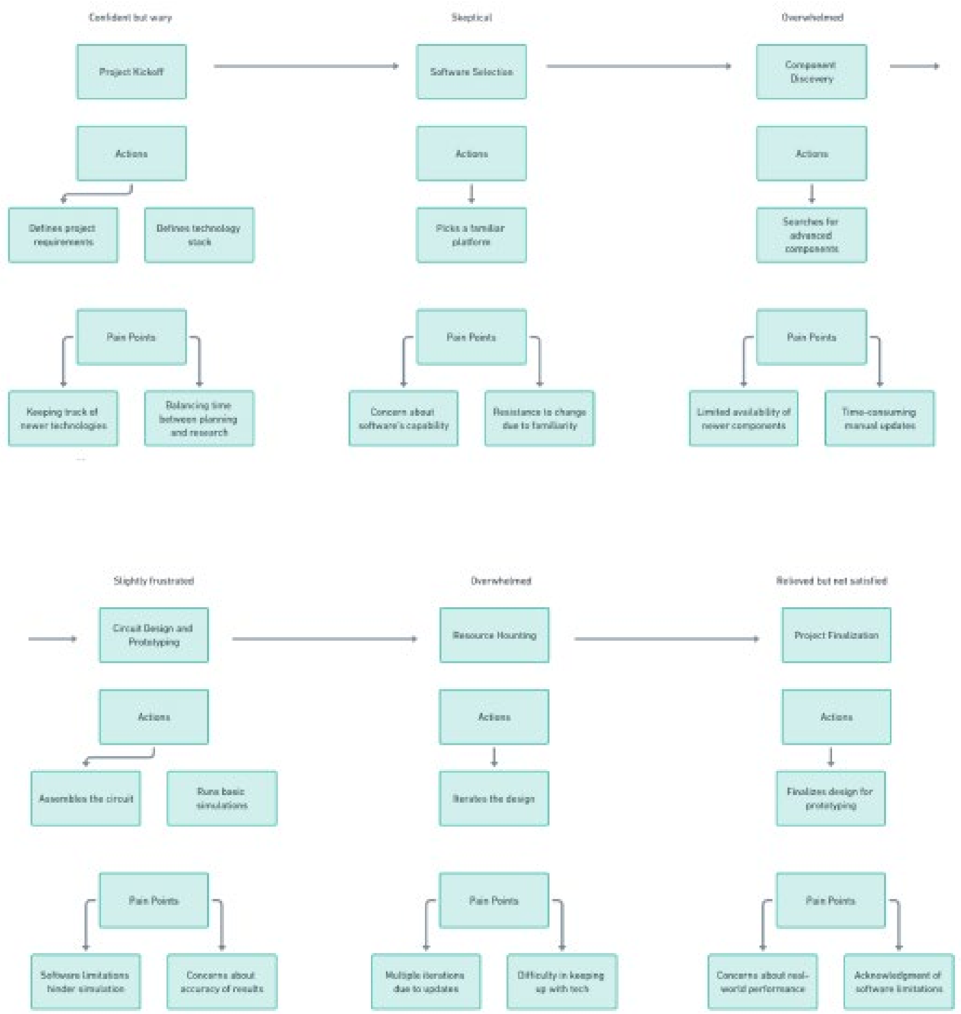
3. Workflow Inefficiency: The component-centric nature of existing platforms requires Raj to undergo multiple design iterations, leading to project delays.

4. Rigid Simulation Capabilities: Raj feels that the simulation functionalities in platforms like TinkerCAD do not sufficiently cover real-world scenarios for complex projects.

Behaviour:

Regularly attends webinars and reads research papers to keep up with new components and technologies. Resorts to creating custom component libraries but finds it a time-consuming process. Feels a growing disconnect between his extensive component knowledge and the tools available for modern design. Wastes time searching for updated component libraries to incorporate new technologies into his designs. Feels frustrated when he has to fall back on traditional components due to the limitations of tools like TinkerCAD. Often reverts to "old-school" prototyping methods because current tools don't offer adequate simulation for emerging technologies.

Table 3 - *User Journey Map of Experienced user*



2.5 Conclusions

This section aimed to investigate the pain points, behavior patterns, and technical limitations experienced by different user personas when engaging with electronic circuit design platforms like TinkerCAD. Three distinct personas—Lorenzo, Emily, and Raj—were studied in depth, revealing several overlapping and unique challenges:

Lack of Real-World Simulation: Across all personas, it was evident that existing platforms do not adequately simulate real-world conditions, resulting in design uncertainties.

Component-Centric Limitations: Is commonly seen that all the users, from the less to the more expert, have problems with the components approach. The more experts users may struggle with keeping up to latest technologies, while the novices may find this step overwhelming and give up or make significant errors, struggling to transform brief requirements into needed components.

Limited Guidance: Beginners like Emily require much more guidance than current platforms offer, while experts like Lorenzo and Raj require better integration with advanced components. Also the platforms had Workflow inefficiencies, all personas encountered several obstacles throughout their design journeys, with frequent iterations due to limitations in component libraries and simulation capabilities.

Need for External Resources: The journey maps for Emily and Lorenzo, and potentially Raj, suggest that users often have to leave the platform to search for additional guidance, tutorials, or component information. This indicates a gap in the in-platform resources and educational support offered by existing tools.

The findings from the user personas and journey maps reveal key areas that call for significant improvements in the existing electronic design platforms. One of the most striking needs is for a shift toward user experience design that is more nuanced and user-centric. The current one-size-fits-all approach falls short in catering to a user base with varying levels of expertise and needs, ranging from novices like Emily to experienced engineers like Raj. Instead, platforms could greatly benefit from tailoring features and workflows to better suit the diverse requirements of different user profiles. Simultaneously, there's a pressing need to keep pace with the rapid technological advancements in the field of electronics. The platforms must be agile enough to frequently update their component libraries and simulation functionalities to accommodate emerging technologies. This is particularly relevant for seasoned professionals like Raj, who find that existing platforms lack in this regard, rendering them less effective and increasingly obsolete. Lastly, the frequent need for users to seek external educational resources indicates a considerable gap in the onboarding and educational support offered by current platforms. Improved onboarding materials, tutorials, or even AI-based support could play a critical role in improving the user experience.

SECTION D – CHAPTER 3
DEVELOPING THE INFORMATION ARCHITECTURE

In the process of developing the information architecture (IA) for our digital platform, we undertook an analysis of several methodologies typically employed in the field of User Experience (UX) Design. Our goal was to determine the most suitable approach for our project. We first considered the Card Sorting technique, a common method utilized to inform the design of navigation and layout in the early stages of IA development. This technique involves participants categorizing different information elements into groups that align with their logical understanding. However, this approach did not prove to be the best fit for our specific needs. The structure of our platform was to be linear, guiding users through a sequence of pages towards a designated end goal. This stands in contrast to the more open, non-linear navigational structure that Card Sorting tends to facilitate. In particular, Card Sorting appeared less relevant to the design of our profile page and resources and settings page, which were common and non-central elements of our platform.

Given these considerations, we chose to employ User Flow diagrams and Tree testing for developing our IA. User Flow diagrams offer a visual representation of the path a user follows to complete a task. This technique allowed us to understand and ensure that our information was structured in a manner that supported the user's progression towards their goal. To further refine our IA, we developed three different architectural trees, drawing on our expertise and an 'expert view', then performed a task analysis on it. The tasks were made to understand if the users had an easy access to the various functions of the platform, especially to the main ones, which were highlighted as most important during the review of the user flows.

3.1 Preliminary user flow

Based on the insights obtained in the previous phase, we then proceeded to create a user journey that represented a unified version of those observed with the three different types of users in the use of this type of software. We noticed that when ideally tracing the streamline needed to move from the beginning to the end of a project, it does not outline the classic "funnel" approach that commonly characterizes the "component selection" method analyzed in chapter 2.3 of section A. Instead, there is a notable attention from users towards aspects of a more managerial nature and concerning more aspects of the product, rather than being technical and limited to only one part of it. Consequently, we outlined a user flow diagram which then proved useful for constructing the initial versions of the Information Architecture.

The essential user flow is shown in *Table 11, figure 1.*

3.2 Preliminary interview

3.2.1 Considerations before interviewing users

Our interview used both predetermined questions and open-ended exploration to gain more in-depth insights into participants' perspectives, attitudes, and experiences.
In order to gauge the necessity for a specified level of abstraction in design processes, independent of component selection, an interview was meticulously crafted. This instrument was envisioned to encapsulate the common challenges unearthed during the analytical phase, specifically within specialized communities. The emphasis was predominantly on user experiences and difficulties encountered during the component selection phase, with the aim of gleaning a deeper understanding of user needs from firsthand data.

The interview delved into various facets of the user experience, particularly honing in on the selection, sourcing, and utilization of electronic components in their projects. The dimensions explored included the frequency and extent of challenges faced due to component selection, the perceived level of difficulty in pinpointing the right components for a project, and instances where component conflicts or incompatibilities arose. Furthermore, it evaluated the time investment required in researching and selecting components, alongside the necessity for design alterations stemming from component-related issues. Moreover, the instrument sought to measure the level of frustration induced by component-related challenges, and the hurdles in attaining optimal performance through the conventional component-import-based model. It also examined the cost implications and communication challenges tied to component selection, as well as the perceived constraints on creativity and design flexibility due to a component-centric approach. The interview also ventured into the realm of online component sourcing, gauging habits and experiences in this domain. It assessed the importance respondents placed on having access to a wide array of components, the influence of component costs on design decisions, and the level of confidence in the quality of components sourced online. Lastly, it evaluated the significance attributed to having detailed resources and documentation for the components utilized. The structured part of this interview, comprising 20 questions, was disseminated to a cohort of 45 users, all of whom were either Designers or Engineers.

3.2.2 Conclusions from interview

This study validated the assumptions made in the concept, as users showed feedback in line with what was anticipated from the research phase: during this preliminary interview, it clearly emerged that choosing the right component for an electronics project is a very challenging task. The frequency with which issues related to the initial choice of an unsuitable component are encountered was found to be quite high. There were frequent situations where the selected components caused conflicts or incompatibilities in the project. The levels of frustration are high due to delays or blockages in the design process caused by component-related problems. A significant number of users had to make major changes to their projects after changing one of the components. The component-based approach proved to be limiting for potential group work and for the creativity of individual users. The issues related to component choice had significant repercussions on the development costs of the projects. Moreover, it was found that almost all designers look online for information regarding individual components, spending a significant amount of time on this research.

3.3 Tree testing analysis

As first, we developed some tasks for our users to perform. The tasks were made to understand if the users had an easy access to the various functions of the platform, especially to the main ones, which were highlighted as most important during the review of the user flow. The task instructions that we used avoid using terms that give away the answers. The outputs of this testing were:

3.3.1 Tracked data

The tracked data was Sueccess Rate, Directness and Time spent. Success rate is the percentage of users who found the right category for that task. The success rate for that task indicates the percentage of users who found the correct location in the tree and identified it as the right place to complete that task. Any trials in which users selected a different final location are reported as failures. The Directness percentage of users who went to the right category immediately, without backtracking or trying any other categories; Time spent was equivalent to the average amount of time elapsed from the beginning to the end of the task;

3.2.3 Tasks Submitted

1) Find the section where you can edit your user_name (Profile)
2) Find the informations regarding the electrical connections of a project of yours (Electric circuit)
3) Create a new project (New Project)
4) Add a new action for the system to perform (Home / Landing)
5) Add details regarding an action of the system and its relation with the others (Define Goals / Actions Page)
6) See the components footprint (Phisical Layout)

7) See the bill of materials of a project from the community (Community – BOM)

3.3.3 Developed Trees

Tree number one [*Table 5, Figure 1*]:

In the first system outlined, the Menu is presented as an all-inclusive Dashboard of all other sections not directly related to the development of a new prototype. Resources are not grouped together and can be accessed in parallel from the Menu page. The three sections "Bill of Materials," "Electric circuit," and "Phisical Layout" are grouped and accessed from the Examine and Modify page as well as being the natural output of Project Building.

Tree number two [*Table 5, Figure 2*]

In the second system outlined, there is no dashboard or general menu, but rather all pages are accessible in parallel. Resources are grouped together and can be reached only by clicking on the "resources" page. The three sections "Bill of Materials," "Electric circuit," and "Phisical Layout" are not grouped but rather accessed in parallel with each other from the Open Project page and the Project Building.

Tree number tree [*Table 5, Figure 3*]

In the third system outlined, the "Open Project, New Project" page acts as an access menu for sections not directly related to the development of a new prototype. Resources are grouped together and can be reached only by clicking on the "resources" page. This results in having the benefits of a Dashboard but more order than the first wireframe. The three sections "Bill of Materials," "Electric circuit," and "Phisical Layout" are grouped and accessible from the Examine and Modify page and the Project Building.

3.3.4 Test results;

Graphs related to the results of these tests are shown in *Table 5, Figure 4.*

In the first scenario, the success rate stood at 78.5%, with a directness of 69% and an average time of 71.25 seconds. Moving on to the second scenario, there was an improvement in the success rate which reached 83%, and a notable increase in directness to 86.5%, all while reducing the average time to 52.5 seconds. In the final scenario, the success rate climbed further to 87%, with a directness of 87.25% and a slight increase in average time to 55 seconds. When comparing the groups, a clear trend emerges: while the first group tends to be slower, the other two show greater time efficiency. This efficiency does not seem to compromise quality, as directness, an indicator of accuracy, remains high in the later groups. The second group represents a balance, with clear improvements over the first, but does not reach the efficiency of the third. In fact, the latter combines fast times with high directricity, emerging as the best performer. The progression suggests that, with experience or optimization, performance can improve significantly. In summary, while the first group is inferior, the other two indicate a promising direction for maximizing performance.

SECTION D – CHAPTER 4
WIREFRAME DEVELOPING

Our journey in the wireframe development phase was deeply rooted in an iterative methodology. From the outset, we recognized improvements step by step in a gradual process, achieved through layers of refinement and feedback. In total, we developed three distinct wireframes, each iteration crafted with the insights and lessons gleaned from its predecessor.

The initial wireframe served as the foundational layout, offering a basic visual guide that encapsulated the core pages and features of our envisioned platform. But as foundational as it was, we knew it was just the beginning. We conducted a contextual inquiry to gain feedback and insights from this phase laid the groundwork for the subsequent iterations, each more detailed and nuanced than the last.

Our second wireframe, though a refined version of the first, was developed with an added emphasis on design elements and interactions. It was here that the intricacies of methodologies like Task Analysis and the System Usability Scale (S.U.S.) score came into play, offering depth to our evaluation process.

In our third and final iteration, the wireframe took a transformative turn. The design, while retaining its core elements, introduced a new approach to the dichiarative section. This shift allowed us to cater to users desires to explore alternative technologies, making the platform more versatile and user-centric. Therefore, we moved to developing the final interface.

In essence, our wireframing journey was not just about creating visual representations; it was an exercise in iteration, feedback, and continuous refinement.

With each wireframe, we moved closer to our goal, ensuring that when we finally reached our prototype stage, it was the crystallized product of meticulous planning and user-centric design.

4.1 First Wireframe

4.1.1 Considerations in building and methodology

The first wireframe is shown in *Table 6, figure 1,2,3.*.
The first wireframe is a rudimentary sketch of a project's design. At this initial stage, it was build to allow the evaluation of structure and therefore of how elements are organized and their spatial relationship. Then hierarchy: the prominence and sequence of information or components. Lastly it was used to evaluate the flow and the interaction points.

This first wireframe was developed to show the main functions of the interface, and therefore the Login page, Profile, Projects from the user and the community, Set goals and Developing of the system (here tried with a nodal system), Phisical Layout, Electrical Circuit and Bill of Materials. The model is showing the arrangement of elements on the page, giving an idea of the hierarchy and is useful to understand how users will move through the app. It already shows where primary and secondary call to actions like buttons will be placed. This is crucial for guiding user actions and achieving desired outcomes.

Then, we proceeded to conduct a contextual inquiry. This inquiry further enriched our understanding of the user interactions and expectations. Through observing and engaging with users within the contextual setting of the designed interface, we garnered valuable insights that are instrumental in refining the wireframe. The feedback obtained from the contextual inquiry is pivotal, providing a more nuanced comprehension of user behaviors and preferences, thereby enabling a more user-centric refinement of the project's design as we advance to subsequent stages of development.

4.1.2 Considerations after user observation:

The user flow is shown in *Table 6, figure 4.*
The navigation between the various phases was found to be clear by the users, wich quickly reached all the sections that they were looking for. This proved our information architecture and the information tree that we developed to be efficient.

Unfortunately, tho, the nodal system, despite being really relevant for the users (that in this wireframe we simulated thanks to reactive circles of the same color) was found not really intuitive. Especially the intermediate connection nodes between inputs and outputs, like “math” and “logic”, were found to be counter-intuitive for the users, wich in this step should just care about the connections between the elements without detailing it. Also, the “Set Goals” page seemed reduntant and detached from the following pages.

From the user comments also emerged the need for additional pages related to the community where it would be possible to share meetups, workshops and in gerneral things that are not directly linkable with a single electronic project. While, about the projects request from the users was to to add hashtags and filters to projects to easily help in finding similar ones and for a general order in the pages.

The users also addressed the fact that the first page, the “home” or “dashboard”, could have been amplied with other elements and links to other pages that could probably save time to the usage.

4.2 Second Wireframe

4.2.1 Considerations in building and methodology

The second wireframe is shown in *Table 7, figure 1,2,3*

Within this wireframe, we re-designed the “Set Goals” pages, concentrating everything in just one canva (the nodal one) and therefore in one page, adding pop-up menus. Also other functions, like changing the colors of the nodes and so on,

were added to this page. Within this wireframe, we added a “Blog” and “Practical Projects” to the Resources section, and the #hashtags to the projects in order for them to be easily filtered and located. The “home” landing page was modified in order to quickly access older projects and also new projects from the community, that can be inspirational for the final user.
We then performed a task analysis and S.U.S. on the users.

4.2.2 Tasks submitted to the users:

1) Find the section where you can add or edit your location -> correct page: Profile
2) Find and download the product schematic of a project of your choice -> correct page: Electrical Circuit
3) Add a "PLAY SOUND" component to your system -> correct page: Nodes, Component Menu
4) Add a "TURN KNOB" component to your system, then connect its output with the output of a "TRACKING" component and finally "DISPLAY". -> correct: Nodes, Component menu and then several actions on it
5) Find and download the BILL of MATERIALS of the project. -> correct page: Nodes -> correct page: Bill of Materials
6) Find a way to get audio from a SD Card. -> correct page: Nodes, Component menu

4.2.3 Considerations after testing:

The second wireframe tests results are shown in *Table 9, figure 1,2.*
First, we noticed how a S.U.S. Average score of 50 signifies a lacking experience from the users. The platform needs detailing to enrich the experience and to make the use of the program more realistic and therefore the testing more accurate, this was something related to all the pages, from the Profile to the Blog and Community pages. Then, the section of the interface for building the system seemed to be too covered with Ui elements and the users were not able to pan and tilt on the canva with efficancy. Also, a very important issue emerged with the test was -still- the lack of clearance of the user for the use of the nodal system in general to establish

relationships between the elements in the canva. The second wireframe was already showing a refined version, but problems emerged mostly regarding the Nodal System. The users were interested to the function but confused mainly regardingfunctional aspects of it. Lastly, users felt restricted or limited by the predefined components and technologies offered by the platform. They were unable to customize or optimize their projects according to their specific needs or preferences, especially when they wanted to use alternative components or technologies that might be more suitable or advanced than the ones originally proposed. This limitation could have hindered creativity, innovation, or the ability to achieve the best possible outcomes in their projects. The inability to explore and integrate alternative options in the physical layout page made the platform less flexible and adaptable to diverse user requirements.

4.3 Third Wireframe

4.3.1 Considerations in building and methodology

The third wireframe is shown in *Table 8, figure 1,2,3*

Within this wireframe, the “BUILD SYSTEM” Canva, wich in the previous versions were still a Nodal System and causing confusion to the users, were changed preferring a functional or dichiarative approach. Therefore, an important change made at this stage of the design was the integration of the “Setup Wizard”. This was made because by guiding users through the setup process step-by-step, there's a lower chance they'll make mistakes, and as users progress through the wizard, they can learn about the important needs when using certain types of components and how to use them. we detailed a lot the interface under the UI point of view. It was not meant to be the final one, but it is showing a lot of improvements from the previous one. Then, the Profile Page was further enriched with the possibility for the user to obtain badges and add more details about him/her like the location and contacts. Lastly, but not less important, when users complete the setup trough a guided process, they often feel more confident in using the application. They're reassured that they've set things up correctly, which can lead to increased trust and satisfaction. Another important change that was made at this stage was the possibility for the users to explore alternative components and technologies from the ones that are originally proposed by the platform. This is all done in the phisical layout page.We then performed a task analysis and S.U.S. on the users.

4.3.2 Tasks submitted to the users

1) Find the section where you can add or edit your location -> Profile
2) Find and download the product schematic of a project of your choice -> correct page: Electrical Circuit
3) Add a "PLAY SOUND" component to your system -> correct page: Actions
4) Add a "TURN KNOB" component to your system, then specify it needs to be waterproof -> correct: Actions, Action Details
5) Find and download the BILL of MATERIALS of the project. -> correct page: Bill of Materials
6) Find a way to get audio from a SD Card. -> correct page: Actions
7) Change a component of any project of your choice with a component using a similar technology. -> correct page: Phisical Layout, Component tap and then several actions on it
8) Specify somewhere in the application that your device is supposed to be really thin, precisely 12x12x5 cm maximum -> correct page: Actions, General system detailssystem details

4.3.3 Considerations after testing

The second wireframe tests results are shown in *Table 9, figure 3,4.*
After testing this wireframe, and comparing the results swith the previous tests that we made, some things emerged:

First of all, the hybrid GUI dichiarative system that we created, wich is the "setup wizard" was showing a lot of improvement respect to the nodal system that we used in the previous versions. This was clear thanks to user comments and also visible with the data coming from the tasks 3,4. While in the previous wireframe tasks having the same goal had an error rate ranging from 20% to 29% and a low User

Rating, Within this wireframe the error rate for these tasks was between 8% and 11%, with higher Ease of Use Ratings. The possibility for the users to explore alternative components and technologies from the ones that are originally proposed by the platform was a feature liked by the users. The task related with the achievement of this goal, the task number 7, is still showing the highest error rate, but this was mainly due to to miscomprehensions in the UI of the interface.

In general, the system achieved a S.U.S. of 75, wich is already quite enough for a prototype that is not yet the high fidelity one. Therefore, we choose to keep this wireframe as guide to work to the final prototype interface.

SECTION D – CHAPTER 5
FINAL INTERFACE TESTING

5.1 Methodology

For the final testing of our interface, we deployed a multi-pronged approach comprising Task Analysis, Questionnaires, and Heatmaps. Each of these tools has its unique strengths that provide invaluable insights into user behavior and interaction with our interface:

- Task Analysis is indispensable for gauging the efficiency, effectiveness, and satisfaction with which users can achieve specific goals in our interface. It allows us to understand the user's path and to identify any potential obstacles or friction points they might encounter.

- User Interview provides a direct channel to gather users' feedback, perceptions, and feelings regarding their experience. This qualitative data is essential to understand the user's subjective experience, ensuring that not only is our interface functionally sound but also emotionally resonant.

- Heatmaps visually represent where users click, move, and scroll on the platform. They offer an immediate visual representation of user behavior and show which parts of the interface engage users the most and which might be overlooked.

By integrating these methods, we aimed to secure a holistic understanding of the user's journey. The ultimate goal of this final testing phase is not merely to identify issues but to validate that our solution meets user needs and aligns with our initial objectives. This comprehensive evaluation ensures that the final interface is not only intuitive and user-friendly but also provides value and achieves its intended purpose.

5.2 Task analysis

5.2.1 Preliminary considerations

This task analysis aims to delve into these subtleties by focusing on two key areas. First, it assesses how intuitively users can comprehend the system's functionalities and features. A well-designed system should be almost self-explanatory, minimizing the cognitive load required for users to understand how it works. Second, the task analysis explores the ease with which users can navigate through the system to achieve their objectives. A user-friendly navigation not only enhances user satisfaction but also contributes to the system's overall effectiveness and efficiency. To gather insights into these aspects, a series of tasks will be submitted to users. These tasks are crafted to simulate real-world scenarios that a typical user might face while interacting with the system. The results will offer valuable perspectives on where the system excels and where it may need improvements. This testing has been performed over 35 users. They were all product designers and production engineers, M/F, with an age ranging from 24 to 50 years old.

5.2.2 Tasks submitted to users

1) Find the section where you can add or edit your location -> Profile
2) Find and download the product schematic of a project of your choice -> Electrical Circuit
3) Add a "PLAY SOUND" component to your system -> System Building
4) Add a "TURN KNOB" component to your system, then note that Durability is a key feature for this action -> System Building, System building Details
5) Find and download the BILL of MATERIALS of the project. -> Bill of Materials
6) Find a way to get audio from a SD Card. -> System Building
7) Change a component of any project of your choice with a component using a similar technology. -> Phisical Layout, Comparison Menu
8) Specify somewhere in the application that your device is supposed to be really thin, precisely 12x12x5 cm maximum -> Wizard (details) pages.

5.2.3 Conclusions

The results from the task analysis are shown in *Table 10, figure 1.*

In general the tasks were well-performed and the results are totally satisfacient. the user take less time to perform the tasks than the previous wireframe, wich was already satisfacient. Not only, te users committ very few errors and find the platform easy to use.

5.3 Heatmaps

5.3.1 Preliminary considerations

Our purpose with our heatmaps was to serve as a pragmatic tool for delving into the user interactions within the prototype. Specifically, they aimed to unveil patterns in user behavior, validate the design decisions taken during the development phase, and provide insights into the navigation efficiency of the platform. By scrutinizing where users directed their attention and how they traversed through various elements, we sought to evaluate the effectiveness of the current design. Our Heatmaps encompassed Click Heatmaps and Move Heatmaps, thereby capturing both these actions. The depicted results, representing an average, encapsulate the interactions of our entire user base.

5.3.2 Conclusions

The results from the heatmaps testings are shown in *Table 10, figure 2, 3, 4*.

The Move + Click heatmap analysis has provided key insights into user behavior and interaction patterns on the tested interface. Across the diverse range of pages, having each its unique UI and GUI characteristics due to the fact that the interface has different steps with different GUI needs, there was a notable consistency in how users navigated and interacted. This uniformity, especially in the face of varied design elements, indicates a successful translation of design intentions into actual user experiences. The seamless user interaction with various design components, be it sliders, buttons, or other interactive elements, underscores the clarity and comprehensibility of the design approach adopted. Such predictability in user behavior reflects a design that aligns well with user expectations and intuitive navigation patterns. One of the standout achievements of this design was its alignment with the defined user interaction goals. Every element on the interface seemed to guide the user predictably, ensuring that the user's journey through the website was both intuitive and in line with the site's objectives. The absence of any unexpected behaviors further accentuates the user-centric focus of the design, eliminating potential points of confusion or misdirection. The reliability and generalizability of these findings are enhanced by the tool's capability to average out the interactions of all users. This consistency ensures that the insights drawn are not mere anomalies but are representative of the broader user base.

In summation, the heatmap analysis offers a clear endorsement of the user interface design developed in this thesis. It reinforces the notion that a well-conceived, user-focused design can lead to predictable and goal-aligned user interactions, thereby optimizing the overall user experience.

5.4 Interview

5.4.1 Preliminary considerations

5.4.2 Pre-defined questions of the semi-structured interview

1. Can you describe your experience in understanding the purpose and functionalities of the system?
2. What are your thoughts on the innovative and usefulness of the approach proposed in the node page?
3. How likely are you to use this system for your future projects and why?
4. What are your impressions of the visual design of the interface?
5. Can you comment on the clarity of the Graphical Aspect and how it affected your experience?
6. How did the system's design impact your ability to focus on the required actions rather than the discrete components?
7. How easy or difficult was it for you to add specific requirements to the system concerning one or all the actions and components? Can you elaborate on any challenges or facilitators you encountered?

8. How helpful did you find the informative elements provided within the system?
9. Did you feel overwhelmed by the information provided? If so, can you describe what elements were overwhelming and why?
10. Did you feel the need to consult other platforms or surf the web to clarify doubts while using the system? Can you share examples?
11. How useful did you find the feature of having recommended different alternatives once the system is generated?
12. Can you share your experience navigating through the Bill of Materials and Electrical Circuit pages? Were they intuitive?
13. In your opinion, how suitable is this system for novices and beginners? Can you elaborate?
14. Similarly, how suitable do you think this system is for experts and professionals?
15. How well do you think the system integrates with other tools or platforms you use? Can you provide examples?
16. How satisfied are you with the speed and responsiveness of the system? Can you share any specific incidents or experiences?
17. Which features of the system were most intuitive to you and why?
18. What would make you more willing to use this system regularly?
19. Which features of the interface did you find most useful?
20. Do you have any other comments or suggestions for improving the system?

5.4.3 Conclusions

The qualitative insights from the semi-structured interviews offer a significant understanding of the system's strengths. Participants found it relatively easy to grasp the purpose and functionalities of the system, reflecting a balanced level of comprehension. They also appreciated the innovative approach proposed on the actions page and expressed a strong inclination towards utilizing this system for their future projects, feeling assisted by the interface and not finding it difficult to use but almost “natural”.

The visual design of the interface received mixed feedback, indicating there might be room for enhancement to make it more appealing or user-friendly. On the other hand, participants highly commended the clarity of the graphical aspect, which aligns with the ease of understanding the system's purpose and functionalities. These aspects collectively contribute to a positive user experience, underscoring the system's effective design in conveying information visually.

Notably, participants did not feel overwhelmed by the information provided, which suggests the amount of information presented was manageable and did not overwhelm the users. This is a positive indicator of the system's user-centered design, ensuring that users can navigate and interact with the system without feeling inundated.

On a broader spectrum, the general positive feedback for most aspects, except for visual design, showcases a high level of satisfaction and acceptance among the participants towards the system. This feedback is instrumental in understanding the system's efficacy and the potential it holds for aiding individuals in their projects.

SECTION D - CHAPTER 6
CONCLUSIONS FROM DEVELOPING PROCESS

The initial validation of our idea was achieved through iterative testing. These tests were critical in confirming the assumptions we had made during the conceptualization phase. Feedback from the initial testings showed that users experienced challenges related to the component selection phase, which was in line with our research expectations.

Furthermore, we undertook testing on the trees to validate our chosen information architecture (IA). Engaging with real users allowed us to ensure that the platform was in sync with their expectations, needs, and preferences. As part of our design approach, we created wireframes, which serve as visual skeletons of a digital platform, outlining its structure and illustrating the interrelation of its elements.

Our initial wireframe provided a foundational layout and flow, acting as a rudimentary visual guide. This allowed stakeholders to grasp and discuss the project's trajectory, showcasing all primary pages and features. We subsequently conducted user observation and a page relevance test on this wireframe. These tests proved invaluable in pinpointing areas for refinement.

Our second wireframe exhibited a more polished design, showcasing enhanced elements and interactions. Despite these advancements, our evaluations highlighted the need for more intricate detailing to enrich user experience and enhance the platform's realism, hence rendering the tests more precise. Most issues cropped up concerning the general UI and the Nodal System. While users were intrigued by its functionality, they were often perplexed about its operational facets. This wireframe was rigorously tested using Task Analysis and the S.U.S. score.

The third wireframe differentiated from the previous primarily due to the modification of the Nodal System, where we transitioned to a Declarative Approach. This shift necessitated the integration of a comprehensive "Wizard" section. An added enhancement was enabling users to explore alternate technologies and solutions, divergent from the platform's primary recommendations. The results from testing this wireframe were satisfactory, propelling us to the final prototype stage.

Our final testing phase corroborated our design choices. Task analysis results were impressive; users executed tasks more promptly than with earlier wireframes and committed fewer mistakes, finding the platform easy to navigate. Our interview insights revealed a strong user affinity for our platform. Predominantly, users showed a preference for the step-by-step declarative approach, which was our research's focal point. Users also commended the system's design, which permitted them to specify as planned. Even though a few felt constrained by the demo's functionalities, the majority expressed eagerness to utilize the full-fledged system. The design's clean visuals, interactive pop-up menus, and community elements, combined with its intuitive nature, catered to both novices and seasoned users.
The heatmap analysis furnished us with crucial insights into user behavior and interaction on our interface. We observed consistent user navigation, indicating that our design aspirations translated successfully into tangible user experiences. A hallmark of this design was its alignment with predetermined user interaction objectives, registering predictability in user behavior and a design well in tune with user expectations and intuitive navigation trends.

SECTION E - CONCLUSIONS

In synthesizing the findings and proposed solutions from the preceding chapters, several pivotal conclusions emerge which underscore the exigency of a paradigm shift in the realm of embedded technology project design. The research emphatically points to a systemic challenge characterized by a rigidity and high cost associated with the prevalent component-based approach. This approach not only steepens the learning curve but also exacerbates the resource constraints in terms of time, financial input, and specialized knowledge, particularly among diverse user backgrounds.

The identified shortfalls in existing hardware and software solutions, as evidenced by the dearth of recognized educational resources and the frequent recourse to external resources, accentuate the necessity for a more intuitive, resource-rich, and user-centric platform. The proposed One-Stop Platform heralds a significant stride towards alleviating these challenges. By adopting a Hybrid Approach that emphasizes system functionalities over individual components, the platform potentially democratizes the design process, catering to both novices and seasoned designers. This approach not only simplifies the initial stages of design but also significantly reduces the iterative loops and the associated frustration and resource drain.

The platform's feature set, which includes a more user-centric design interface, an accurate physical simulation capability, and a seamless integration of community support, further bolsters its potential to bridge the existing knowledge gap. Furthermore, the continuous adaptation feature of the platform, facilitated by a higher level of automation, ensures that the project remains within the user-defined bounds even when modifications are made, thereby reducing the likelihood of errors and the need for extensive re-iterations.

Moreover, the platform’s ability to offer an exhaustive bill of materials and electrical circuit layout, coupled with its feature that enables users to easily navigate between similar technologies and components, is poised to significantly enhance the workflow. This is not only advantageous in terms of cost and resources but also alleviates the frequent feelings of frustration and stress associated with the current design process.

Lastly, the platform's innovative market positioning, which endeavors to meld the precision and advanced capabilities of competitors software like Altium with the user-friendly interface of programs like Tinkercad, without falling into the pitfalls of module and breakout board reliance as seen in Arduino, underscores its potential to fill the identified market gap. By leveraging AI-driven functions, the platform aspires to streamline the design process to a level of precision previously deemed unattainable.

Building on the robust foundation of research and iterative testing, the journey from conceptualization to final validation emerges as a meticulous endeavor, attesting to the rigor and the user-centric ethos underpinning this project. The initial validation was a pivotal phase, shedding light on real-world user challenges, particularly around the component selection phase, resonating with the identified research pain points. The subsequent engagement in iterative testing, encompassing tree testing to validate the chosen information architecture (IA) and wireframing to visualize and refine the platform's structure and interrelations, showcased a commitment to grounding the platform in user expectations, needs, and preferences.

In summation, the research and the consequent proposal of the One-Stop Platform underscore a clear and urgent call to action for developing holistic solutions that obliterate existing barriers, enrich the resource pool, and foster a more collaborative, flexible, and user-centric ecosystem in both hardware and software domains. This initiative not only holds promise for catalyzing a more intuitive and supportive design environment but also for significantly accelerating the development process, ultimately driving a more efficient and less resource-intensive pathway to realistic prototyping and product development in the field of embedded technology.

BIBLIOGRAFIA

[1] Tripathi, S. L. and Dwivedi, S. (2022) "Electronic Devices and Circuit Design." 1st edn. Apple Academic Press.

[2] Rafiquzzaman M. (2011) "Microcontroller Theory and Applications with the PIC18F." 1st edn. Wiley Publishing.

[3] Güven Y., Coşgun E., Kocaoğlu S., Gezici H., Yilmazlar E. (2017) "Understanding the concept of microcontroller based systems to choose the best hardware for applications." Res. Inventy Int. J. Eng. Sci., 7 (38).

[4] Kiruthika, J. and Dhandapani, A. (2016) "Making Washing Machines Smart through IoT." International Journal of Modern Trends in Engineering and Science, 3, pp.39-41.

[5] Farion, C. (2022) "The Ultimate Guide to Informed Wearable Technology." 1st edn. Packt Publishing.

[6] Raad, H. (2020) "Fundamentals of IoT and Wearable Technology Design." 1st edn. Wiley.

[7] Global Market Insights (2023) "Embedded System Market Share | Global Report, 2023-2032." Available at: www.gminsights.com.

[8] ResearchGate (2023) "Embedded Systems Market Size, Share & Growth Report 2023." Available at: www.researchgate.net.

[9] Zhang, J. (2018) "An Ebd Approach To Embedded Product Design." 1st edn. LAP LAMBERT Academic Publishing.

[10] Johannes Hinckeldeyn, Rob Dekkers, Nils Altfeld, Jochen Kreutzfeldt (2014) "Expanding bottleneck management from manufacturing to product design and engineering processes." Computers & Industrial Engineering, Volume 76, Pages 415-428.

[11] Fowler, K. (2014) "Developing and Managing Embedded Systems and Products." Elsevier Science. Chapter 10 - Electronic Design.

[12] Pries, K. and Quigley, J. (2008) "Project Management of Complex and Embedded Systems." 1st edn. CRC Press.

[13] Greenfield, Adam (2006) "Everyware: The Dawning Age of Ubiquitous Computing." New Riders.

[14] Pries, K. and Quigley, J. (2008) "Project Management of Complex and Embedded Systems." 1st edn. CRC Press.

[15] Tennant, D. (2022) "Product Development." 1st edn. Wiley.

[16] Greenfield, Adam (2006) "Everyware: The Dawning Age of Ubiquitous Computing." New Riders.

[17] Zhang, J. (2018) "An Ebd Approach To Embedded Product Design." 1st edn. LAP LAMBERT Academic Publishing.

[18] Skibińska, J., Saafi, S., Pascacio, P., ... & Lohan, E. S. (2021) "A survey on wearable technology: History, state-of-the-art, and current challenges." Computer Networks, 193, 108074.

[19] Page, T. (2010) "A Methodology for Decision-Support in Electronic Product Design." 1st edn. LAP LAMBERT Academic Publishing.

[20] Liu, A. C.-C. and Law, O. M. K. (2021) "Artificial Intelligence Hardware Design." 1st edn. Wiley.

[21] Kalkan Okur, E., Yesevi Okur, F., and Altunişik, A.C. (2018) "Applications and usability of parametric modeling." Journal of Contemporary Engineering and Management Innovations.

[22] Stevek, J., Kvasnica, M., Fikar, M. and Gomola, A. (2017) "A Parametric Programming Approach to Automated Integrated Circuit Design." IEEE Transactions on Control Systems Technology, [online] (99), pp.1-12.

[23] Shabnam Gholamshahi, Seyed Mohammad Hossein Hasheminejad (2018) "Software component identification and selection: A research review." First published: 31 October 2018.

[24] Byoung Kyu Choi and Donghun Kang (2013) "State-Based Modeling and Simulation." In: Modeling and Simulation of Discrete-Event Systems, Wiley Online Library, Chapter 9.

[25] Shelley Gretlein (2013) "Software Modeling for Embedded Systems." In: Robert Oshana and Mark Kraeling, ed., Software Engineering for Embedded Systems, Newnes.

[26] IMT (n.d.) "Event-based software architecture in embedded system development." [online] Available at: https://www.imt.ch.

[27] Bobrow, D. (1984) "A Layered Approach to Software Design." Interact Program Environmen.

[28] Shelley Gretlein (2013) "Software Modeling for Embedded Systems." In: Robert Oshana and Mark Kraeling, ed., Software Engineering for Embedded Systems, Newnes.

[29] G. Karsai, F. Massacci, L. Osterweil and I. Schieferdecker (2010) "Evolving Embedded Systems." Computer, vol. 43, no. 5, pp. 34-40, May 2010, doi: 10.1109/MC.2010.135.

[30] Johannes Deichmann (2022) "Cracking the complexity code in embedded systems development." March 25, 2022 | Article.

[31] Kaisti, M., Rantala, V., Mujunen, T. et al. (2013) "Agile methods for embedded systems development - a literature review and a mapping study." J Embedded Systems, 15 (2013).

[32] Klaassen, R.G. (2018) "Interdisciplinary education: a case study." European Journal of Engineering Education, 43(6), pp.842-859.

[33] Salazar, M.R., Lant, T.K., Salas, E. et al. (2012) "Facilitating Innovation in Diverse Science Teams Through Integrative Capacity." Small Group Research, 43(5).

[34] Vestal, A. and Mesmer-Magnus, J. (2020) "Interdisciplinarity and Team Innovation: The Role of Team Experiential and Relational Resources." Small Group Research.

[35] Tennant, D. (2022) "Product Development." 1st edn. Wiley.

[36] Tan, M., Yang, Y. & Yu, P. (2016) "The Influence of the Maker Movement on Engineering and Technology Education."

[37] Papanikolaou, V. (2015) "Surface-mount device prototyping in education - A feasible alternative to conventional through-hole practice."

[38] McRoberts M. (2013) "Beginning Arduino." 2nd edn. Apress, USA.

[39] Oellermann, M., Jolles, J.W., Ortiz, D., Seabra, R., Wenzel, T., Wilson, H. and Tanner, R.L. (2022) "Open Hardware in Science: The Benefits of Open Electronics." Integrative and Comparative Biology, 62(4), pp.1061-1075. https://doi.org/10.1093/icb/icac043. Published: 20 May 2022.

[40] D'Ausilio Alessandro (2012) "Arduino: A low-cost multipurpose lab equipment." Behav. Res. Methods, 44 (2012), pp. 305-313.

[41] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021) "A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations." Computer Science Review, 40.

[42] A. Hars (2001) "Working for free? Motivations of participating in open source projects." Proceedings of the 34th Hawaii International Conference on System Sciences, Maui, HI, USA, pp. 1–9.

[43] Banzi M., Shiloh M. (2014) "Make: Getting Started with Arduino the Open Source Electronics Prototyping Platform." 3rd edn. Maker Media, Inc, Sebastopol, CA, USA.

[44] D'Ausilio Alessandro (2012) "Arduino: A low-cost multipurpose lab equipment." Behav. Res. Methods, 44 (2012), pp. 305-313.

[45] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021) "A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations." Computer Science Review, 40.

[46] Blum, J. (2019) "Exploring Arduino." 2nd edn. Wiley.

[47] Wikipedia (2023) "Arduino Uno." Available at: https://en.wikipedia.org/wiki/Arduino_Uno.

[48] Bolanakis, D. (2021) "Microcontroller Prototypes with Arduino and a 3D Printer." 1st edn. Wiley.

[49] Kondaveeti, H.K. and Mathe, S.E. (2021) "A Systematic Literature Review on Prototyping with Arduino: Applications, Challenges, Advantages, and Limitations." Computer Science Review, 40(38).

[50] Marzoli, I., Rizza, N., Saltarelli, A., & Sampaolesi, E. (2021) "Arduino: From Physics to Robotics." In: D. Scaradozzi, L. Guasti, M. Di Stasio, B. Miotti, A. Monteriù, & P. Blikstein, eds. Makers at School, Educational Robotics and Innovative Learning Environments.

[51] Muller, P. O., Stich, C. and Zeidler, C. (2001) "Components @ work: component technology for embedded systems." Proceedings 27th EUROMICRO Conference. 2001: A Net Odyssey, Warsaw, Poland.

[52] Fowler, K. (2014) "Developing and Managing Embedded Systems and Products." Elsevier Science. Chapter 10 - Electronic Design.

[53] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021) "A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations." Computer Science Review, 40.

[54] Muller, P. O., Stich, C. and Zeidler, C. (2001) "Components @ work: component technology for embedded systems." Proceedings 27th EUROMICRO Conference. 2001: A Net Odyssey, Warsaw, Poland.

[55] Muller, P. O., Stich and C. Zeidler, C. (2001) "Components @ work: component technology for embedded systems." Proceedings 27th EUROMICRO Conference. 2001.

[56] Elverum, C.W., Welo, T. and Tronvoll, S. (2016) "Prototyping in New Product Development: Strategy Considerations." Procedia CIRP, 50, pp.117-122.

[57] Arduino (n.d.) "Shields & Carriers." Available at:
https://store.arduino.cc/collections/shields-carriers.

[58] Kondaveeti, H.K. and Mathe, S.E. (2021) "A Systematic Literature Review on Prototyping with Arduino: Applications, Challenges, Advantages, and Limitations." Computer Science Review, 40(38).

[59] Marzoli, I., Rizza, N., Saltarelli, A., & Sampaolesi, E. (2021) "Arduino: From Physics to Robotics." In: D. Scaradozzi, L. Guasti, M. Di Stasio, B. Miotti, A. Monteriù, & P. Blikstein, eds. Makers at School, Educational Robotics and Innovative Learning Environments.

[60] Muller, P. O., Stich, C. and Zeidler, C. (2001) "Components @ work: component technology for embedded systems." Proceedings 27th EUROMICRO Conference. 2001: A Net Odyssey, Warsaw, Poland.

[61] Fowler, K. (2014) "Developing and Managing Embedded Systems and Products." Elsevier Science. Chapter 10 - Electronic Design.

[62] Kondaveeti, H. K., Kumaravelu, N. K., Vanambathina, S. D., Mathe, S. E., & Vappangi, S. (2021) "A systematic literature review on prototyping with Arduino: Applications, challenges, advantages, and limitations." Computer Science Review, 40.

[63] Muller, P. O., Stich, C. and Zeidler, C. (2001) "Components @ work: component technology for embedded systems." Proceedings 27th EUROMICRO Conference. 2001: A Net Odyssey, Warsaw, Poland.

[64] Muller, P. O., Stich and C. Zeidler, C. (2001) "Components @ work: component technology for embedded systems." Proceedings 27th EUROMICRO Conference. 2001.

[65] Elverum, C.W., Welo, T. and Tronvoll, S. (2016) "Prototyping in New Product Development: Strategy Considerations." Procedia CIRP, 50, pp.117-122.

INDICE