Kingdom of Saudi Arabia
Ministry of Education
College of Computer
Computer Engineering Dept. (452)

المملكة العربية السعودية
وزارة التعليم
كلية الحاسب
قسم هندسة الحاسب (452)

==

Design and Implementation of Device Automation using Augmented Reality and Internet of Things

Student

Abdulmalek Aldossery	**ID: 411109345**
Khaled Alneef	**ID: 411109371**
Omar Alsaeed	**ID: 411109381**

Supervisor

Dr. Syed Sohail Ahmed

A project report submitted in partial fulfillment of the requirements for B.Sc. degree in Computer Engineering.

Qassim-Saudi Arabia

Second Semester 1445(2024)

P.O. Box 6688, Qassim 51452, Tel: +966-16-3801730, Fax: +966-16-3801731 ص.ب 6688 القصيم 51452 هاتف: +966163801730 فاكس:+96663801731
www.coc.qu.edu.sa coc@qu.edu.sa cocd@qu.edu.sa

Table of Contents

List of Figures

List of Tables

Certificate

I certify that the project report has been prepared and written under my direct supervision and guidance.

This report is approved for submission for its evaluation.

Dr. Syed Sohail Ahmed

Dedication

This report is dedicated to all the individuals who have contributed to the success of this project. Their unwavering commitment, hard work, and dedication have been instrumental in achieving the goals and objectives set forth.

Abdulmalek Aldossery
Khaled Alneef
Omar Alsaeed

Qassim University
College of Computer | كلية الحاسب

Acknowledgment

First and foremost, we express gratitude to Allah for bestowing us with a blessed mind. Next, we extend our thanks to our supervisor Dr. Syed Sohaill and families who have provided unwavering support, stood by us, encouraged us, and propelled us toward achieving excellence. Additionally, we acknowledge our friends for their support and encouragement throughout the completion of this project.

Abdulmalek Aldossery

Khaled Alneef

Omar Alsaeed

Abstract

These days, Device Automation through IoT has become an essential component, and its reach can be expanded to encompass Augmented Reality. Augmented Reality is employed to overlay interactive content onto real-world objects, facilitating the placement of 3D objects for the control of various IoT devices. This report investigates the fusion of Augmented Reality (AR) with IoT-based Device Automation. It focuses on using AR to overlay interactive elements onto real-world objects, enabling IoT device control through 3D objects. The motivation behind this project stems from the growing significance of both IoT and AR in modern technology. IoT has revolutionized device automation, offering convenience and efficiency, while AR has the potential to further enhance user interactions with IoT devices. We aim to bridge the gap between these two domains to create more intuitive and immersive smart device experiences. Our findings demonstrate that AR integration enhances user control and engagement. Users can interact with 3D device representations, simplifying smart device management. This integration offers potential benefits like improved user satisfaction and efficient device control, paving the way for a more accessible. In conclusion, this project explores the fusion of Augmented Reality and IoT in the context of device automation, driven by the desire to enhance user experiences. The findings underscore the potential benefits of this integration and its capacity to redefine how we interact with and manage IoT devices.

Chapter 1: Introduction

1.1 Introduction

The integration of Internet of Things (IoT) and Augmented Reality (AR) technologies represents a significant advancement in the development of our digital landscape in a world that is quickly developing and where connected smart devices are becoming more common. Our interactions with and control over the technologies that are all around us could be drastically changed by the convergence of these two transformative domains.

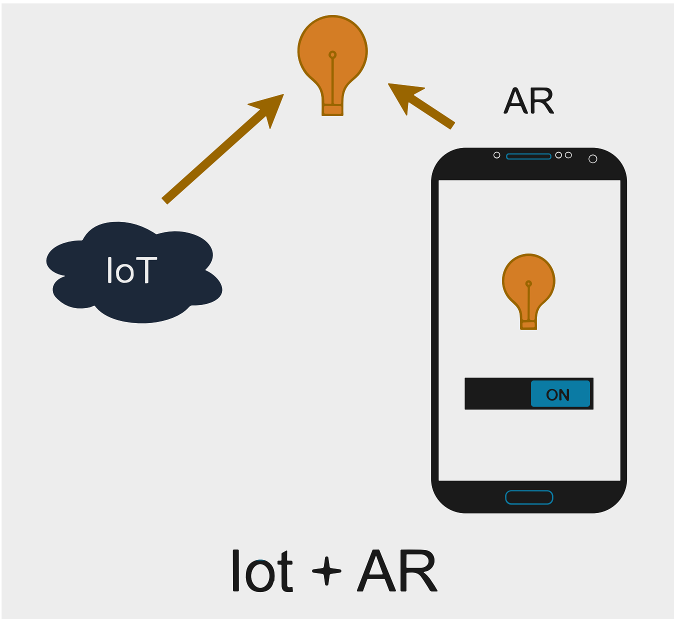


Figure 1: This figure shows that how we will use IoT devices with AR technology to control certain device.

The number of active IoT connections is expected to increase to 25.5 billion by the year 2026[1]. This rapid growth in IoT connectivity is paving the way for an even more immersive and intuitive interaction with everyday devices like lighting, appliances, and even industrial machinery. By incorporating intelligence into these devices, the Internet of Things has already had a significant impact on how we live our lives, go about our daily business, and interact with our surroundings. With their seamless communication, these networked devices increase convenience and effectiveness and provide data-driven insights into our surroundings.

Although human engagement with these devices is still restricted by things like voice commands, there is still an opportunity for a more intuitive and immersive approach.

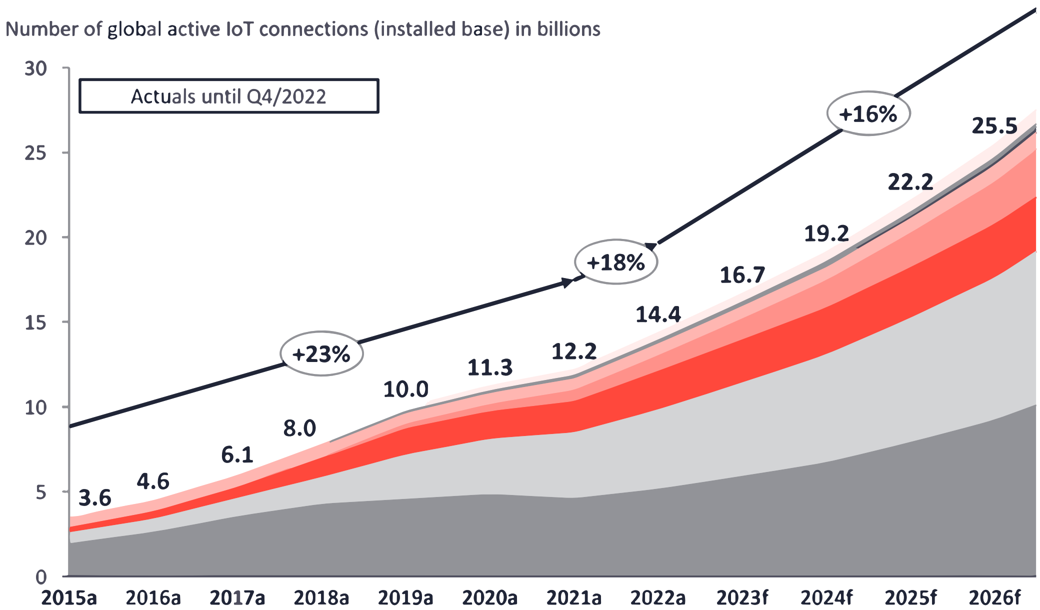


Table 1: ***Number of global active IoT connections (installed base) in billions [1].***

While AR, a technology that overlays digital data and virtual objects on our physical reality, has proven its potential in a variety of industries, including gaming, entertainment, healthcare, and education. With its seamless merging of the actual and digital worlds, augmented reality (AR) offers a fresh way for us to interact with our surroundings and the technology found there. IoT is currently viewed primarily as an infrastructure, and how people interact with it is still a difficult problem to solve [2]. The methods by which users connect to and interact with the IoT are anticipated to have a significant impact on the delivery of valuable services via it. AR stands out as an alluring solution in this regard [3].In this context, AR and IoT appear to complement each other. Specifically, IoT serves as a foundational framework that facilitates the scalability and dependability of AR, while AR, in turn, functions as an efficient interface to IoT [4]. Our project explores the integration of IoT and AR, imagining a time where operating smart devices will be as natural as physically reaching out and interacting with them. Our goal is to incorporate device controls and IoT data streams into an AR interface, making it easier and more immersive to interact with the growing network of interconnected devices. This combination has the potential to improve our daily lives by letting us easily manage our smart homes, expedite complex jobs, and optimize industrial processes with just one finger or hand movement.

1.2 Problem Statement and Summary of Functional Requirements

1.2.1 Problem Statement:

The traditional interfaces for interacting with IoT devices may not be user-friendly or intuitive. How can we visualize real-time data from IoT devices in a contextual manner?

The fusion of Augmented Reality (AR) and Internet of Things (IoT) is crucial to bridge the gap between physical and digital worlds for efficient device management and interaction. So that, the integration of Augmented Reality (AR) and Internet of Things (IoT) for device control addresses the need for a seamless and intuitive interface to interact with a growing number of IoT devices. By using AR technology we can visualize data through it for example we can see the devices progress and control it.

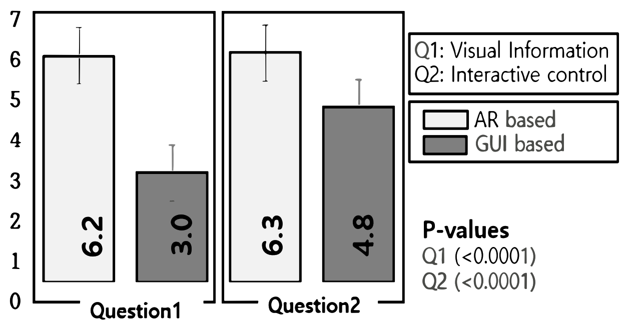


Figure 2: ***This figure shows user responses were collected to assess the effectiveness of two different interfaces, one based on Augmented Reality (AR) represented by the light bar on the left, and the other using a conventional Graphical User Interface (GUI) represented by the dark bar on the right [4].***

1.2.2 Summary of Functional Requirements:

1- *The system must provide real-time data updates and alerts, enhancing user awareness and responsiveness to changing device conditions.*

2- *Seamless compatibility with a variety of IoT devices, coupled with an intuitive user interface, will empower users to harness the full potential of AR and IoT integration.*

3- *This project aims to develop a system that allows users to control IoT devices using AR interfaces, reducing the complexity of device management.*

4- *This project must be compatible with smartphones, and tablets.*

1.3 Goals & Objectives

Since we came up with the idea of our project, we made clear goals and objectives that will help us to construct and finalize our project in an easy manner.

1.3.1 Goals:

The goals of our project are:

1- *The primary goal is to provide a seamless and intuitive experience for controlling devices. By combining AR and IoT.*
2- *Aim to create an interface that feels natural and immersive, enabling us to interact with our devices effortlessly.*
3- *Aim to remote control and monitor devices using AR and IoT.*
4- *The goal is to embed AR and IoT technologies to enable automation and intelligent interactions between devices.*

1.3.2 Objectives:

The objectives for our project are:

1- *Simplify the process of managing diverse devices, reducing the need for individual control interfaces and streamlining the user's control experience.*

2- *An objective is to enable remote accessibility and control of devices. By using AR interfaces integrated with IoT, we can remotely access and control our devices.*

3- *One of the objectives is to centralize device management, AR interfaces integrated with IoT enable us to control and monitor multiple devices from a single interface.*

1.3.3 Closure:

In conclusion, the goals and objectives of using AR and IoT to control house devices revolve around enhancing user convenience, streamlining device management and providing real-time feedback. These goals aim to create a seamless and efficient control experience.

1.4 Motivation

The convergence of Augmented Reality (AR) and Internet of Things (IoT) technologies has opened up exciting possibilities in various domains, revolutionizing the way we interact with our environment. This project aims to harness the power of AR and IoT to create innovative solutions that enhance user experiences, improve efficiency, and enable new opportunities across industries [5]. One of the main motivations in this project is the exciting opportunities of combining AR and IoT arise from their ability to transform how we interact with our physical surroundings. AR uses digital information onto the real world, providing us with a rich and immersive experience. By integrating IoT devices into this augmented environment, we can go to new levels of interactivity and personalized experiences [6]. AR enhances the control experience by providing contextual information and feedback about the devices. When interacting with a device through AR, we can instantly access relevant information for this device, such as status updates. This real-time feedback helps us to control our device usage for comfort.

1.5 Overview of the Report

This report includes five chapters, all of which are intricately linked to the Design and Implementation of Device Automation using Augmented Reality and Internet of Things. Chapter 1 sets the stage by delineating the project's prerequisites, objectives, and scope, anchoring it firmly in the context of AR and IoT. Chapter 2 delves into theoretical underpinnings and reviews the relevant literature, emphasizing the crucial role of hardware components in the project while highlighting the advantages and disadvantages of a Device Automation using Augmented Reality and Internet of Things. Moving forward to Chapter 3, *the focus shifts to the hardware design and the various device automation technologies employed in this IoT-enabled setup. This chapter also underscores the prerequisites for successful system implementation. Chapter 4 navigates into the software aspects of the project, detailing its implementation and the testing processes. Lastly, Chapter 5 presents the outcomes and conclusions derived from the project, offering insights into potential future developments within the realm of AR, IoT, and device automation.*

1.6 Primary Constraints

1 ***Latency and Network Connectivity:*** *AR applications require real-time responsiveness in other words less than one second delay to provide users with a seamless experience. IoT devices rely on network connectivity to communicate and exchange data. Latency in network communication can result in delays and disrupt the user experience.*

- **1.1** ***Latency in AR Applications:*** *Latency in AR applications refers to the delay between user actions and the corresponding response from the AR system. In AR, real-time responsiveness is critical for providing users with a seamless and immersive experience. High latency can lead to several issues:*
  - **1.1.1** ***Object Placement Accuracy:*** *Latency can result in inaccuracies in placing virtual objects in the real-world environment. Users may perceive objects as not properly anchored or "floating" in space, disrupting the illusion of reality [8].*
  - **1.1.2** ***Interaction Delays:*** *AR applications often involve real-time interaction, such as touching virtual objects or responding to some events. Latency can introduce delays in these interactions, affecting user experience and overall satisfaction.*
- **1.2** ***Network Connectivity in IoT Devices:*** *IoT devices rely on network connectivity, typically the internet, to communicate with each other, cloud servers, and end-users. Network-related issues in IoT can have significant consequences:*
  - **1.2.1** ***Data Loss:*** *Latency can result in inaccuracies in placing virtual objects in the real-world environment. Users may perceive objects as not*

properly anchored or "floating" in space, disrupting the illusion of reality [9].

1.2.2 ***Data Delays:*** *AR applications often involve real-time interaction, such as touching virtual objects or responding to some events. Latency can introduce delays in these interactions, affecting user experience and overall satisfaction [10].*

2 ***Data Security and Privacy:*** *IoT devices collect and transmit sensitive data. Integrating AR can introduce additional security concerns, such as the potential for unauthorized access to AR interfaces or IoT devices. Ensuring the security and privacy of both AR and IoT components is crucial.*

3 ***Compatibility and Standardization:*** *The IoT ecosystem consists of a wide range of devices from different manufacturers, each using different communication protocols and standards. Integrating AR with IoT may require significant effort to ensure compatibility and standardization. Standardization is necessary because, without established regulations, precise guidelines, and worldwide standards, the industry will eventually face serious incompatibilities from unregulated IoT expansion which are more difficult to track and examine their impacts to different sectors [11]. The standardization of IoT protocol stacks is critically important for the successful and widespread adoption of Internet of Things (IoT) technologies. A standardized IoT protocol stack helps ensure interoperability, security, and reliability within the IoT ecosystem [12].*

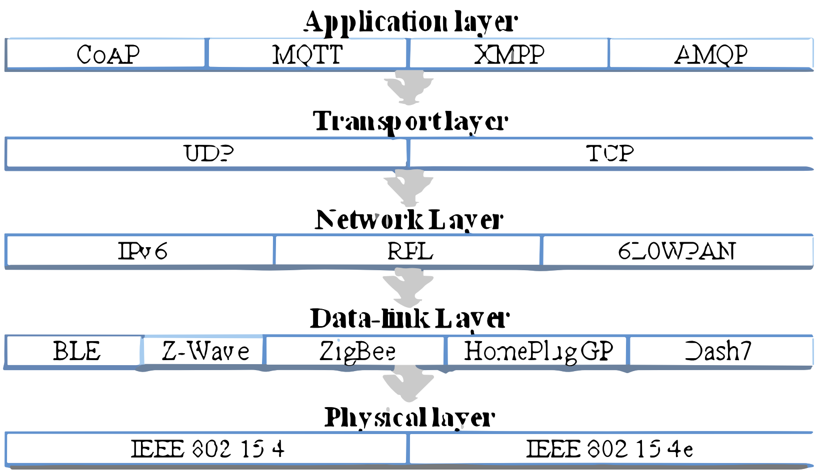


Figure 3: ***IoT Protocol Stack standard [12].***

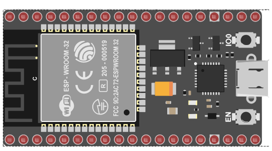
4 ***Scalability:*** *As the number of IoT devices increases, managing and controlling them through AR interfaces can become complex. Scalability issues need to be addressed to handle a growing number of devices efficiently.*

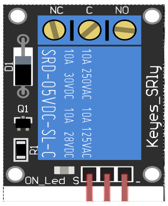
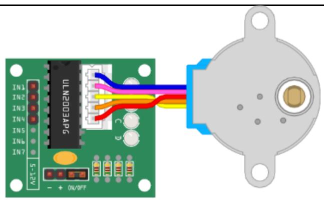
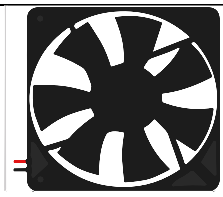
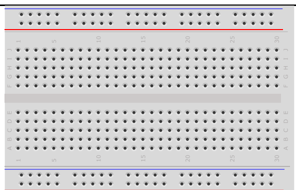
5 ***Power Consumption**: Many IoT devices operate on battery power. Integrating AR applications can increase the power consumption of devices, which may result in shorter battery life. Optimizing power usage is essential for IoT devices.*

6 ***Environmental Factors:*** *Both AR and IoT devices may need to operate in various environmental conditions. Ensuring robustness and reliability under different circumstances is crucial.*

7 ***Cost:*** *Integrating AR and IoT technologies can be costly, both in terms of hardware and software development. Organizations need to assess the cost-benefit ratio and consider whether the investment is justified.*

Here table shows the approximate cost for hardware components:

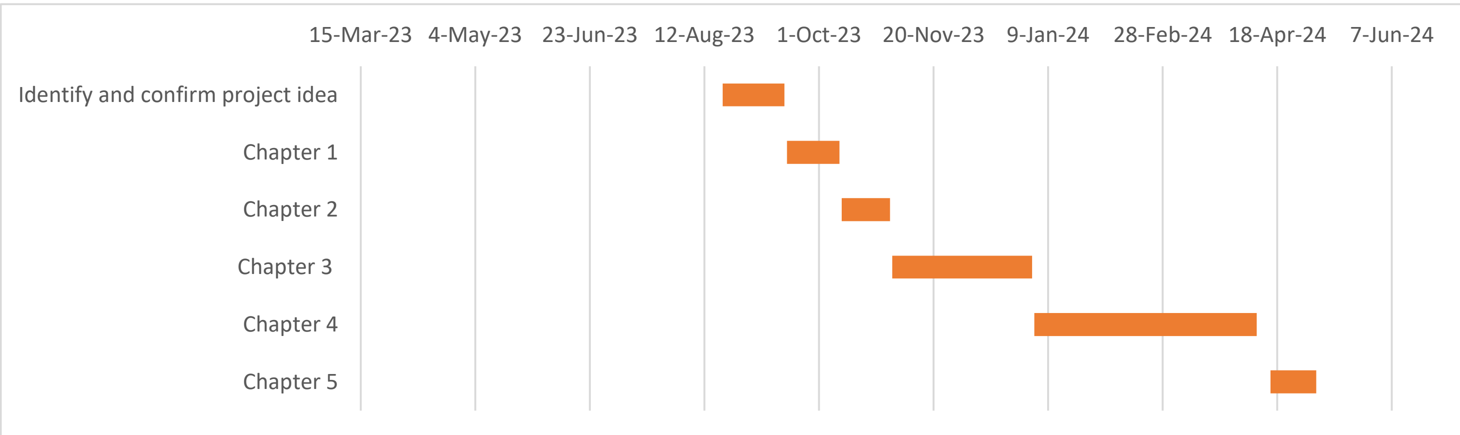
Hardware component	***Cost in dollar***	***Cost in riyals***
 *Lamp socket*	*$1.33*	*5 SAR*
 *ESP32*	*$13.33*	*49.99 SAR*

 *Four channels relay*	*$14.93*	*56 SAR*
 *Jumper wires*	*$4*	*15 SAR*
 *LED lamp*	*$1.78*	*7 SAR*
 *Stepper Motor*	*$17.33*	*65 SAR*
 *Fan*	*$14.66*	*54.98 SAR*
 *Breadboard*	*$3.17*	*11.89 SAR*

 Battery	$9.86	37 SAR
 Batteries cell	$15.28	57.29 SAR
	Total cost is $120.44	Total cost is 451.71 SAR

Figure 4: ***Hardware cost in dollars and Saudi Riyals.***

1.7 Timeline



Chapter 2: System Overview

2.1 Literature Review

For AR development, it appears that researchers overwhelmingly prefer Unity - Unity3D as their primary tool for developing AR applications, with Vuforia being the second most commonly used option. Unity3D's wide appeal can be linked to its adaptability and strong support for both 2D and 3D development, whereas Vuforia is favored for its specialized AR features. And for hardware, the main focus was on creating augmented reality (AR) applications for mobile devices, primarily smartphones and tablets, as these were the most widely used devices. A few scientific papers selected AR glasses as their primary hardware target, followed by a small number that chose head-mounted displays (HMDs)[15]. Combining Augmented Reality with the Internet of Things offers a dynamic fusion of 3D object manipulation to enable smart device automation. This is facilitated through a user-friendly web server built to control appliances, leveraging Message Queuing Telemetry Transport protocol for seamless communication, it is an application-layer protocol. It is an extremely simple and lightweight messaging protocol that not only displays sensor data but also manages switch controls, illuminating the concept of IoT-driven device automation [13]. Users can easily establish connections based on predefined relationship mappings utilizing only an augmented reality browser program, devoid of tracking information and augmented reality sensors. To access objects in the environment, a user can access the collected augmented reality attributes and tracking information shared through the server. These studies are primarily concerned with how to effectively integrate tracking information and augmented reality presentation datasets from a remote server. Recent research has placed emphasis on the process of registering and managing augmented reality datasets using cloud computing infrastructure. These investigations have highlighted the advantages of leveraging computational power to alleviate the demanding task of matching a substantial quantity of features, especially in light of the limited computing capabilities of mobile devices. Numerous studies have explored the integration of augmented reality with cloud-based computing resources [20]. Sung Lae Kim introduced a concept on integrating Unity3D objects into the actual physical environment [14]. Introducing virtual buttons for appliance control via the Blynk server enhances communication and extends this capability to multiple targets for real-time monitoring as well [7]. Blynk stands as a dynamic IoT platform tailored for iOS and

Android smartphones. Its core purpose lies in facilitating the remote management of devices like Arduino, Raspberry Pi, and NodeMCU through Internet connectivity. This versatile application empowers users to design graphical interfaces or Human Machine Interfaces (HMI) by configuring and linking various widgets with appropriate addresses. Built with a specific focus on the Internet of Things, Blynk excels in numerous functions. It not only enables remote hardware control but also offers the capability to display sensor data, store and visualize data, and perform a range of other impressive tasks. Within the Blynk platform, three primary components play crucial roles: Blynk App: This user-friendly mobile application provides a rich assortment of widgets, empowering users to craft captivating interfaces for their projects. Blynk Server: Serving as the pivotal communication nexus, the Blynk Server manages all interactions between smartphones and the connected hardware. Users have the flexibility to opt for the Blynk Cloud or establish a localized, private Blynk server. Notably, it's an open-source solution, capable of seamlessly handling a large number of devices, and can even be deployed on hardware like the ESP32 and Arduino. [16].

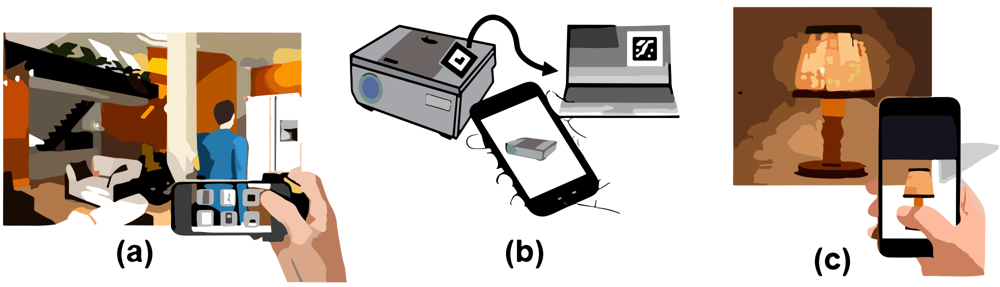


Figure 5: ***Different augmented reality (AR) interactions with Internet of Things (IoT) devices: (a) controlling devices in their physical location or remotely using a traditional graphical user interface (GUI) button, (b) employing metaphorical and intuitive interactions like virtual dragging to activate device functions [18], (c) engaging with a virtual or augmented environment to influence real-world actions [19].***

Furthermore, it's crucial to take into account the distinct attributes of augmented reality (AR) content in relation to various Internet of Things (IoT) devices, given the extensive diversity within the IoT device landscape. It's noteworthy that this concept closely parallels the way in which website components require different configurations when viewed on mobile and desktop computing devices. Therefore, to ensure the seamless and universal deployment of an IoT-enabled AR platform, it is essential for the platform to intelligently modulate the extent of AR *content representation in accordance with the unique characteristics of each IoT device, thereby tailoring the AR experience to its specific context and capabilities.*

2.2 System Analysis and Specification

Our project's main function is to allow us to control devices using IoT and AR at the same time, but there are several requirements, operational parameters, limitations and restrictions.

2.2.1 Hardware requirements:

1- ESP32: a chip used for WIFI connection, so that we can control the target hardware. ESP32 provides – 1 core at 240 MHz speed of processor, with 520 KB SRAM, and it can receive and send 1.8 V-5.5 V [51]

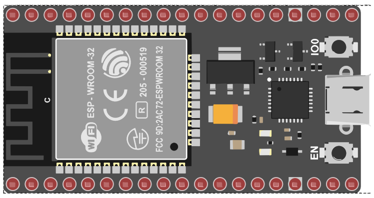


Figure 6: ESP32 chip [24].

2- Relay: used to amplify the voltage that comes from the voltage source; because some hardware components need more voltage. The maximum switching voltage of the Pcchips brand is 5V.

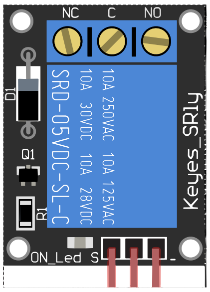


Figure 7: Relay [25].

3- Jumper wires: used to connect all other hardware components.

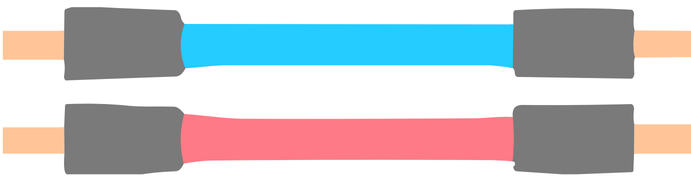


Figure 8: Jumper wires [26].

4- LED lamp: one of the targeted hardwares, so that we control it by using AR and IoT. It works with 12V.

Figure 9. LED lamp [26].

5- Lamp socket: to connect the LED lamp to the circuit [29].

Figure 10. Lamp socket [26].

6- Motor: it is also one of the targeted hardwares, and we can control it. Rated voltages for stepper motors are 5V DC [29].

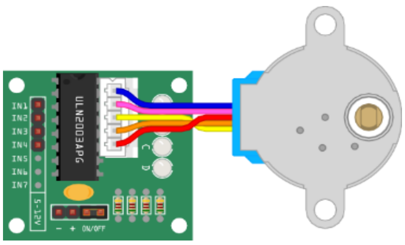


Figure 11: Stepper Motor [26].

7- Fan: is the third and last targeted hardware to control. This fan needs 12V DC to operate.

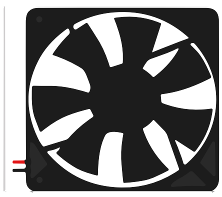


Figure 12: Fan [26].

8- Breadboard: the base of the whole project, used with wires to connect hardware components.

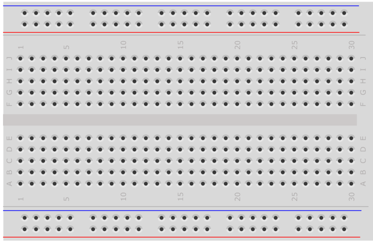


Figure 13: Breadboard [26].

9- Power source (Battery): to provide the essential power to run the whole project. 12V battery so that we can control the fan and the LED lamp.

Figure 14. Power source (Battery) [26].

10- Batteries cell: to connect the batteries to the circuit.

Figure 15. Batteries cell [26].

11- A target to trigger the AR gestures: it can be image, text or any recognizable thing.

2.2.2 Software requirements:

1- Arduino IDE: The Arduino IDE, or Integrated Development Environment, is an open-source platform central to the Arduino. It simplifies coding, compiling, and uploading for Arduino microcontrollers and other boards. With a user-friendly code editor, libraries. It is used.

2- Blynk application: Blynk is an IoT platform designed for IOS and Android smartphones. Used to establish communication between the devices like Arduino, Raspberry Pi, and ESP32 over the Internet and other hardware components [16].

3- UNITY: Unity is a popular cross-platform game engine and development tool used for creating interactive 2D and 3D content, including video games, simulations, and other real-time experiences. It provides a comprehensive set of tools, a visual editor, and a powerful scripting API that allows developers to design and build games for various platforms such as desktop, mobile, consoles, and virtual reality (VR) devices[21].

4- Vuforia Engine:

Enables developers to build AR experiences by recognizing and tracking images, objects, and environments in the real world and overlaying digital content on top of them, it also utilizes computer vision technology to detect and track visual targets, enabling virtual content to be seamlessly integrated with the physical world [17].

2.2.3 Operational parameters:

1- Control commands: are a critical aspect of device management, enabling us to interact with and manage various devices efficiently. These commands serve as instructions that allow us to control and manipulate the behavior, settings, and functionality of devices to meet specific requirements [22].

2- Data feedback: plays a main role in the domain of device control, serving as a vital mechanism for users to gain awareness of the current state and conditions of the devices they are controlling. It involves the transmission and presentation of relevant information, allowing users to monitor, analyze, and make informed decisions based on real-time data.

3- Communication protocols: communication protocols are fundamental things if it comes to connecting two different technologies, and to establish communication between the AR interface, IoT platform, and device controller, we are using WIFI by ESP32 hardware component.

2.2.4 Limitations and restrictions:

1- Complexity and integration challenges: it is not easy to integrate 2 of new technologies around the world so we need to be accurate in choosing the hardware and software we need to implement.

2- Network connectivity: our system relies on a stable and reliable network connection for communication between the AR interface, IoT platform, and device controller.

3- Cost: implementing AR and IoT technologies for device control may cost more, like: hardware and software requirements.

4- Trigger target: the target we use to trigger the AR gestures should be clear and recognizable.

2.3 Alternate Designs

The ESP32 is linked to the relay input through one of its GPIO pins. This relay module functions as an electronic switch to control circuit activation and deactivation. Its primary purpose is to manage voltage and current levels beyond the capabilities of the microcontroller. Specifically, a 5V relay module is employed, featuring three connection terminals known as NC (Normally Open), NO (Normally Closed), and COM ports. On the other side, there are VCC, GND, and IN pins that must be connected to the board. The relay module utilizes an electric current to manipulate the switches, thanks to a coil that attracts and releases the switch contacts, assisted by a spring that separates them when the coil is not powered. To control a device, the appliance wire is split into two parts, and these are linked to the COM and NC ports, while the VCC pin is connected to the ESP32's Vin pin.

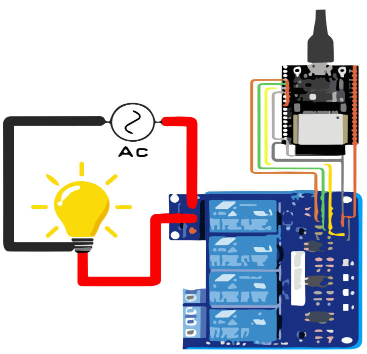


Figure 16: Implementation of the ESP32 and Relay for Home Appliance Connectivity

2.4 Proposed Design

2.4.1 Flowchart:

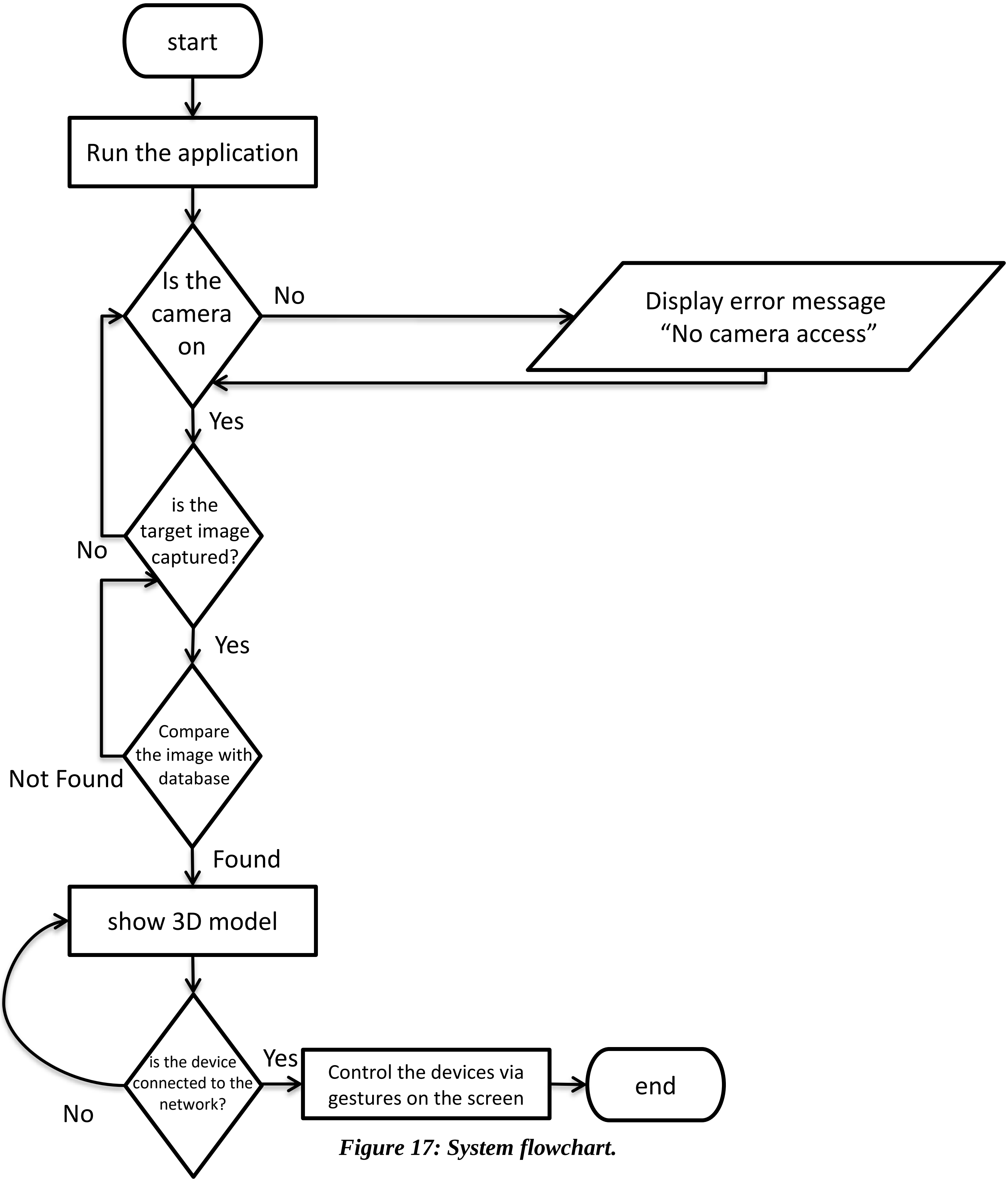


Figure 17: System flowchart.

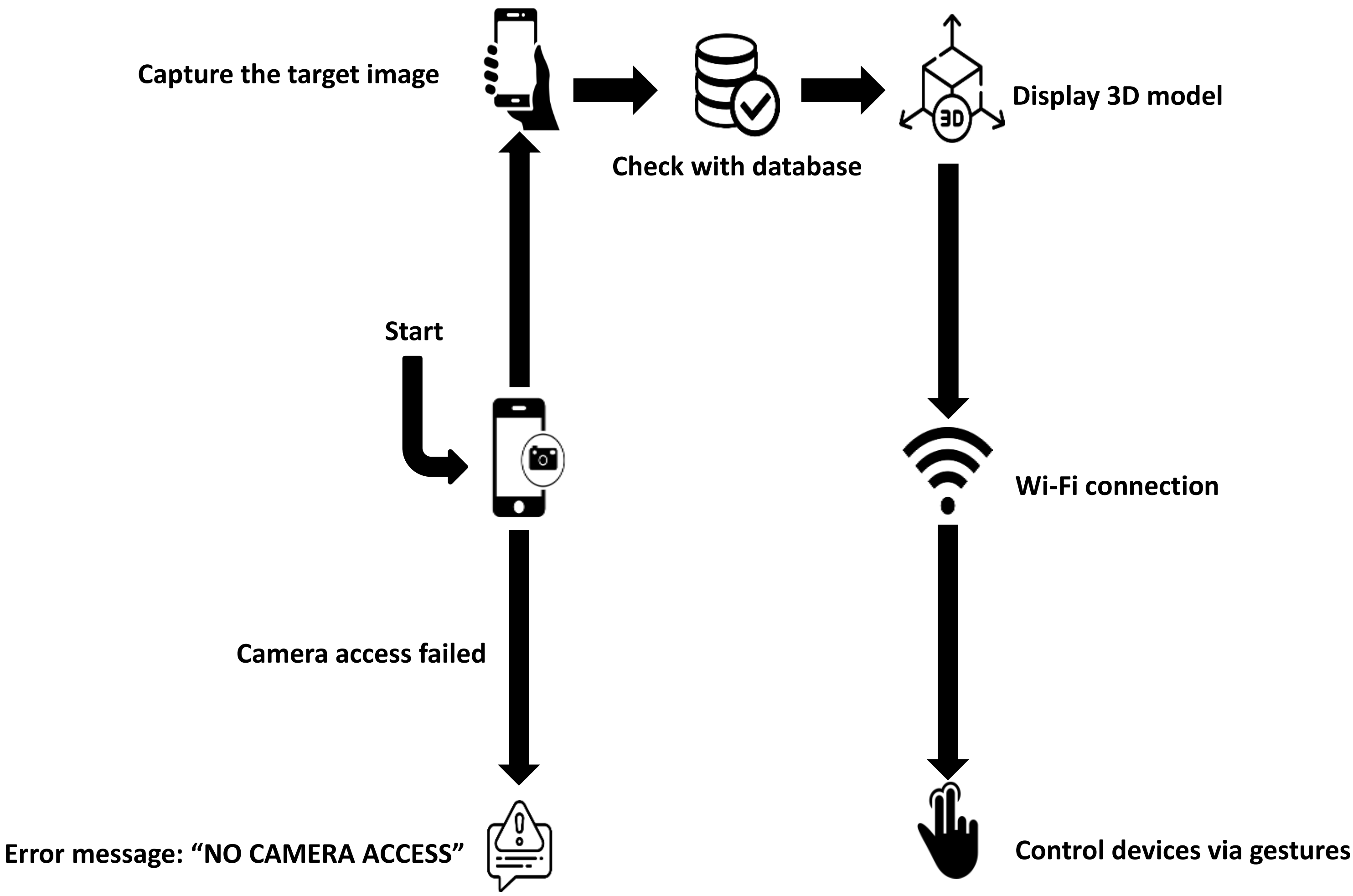


Figure 18: System physical diagram.

2.4.2 Algorithm:

```
1. Start
2. Run application
3. If camera access is not allowed:
4.     Display error message "No camera access"
5. Else:
6.     While camera is on:
7.         If camera is off:
8.             End
9.         Else:
10.            Capture target image
11.            If target image is not captured:
12.                Continue capturing images (go to step 10)
13.            Else:
14.                Compare captured image with database
```

```
15.           If image not found in database:
16.               Continue capturing images (go to step 10)
17.           Else:
18.               Display 3D model
19.               If device is not connected to the network:
20.                   Continue capturing images (go to step 18)
21.               Else:
22.                   Control devices via gestures
23. End
```

2.4.3 Circuit diagram:

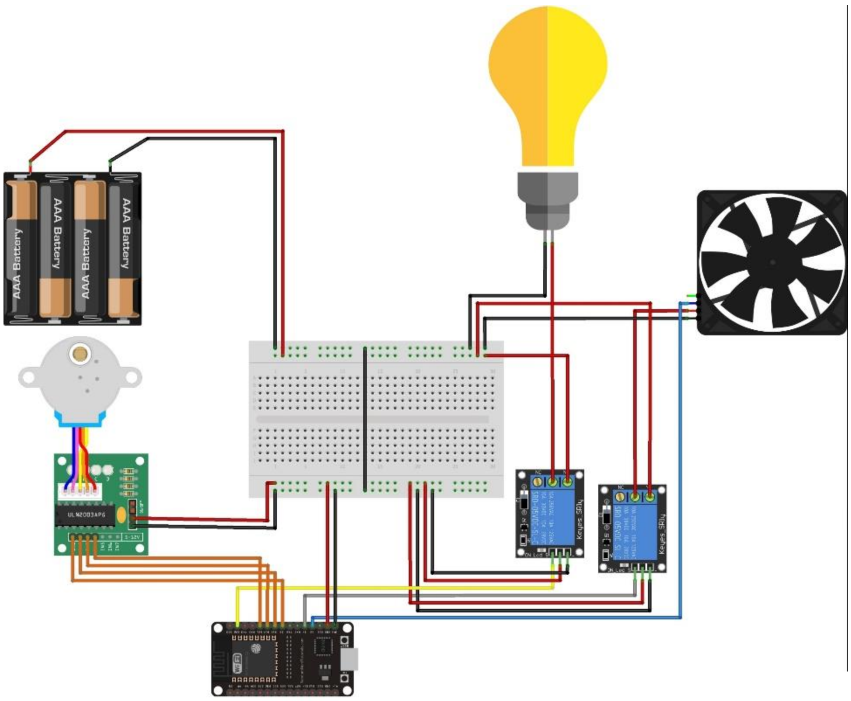


Figure 19: Circuit diagram.

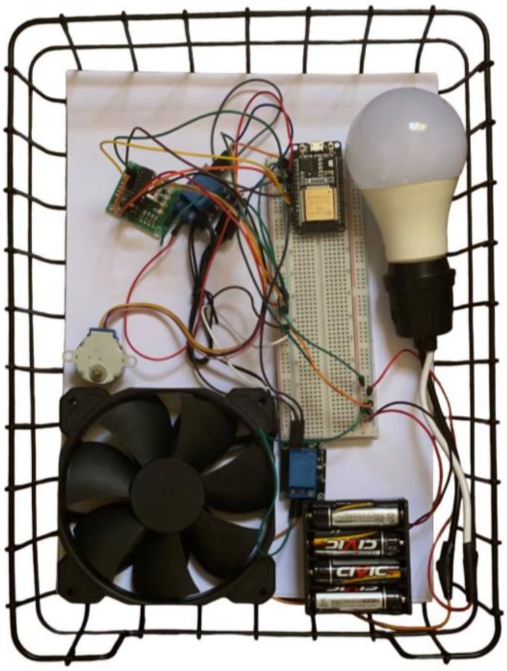


Figure 37: Prototype circuit.

Figure 38: Final hardware.

Chapter 3: Modeling and System Design

3.1 Introduction

Modeling and simulation enable designers to test whether design specifications are met by using virtual rather than physical experiments. The use of virtual prototypes significantly shortens the design cycle and reduces the cost of design. In this chapter, we have discussed the physical components of our system and the assessment and testing methods to verify our design. We have also discussed our system's social, ethical, health, and political impact.

3.2 Details of Theoretical Model

A mathematical idea known as homography transformation is applied in computer vision and image processing, especially in relation to geometric transformations and image registration. If two separate images of the same scene are connected by a projective transformation, it describes the relationship between the points in the two images. Homography is frequently used in the context of AR to align virtual objects with the scene as seen through a camera. The coordinates of points in one image (the source image) are related to the coordinates of corresponding points in another image (the destination image) by the homography transformation matrix H, a 3x3 matrix.

The formula for a homography transformation is given by:

$$s\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{H}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The homography matrix's parameters are represented by the elements of h. A collection of corresponding points from the source and destination images can be used to estimate these parameters. Any point from the source image to the destination image can be transformed using the homography matrix once it has been found. When it comes to augmented reality, *homography is frequently utilized for tasks like image rectification, where the objective is to eliminate camera distortion and align the image with a desired reference frame. It can also be used to position virtual objects in the physical world by changing their coordinates to match the viewpoint of the image captured by the camera [30].*

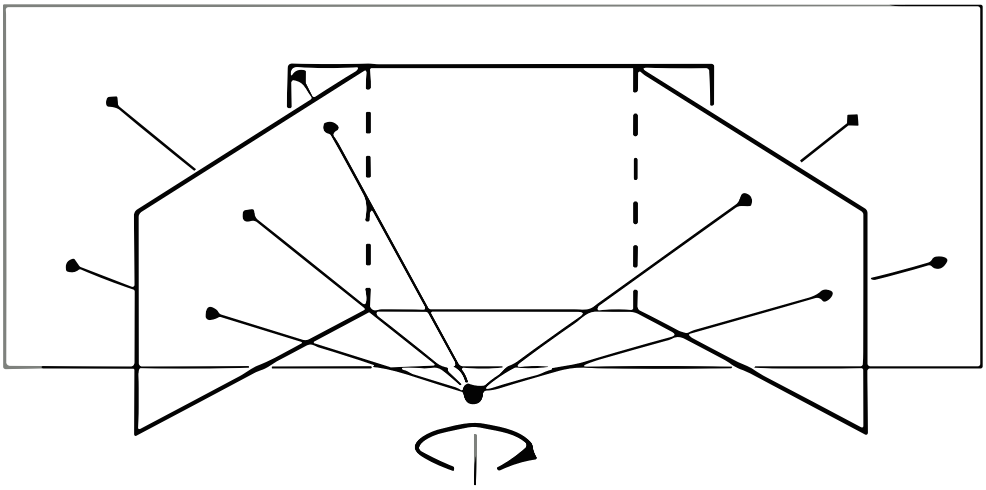


Figure 20: Rotating/translating camera [31].

In the domain of augmented reality (AR), the field of view (FoV), which defines the angular extent of the observable environment recorded by a camera system, is a fundamental concept. In the field of AR development, FoV plays a crucial role as a guiding principle with a variety of implications. It is important because it guarantees the authenticity of virtual object placement and content dimension optimization, which makes it easier for virtual elements to blend in with the user's real-world spatial context. FoV data is essential for spatial anchoring because it gives developers the necessary accuracy to place virtual objects precisely within the camera's field of view, improving the AR experience's overall stability and consistency. FoV also has an impact on the strategic design of user interfaces, helping to place interactive elements wisely and optimizing the use of computational resources by focusing attention on areas that are clearly visible. To summarize, a deep understanding of field of view is a fundamental principle in the creation of immersive augmented reality (AR) experiences. It creates settings that blend in with the real world and produces experiences that are intuitive to interact with and visually coherent [32].

$$HFOV = 2 \cdot \arctan\left(\frac{\text{Sensor Width}}{2 \cdot \text{Focal Length}}\right)$$

$$VFOV = 2 \cdot \arctan\left(\frac{\text{Sensor Height}}{2 \cdot \text{Focal Length}}\right)$$

$$DFOV = 2 \cdot \arctan\left(\frac{\sqrt{\text{Sensor Width}^2 + \text{Sensor Height}^2}}{2 \cdot \text{Focal Length}}\right)$$

These formulas are used to calculate the horizontal, vertical, and diagonal fields of view for a camera based on its sensor dimensions and focal length.

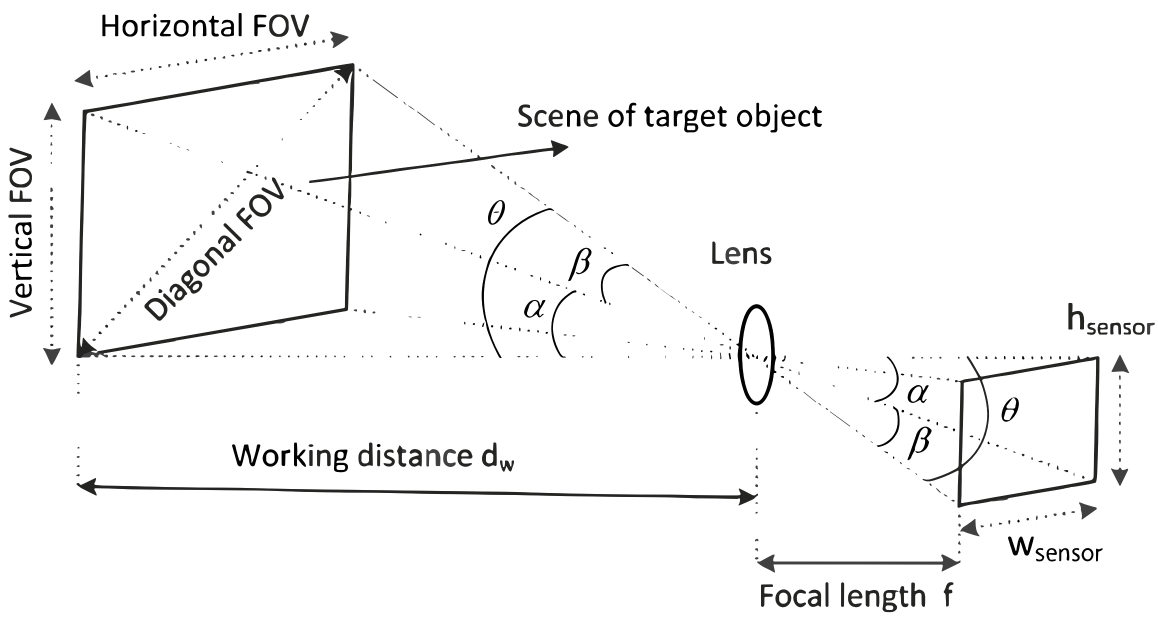


Figure 21: Illustration of camera lens's field of view (FOV) [33].

In applications of augmented reality (AR), distance is an important consideration. Recognizing and effectively applying distance data is crucial for different aspects of AR development. The distance formula in a three-dimensional space between two points is given by the Euclidean distance formula:

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

This formula calculates the straight-line distance between two points in three-dimensional space.With the aid of this formula, developers can place virtual objects precisely in the real world, allowing for realistic interactions and a smooth integration with the user's physical surroundings. Calculating distances makes virtual content appear at the right distances for *maximum realism, which greatly enhances depth perception. This formula helps with spatial comprehension, which is important for collision detection, directing interactive features, and dynamically varying the rendering of virtual content according to the user's distance. Furthermore, by enabling developers to prioritize rendering and interactions according to the user's proximity to virtual objects, it functions as a crucial tool for optimizing AR performance[35]. To sum up, the distance formula plays a crucial role in improving augmented reality applications by supporting spatial accuracy, immersive interactions, and the overall coherence of the experiences. In general, AR apps function best when the camera is placed between 50 and 80 cm away from the object. The surroundings, particularly lighting and reflective surfaces, has a significant impact on how well augmented reality (AR) experiences work. Variations in lighting can have a major effect on the accuracy of tracking and visual recognition in augmented reality applications. The system's ability to precisely overlay virtual content and detect objects in the real world may be compromised in low-light conditions. On the other hand, extreme variations in lighting can make it difficult to maintain a consistent and accurate augmented reality experience. Reflective surfaces—like mirrors or glass—present additional challenges because they introduce reflections that can trick augmented reality (AR) systems, leading to inaccurate object placement and recognition. Improving the dependability and smoothness of AR interactions in a variety of contexts and situations requires addressing these environmental factors. Our project, which integrates AR with IoT, is contingent on a Wi-Fi connection for its core functionality. The synergy between AR and IoT relies on seamless communication and data exchange between devices, and a reliable Wi-Fi connection serves as the linchpin for this integration. Therefore, ensuring a consistent and robust Wi-Fi connection is imperative for the successful operation and synergy of our AR and IoT integrated system. Advancements in camera and sensor technology have significantly mitigated the limitations of AR. Depth sensors and LiDAR scanners enhance object recognition, spatial understanding, and occlusion with real-world objects[37]. Time-of-flight cameras improve object tracking and gesture recognition, providing a natural AR experience. High-resolution cameras capture detailed visual information for realistic virtual overlays. High frame rate cameras ensure smoother motion and reduce latency, aiding in minimizing motion sickness. Some AR devices use multiple cameras for enhanced tracking and 3D reconstruction. Integrated eye-tracking sensors enable intuitive interactions by detecting the user's gaze direction. Image stabilization reduces motion blur, crucial for fast movements or handheld devices. Infrared cameras in AR devices can detect heat signatures, providing additional environmental information while* *driving. AI-powered camera features improve image processing, noise reduction, and low-light performance for a better visual experience in challenging conditions.*

The maximum static friction is:

$$F_{static} = \mu_{static} \times m_{curtain} \times g$$

To ensure the curtain achieves full unfolding within a 10-second timeframe, the motor's required maximum power can be determined as:

$$P_{max} = F_{static} \times distance/time$$

The stepper motor accepts a power input ranging from 5V DC and generates a discrete output current within the range of 0.1 to 0.3 amps per phase[29]. As specific details regarding the exact power output are unavailable, we will compute the motor's power output using basic formula:

$$P = V \times I$$

The lower limit of theoretical power output from the motor will be:

$$P_{min} = V_{min} \times I_{min}$$

The upper limit of theoretical power output from the motor will be:

$$P_{max} = V_{max} \times I_{max}$$

3.3 Physical Realization

3.3.1 Circuit Design:

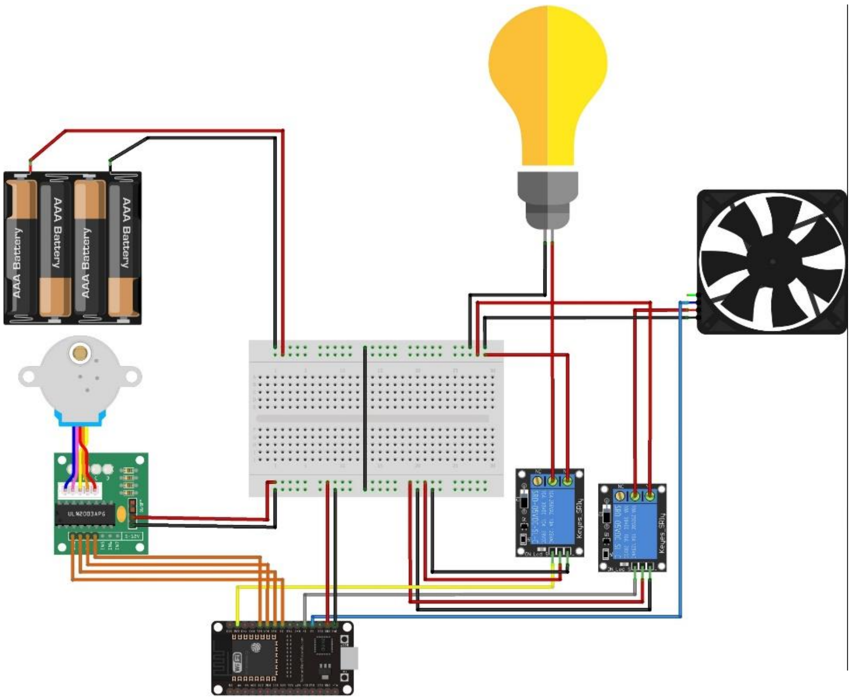


Figure 19: Circuit diagram.

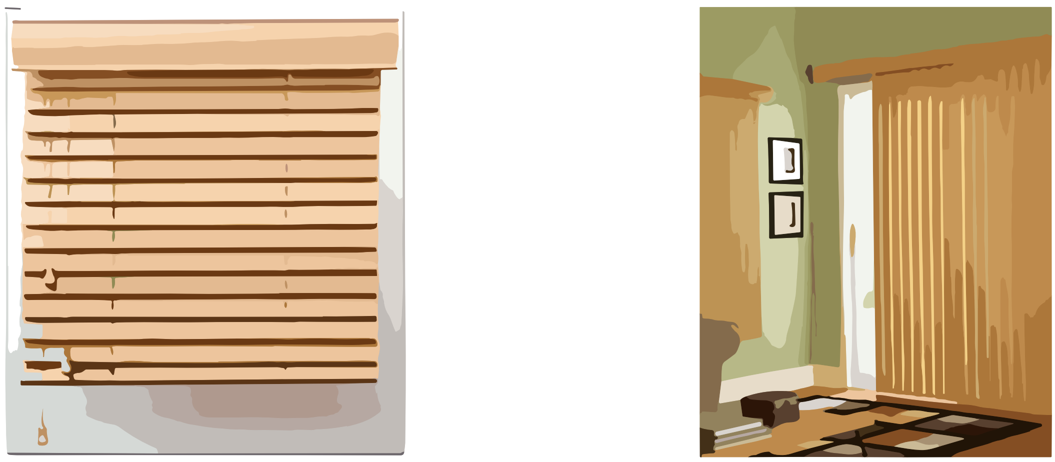
3.3.2 Algorithm:

```
1. Start
2. Run application
3. If camera access is not allowed:
4.      Display error message "No camera access"
5. Else:
6.      While camera is on:
7.          If camera is off:
8.              End
9.          Else:
10.             Capture target image
11.             If target image is not captured:
12.                  Continue capturing images (go to step
10)
13.             Else:
14.                 Compare captured image with database
15.                 If image not found in database:
16.                       Continue capturing images (go to
step 18)

17.                 Else:
18.                     Display 3D model
19.                       If device is not connected to the
network:
```

```
20.                         Continue capturing images (go
to step 10)
21.                   Else:
22.                        Control devices via gestures
23. End
```

3.3.3 Mechanical Superstructure:

Energy management, particularly in terms of power monitoring and energy conservation, represents a prevalent application of device automation systems. This involves employing devices like smart plugs, and monitoring systems for direct control and monitoring, or indirect control through environmental manipulation. An example of indirect control is the adjustment of curtains or blinds, which plays a role in regulating lighting and heat within a household. Curtains serve various purposes, including acting as a shield against sunlight, facilitating air circulation, and contributing to the overall decor. Curtains can be broadly categorized as decorative elements or fully-operational items designed to manage the room, with variations such as roller blinds, Roman blinds, horizontal blinds, vertical blinds, and wooden blinds falling under the fully-operable category [28]. In our project, we are going to use horizontal blinds.



(a) *(b)*

Figure 22: Types of curtains (a) is horizontal blind, and (b) is vertical blind [28].

The stepper motor is a mechanical actuator employed in the controller is a 4-wire unipolar stepper motor, specifically the 28BYJ-48 model, while the driver utilizes the ULN2003A integrated circuit. The intrinsic circuit of the ULN2003A IC incorporates the Darlington circuit *for current amplification. The driver circuit establishes a connection with the STM32L100RCT6 microcontroller through four GPIO ports. The control of the stepper motor involves utilizing the PWM method by manipulating the logic signals. The hardware configuration of the curtain controller. The driver, through PWM control, governs the stepper motor's rotational direction, enabling it to turn clockwise (CW) or counter-clockwise (CCW). The motor's direction is instrumental in the opening or closing of the curtain [28].*

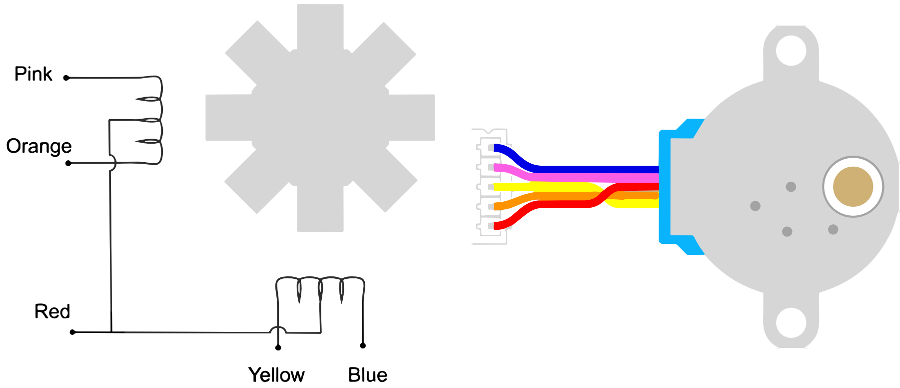


Figure 23: Stepper motor symbol and schematic [29].

Stepper motors employ a toothed wheel and electromagnets to incrementally advance the wheel through individual "steps." By energizing the coils in a specific sequence, the motor undergoes rotation. Interestingly, the count of steps in a 360-degree rotation corresponds to the number of teeth on the cog.In the case of the motor being utilized, it possesses 48 steps. However, the motor is equipped with a reduction gearbox with a ratio of 1:16. This implies that the total steps required for a full rotation amount to 16 times 48, resulting in 768 steps.

Step	***Blue***	***Pink***	***Yellow***	***Orange***	***Rotation Degree***
1	*1*	*0*	*0*	*0*	*5.625*
2	*1*	*1*	*0*	*0*	*11.25*
3	*0*	*1*	*0*	*0*	*16.875*
4	*0*	*1*	*1*	*0*	*22.5*
...	*...*	*...*	*...*	*...*	*...*
61	*0*	*0*	*0*	*1*	*343.125*
62	*0*	*0*	*1*	*1*	*348.75*

63	0	0	1	0	354.375
64	0	0	0	0	360

Table 2: ***Achieve a clockwise rotation in the stepper motor [29].***

Step	Blue	Pink	Yellow	Orange	Rotation Degree
1	0	0	1	0	-5.625
2	0	1	1	0	-11.25
3	0	1	0	0	-16.875
4	0	1	0	1	-22.5
…	…	…	…	…	…
61	1	0	0	0	-343.125
62	1	0	0	1	-348.75
63	1	0	1	1	-354.375
64	0	0	0	0	-360

Table 3: ***Achieve a anticlockwise rotation in the stepper motor [29].***

3.4 Assessment of Design

When assessing a project that uses augmented reality (AR) and the Internet of Things (IoT) to control devices, it is essential to consider various criteria to evaluate its success, effectiveness, and feasibility. Here are some assessment criteria to consider:

Functionality: *Assess the functionality of the AR and IoT system in controlling devices. Does the system provide intuitive and seamless control over the devices? Does it offer a wide range of control options and features? Evaluate how well the system meets the specific control requirements of the devices.*

User Experience: *Evaluate the user experience provided by the AR and IoT system. Consider factors such as ease of use, interactivity, responsiveness, and user satisfaction. A well-designed and user-friendly interface that enhances the overall experience is crucial for the success of the project [42].*

Device Compatibility: *Assess the compatibility of the AR and IoT system with different types of devices. Consider whether the system can control a wide range of devices, including various brands, models, and types. Compatibility with different communication protocols and standards should also be evaluated [43].*

Connectivity and Reliability: *Evaluate the connectivity and reliability of the AR and IoT system. Assess whether the system establishes stable and robust connections with the devices and maintains them consistently. Consider factors like network reliability, latency, and responsiveness of the system [44].*

Security: *Assess the security measures implemented in the AR and IoT system. Evaluate the authentication mechanisms, data encryption, access controls, and overall system security to ensure the protection of user data and prevent unauthorized access or tampering [45].*

Scalability: *Consider the scalability of the AR and IoT system. Evaluate whether the system can accommodate a growing number of devices, users, and interactions without significant performance degradation. Scalability is crucial for future expansion and the system's ability to handle increasing demands [46].*

Integration and Interoperability: *Assess the system's ability to integrate with existing infrastructure and other technologies. Evaluate whether the AR and IoT system can seamlessly integrate with other systems, platforms, or APIs, allowing for interoperability and leveraging existing resources.*

Maintenance and Support: *Evaluate the ease of maintenance and support for the AR and IoT system. Consider factors like system updates, bug fixes, troubleshooting procedures, and the availability of technical support channels. Assess the long-term sustainability and supportability of the project [47].*

Cost-effectiveness: *Assess the cost-effectiveness of implementing the AR and IoT system. Consider factors such as initial setup costs, ongoing maintenance costs, and potential return on investment. Evaluate whether the benefits provided by the system justify the associated costs [48].*

Regulatory Compliance: *Ensure that the AR and IoT system adheres to relevant regulatory requirements and standards. Evaluate whether the system meets necessary safety, privacy, and data protection regulations, depending on the nature of the controlled devices and user data involved [49].*

3.5 Relevant Engineering Standards and Codes

#	Engineering Standards and Codes
1	*IEC 61508 Functional Safety Standard*

The IEC 61508 standard, titled "Functional safety of electrical/electronic/programmable electronic safety-related systems," is a widely recognized international standard that provides guidelines for ensuring the functional safety of various systems, including those involving electrical, electronic, and programmable electronic components. It encompasses the entire lifecycle of a safety-related system, from concept and design to operation and maintenance. While the IEC 61508 standard focuses on functional safety rather than specific technologies like AR and IoT, it is still relevant to our project as it provides a framework for managing safety risks associated with the devices controlled by AR and IoT technologies [39]. By adhering to the principles and requirements outlined in IEC 61508, we can ensure that our AR and IoT-based device control system meets the necessary safety criteria [40]. Specifically, the IEC 61508 standard emphasizes the following aspects that can be related to our project:

1. *Risk Assessment: The standard emphasizes the need for a thorough risk assessment process, which involves identifying hazards, analyzing potential risks, and determining the required safety functions to mitigate those risks. In our project, we need to assess the potential risks associated with device control using AR and IoT technologies and implement appropriate safety measures.*

2. *Safety Integrity Levels (SIL): The standard defines four safety integrity levels (SIL 1 to SIL 4) that quantify the required level of risk reduction for safety-related systems.*

Depending on the criticality of the devices being controlled, we need to determine the appropriate SIL and design your AR and IoT system accordingly to achieve the required level of functional safety.

3. *Verification and Validation: IEC 61508 emphasizes the importance of rigorous verification and validation processes to ensure that the safety-related system functions as intended. This includes activities such as testing, inspection, and analysis to verify that the system meets the specified safety requirements. In our project, we need to perform thorough testing and validation of the AR and IoT-based device control system to ensure its safety and functionality [41].*

3.6 Implementation Plan

We will begin by constructing the IoT device and thoroughly testing its functionality. Next, we'll establish and initialize the Blynk server, ensuring seamless connectivity with the devices for smart operations. Moving forward, we'll utilize Unity to develop both the application and the accompanying 3D model. Finally, we'll integrate these components with the IoT devices and conduct comprehensive testing. This project can be outlined into four key stages:

1. *Hardware Construction and Testing (2 Weeks)*
2. *Blynk Server Initialization (1 Week)*
3. *Unity Application and 3D Model Development (2 Weeks)*
4. *Integration of Components and Testing (2 Weeks)*

This plan totals approximately two months to complete.

3.7 Manufactured on a Commercial Basis

The estimated number of devices sold per year depends on the demand of the customers, time of manufacturing and cost of manufacturing. Let us say that a capable company will manufacture our project and they have a good marketing section. The number of devices manufactured per year will be determined by the number of demands by the customers, so if the customers demand 100,000 devices then we will be manufacturing 100,000 devices and so on. The estimated price of device manufacturing will depend on our experience with the prices, since that our total result cost is $105.16 then in the manufacture it will be $80 - $300; because *we may remove or add some details, like remove LED and add AC as devices to control or perhaps we will some slight modifications to have a better product for customers. The estimated purchase price for our device will obviously depend on the manufacturing and marketing total price, so if the total is $200 then the purchase cost will be $350-$450. The estimated profit per year depends on the number of devices sold, so if we sell 1000 and the price is $400 and the manufacturing cost is 300$ for each device, then the total profit per year is $100,000. The estimated cost for the user to operate the device, per unit time is only the purchase price and if the device is damaged and needs some maintenance, if the purchase price is $400 and maintenance price is $50 so the total will be $450.*

3.8 Environmental

Manufacturing and deploying electronic devices present substantial environmental challenges, contributing to the growing issue of electronic waste (e-waste) due to inadequate recycling practices[52]. The considerable energy consumption of IoT devices, especially when consistently connected to the internet for data transfer, exacerbates these environmental impacts [53]. Additionally, the production of electronic devices relies on earth materials, creating a dual challenge of environmental harm and difficulties in obtaining these resources sustainably. This underscores the urgent need to embrace sustainable practices in both the development and application of these technologies. Integrating IoT sensors and devices into various applications offers a promising avenue to address environmental concerns by optimizing energy consumption, waste disposal, and enhancing services for more efficient resource utilization and improved overall quality of life[54]. On a broader scale, our project serves as a catalyst for promoting sustainability in both manufacturing and energy consumption, empowering decision-makers and individuals with valuable insights to achieve tangible reductions in waste and energy consumption. A challenge hindering progress in sustainable IoT development is the lack of uniformity and compatibility among devices, leading to inefficiencies and increased waste. Our project proactively tackles this challenge by establishing a unified platform, ensuring compatibility, and mitigating potential inefficiencies and waste. While the Internet of Things (IoT) is a powerful tool for fostering sustainable practices and minimizing environmental impacts, realizing its full potential requires a concerted effort to overcome existing obstacles. Designing and deploying IoT devices with *sustainability at the forefront is essential to reducing their environmental footprint and steering towards a more sustainable future [55]. The commitment to sustainable practices ensures that the benefits of IoT technologies align with environmental improvements, paving the way for a greener and more sustainable tomorrow.*

3.9 Ethical

Before undertaking any project, it's imperative to thoroughly address ethical considerations, particularly regarding user privacy. In our scenario, a significant concern arises due to the potential invasion of privacy resulting from the camera's image capture. This concern is particularly relevant in societies where individuals may be uncomfortable with their images being captured without consent. Balancing the camera's ability to provide valuable information with the risk of infringing upon individuals' privacy underscores the importance of securely handling data captured by AR and IoT technologies to mitigate privacy risks.

3.10 Health and Safety

Before beginning any project, it is imperative to carefully evaluate health and safety issues, particularly with regard to device automation utilizing augmented reality (AR) and the Internet of Things (IoT). In our situation, a noteworthy concern arises with respect to possible threats to physical health and security as a result of automated procedures. This is especially important in settings where user interaction with automated devices may present risks. Weighing the advantages of automation against user safety emphasizes the significance of incorporating safety features and user-friendly interfaces in AR and IoT systems. Moreover, strong data security protocols and reducing environmental impact are also important factors to take into account when addressing health and safety concerns.

3.11 Social and Political

In the Design and Implementation of Device Automation using Augmented Reality and Internet of Things, social and political considerations are crucial:

3.11.1 Impact: *Our idea may disrupt job roles and face cultural resistance, affecting societal acceptance.*

3.11.2 Stakeholders: *Direct stakeholders include developers, manufacturers, users, and regulators. Indirect stakeholders involve local communities, and businesses.*

3.11.3 Benefits and Harms: *While AR and IoT technologies can enhance productivity and provide convenience, they may also raise concerns about job displacement and economic insecurity for certain individuals.*

3.11.4 Geographical Impact: *In Saudi Arabia, the effects could be different between rural areas and cities because of variations in technology availability and infrastructure.*

Chapter 4: Implementation Process and Testing

4.1 Software Development Platform

In our project we used several platforms, and these platforms are:

1- Unity: *is a versatile game engine used to create interactive experiences for various platforms like games, VR, AR, and simulations, and we used it to initiate AR models to control our devices.*

2- Visual Studio 2022: *is an IDE by Microsoft for software development. It offers tools for code editing, debugging, and project management, supporting multiple programming languages and platform, and we used it to write the C# scripts for each target image so that we can make the virtual buttons functionable.*

3- Vuforia*: is an AR development platform for creating AR applications on smartphones, tablets, and smart glasses. It offers tools for detecting, tracking, and rendering virtual content in the real world, and we used it as a database so that we can use our target images, also it allows us to insert and use the virtual buttons.*

4- Blynk: *is a platform for developing mobile apps that control and monitor IoT devices. It offers a user-friendly interface and tools to connect IoT hardware with mobile devices, and we used it for our IoT part of our project devices. It allowed us to control our devices using ESP32 and WIFI.*

4.2 Code Design

In our project, we utilized two prominent programming languages C++ and C#. C++ was employed for the IoT aspect due to the Arduino IDE's reliance on it, whereas C# was chosen for the AR component, as Unity utilizes C# for programming virtual buttons. For virtual buttons we need an API that sends data from unity to the blynk server to perform some action.

4.2.1 ESP32 code:

```
#define BLYNK_TEMPLATE_ID           "TMPL61h0AooTF"
#define BLYNK_TEMPLATE_NAME         "Quickstart Template"
#define BLYNK_AUTH_TOKEN            "LxTJ-5uaHH_Qyp5X94a0SgmNyKeplKXF"

#include <AccelStepper.h>
#include <BlynkSimpleEsp32.h>

const char auth[] = BLYNK_AUTH_TOKEN;         // Blynk authentication token
const char ssid[] = "**************";         // Your WiFi SSID
const char pass[] = "**************";         // Your WiFi password

const int speedPin = 2;     // Variable that control the speed of fan
const int LEDPin = 22;      // Variable that control the LED
const int FANPowerPin = 4; // Variable that switch pn/off fan


int currentPosition = 0;    // Variable that keep track the curnt position of the motor
#define UPPER_LIMIT 13500   // Maximum upper limit
#define LOWER_LIMIT 0       // Maximum lower limit

// Flags indicator
bool clockwiseFlag = false;
bool counterclockwiseFlag = false;

// ULN2003 Motor Driver Pins
#define IN1 5
#define IN2 18
#define IN3 19
#define IN4 21

// Blynk virtual pins for the buttons
#define buttonClockwisePin V1          // Down
#define buttonCounterclockwisePin V0   // Up
#define buttonStepperOFFPin V3   // Up
#define LED V4                         // Led
#define FAN V2                         // FAN

// initialize the stepper library
AccelStepper stepper(AccelStepper::HALF4WIRE, IN1, IN3, IN2, IN4);

void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  pinMode(LEDPin, OUTPUT);
  pinMode(speedPin, OUTPUT);
  pinMode(FANPowerPin, OUTPUT);
```

```
  // set the speed and acceleration
  stepper.setMaxSpeed(2000);
  stepper.setAcceleration(1000);
}

void loop() {
  Blynk.run();
  // Move the stepper motor (one step at a time)
  stepper.runSpeed();
    if (clockwiseFlag) {
    if (currentPosition < UPPER_LIMIT) {
      stepper.setSpeed(-1000);   // Set speed for clockwise rotation
      stepper.enableOutputs();  // Enable motor outputs
      stepper.runSpeed();       // Move the motor
      delay(2);
      currentPosition++;
    } else {
      stepper.setSpeed(0);       // Set speed to 0 to stop the motor
      stepper.disableOutputs();  // Disable motor outputs
    }
  } if (counterclockwiseFlag) {
    if (currentPosition > LOWER_LIMIT) {
      stepper.setSpeed(1000);  // Set speed for counterclockwise rotation
      stepper.enableOutputs();  // Enable motor outputs
      stepper.runSpeed();       // Move the motor
      delay(2);
      currentPosition--;
    } else {
      stepper.setSpeed(0);       // Set speed to 0 to stop the motor
      stepper.disableOutputs();  // Disable motor outputs
    }
  } if(counterclockwiseFlag == false && clockwiseFlag == false) {
      stepper.setSpeed(0);       // Set speed to 0 to stop the motor
      stepper.disableOutputs();  // Disable motor outputs
    }
}

BLYNK_WRITE(buttonClockwisePin) {
  // Set incoming value from pin V4 to a variable
  int buttonClockwiseState = param.asInt();
  if (buttonClockwiseState == HIGH) {
    clockwiseFlag = true;
    counterclockwiseFlag = false;
  } else {
    clockwiseFlag = false;
  }
```

```
}

BLYNK_WRITE(buttonCounterclockwisePin) {
  // Set incoming value from pin V1 to a variable
  int buttonCounterclockwiseState = param.asInt();
  if (buttonCounterclockwiseState == HIGH) {
    counterclockwiseFlag = true;
    clockwiseFlag = false;
  } else {
    counterclockwiseFlag = false;
  }
}

BLYNK_WRITE(buttonStepperOFFPin) {
  // Set incoming value from pin V1 to a variable
  int buttonStepperOFFState = param.asInt();
  if (buttonStepperOFFState == HIGH) {
    clockwiseFlag = false;
    counterclockwiseFlag = false;
    stepper.setSpeed(0);       // Set speed to 0 to stop the motor
    stepper.disableOutputs();  // Disable motor outputs
  }
}
BLYNK_WRITE(LED) {
  // Set incoming value from pin V4 to a variable
  int value = param.asInt();
  digitalWrite(LEDPin, value);
}

BLYNK_WRITE(FAN){
  // Set incoming value from pin V2 to a variable
int value = param.asInt();
 if(value <= 0){
  digitalWrite(FANPowerPin, LOW);
 }
 else{
  digitalWrite(FANPowerPin, HIGH);
  // Set the fan speed using PWM
  analogWrite(speedPin, value);
 }
}
```

4.2.2 Target images codes:

1- Fan target image code:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;
using UnityEngine.Networking;

public class Fan : MonoBehaviour
{
    public VirtualButtonBehaviour Vb_high;
    public VirtualButtonBehaviour Vb_medium;
    public VirtualButtonBehaviour Vb_low;
    public VirtualButtonBehaviour Vb_off;
    public string url_high;
    public string url_medium;
    public string url_low;
    public string url_off;

    IEnumerator GetRequest(string uri)
    {
        using (UnityWebRequest webRequest = UnityWebRequest.Get(uri))
        {
            // Request and wait for the desired page.
            yield return webRequest.SendWebRequest();

        }
    }
    void Start()
    {
        Vb_high.RegisterOnButtonPressed(HighButtonPressed_high);

        Vb_medium.RegisterOnButtonPressed(MediumButtonPressed_medium);

        Vb_low.RegisterOnButtonPressed(LowButtonPressed_low);

        Vb_off.RegisterOnButtonPressed(OnButtonPressed_off);

    }
    public void HighButtonPressed_high(VirtualButtonBehaviour Vb_high)
    {
        StartCoroutine(GetRequest(url_high));
        Debug.Log("FAN IS HIGH");
    }
    public void MediumButtonPressed_medium(VirtualButtonBehaviour Vb_medium)
    {
        StartCoroutine(GetRequest(url_medium));
        Debug.Log("FAN IS MEDIUM");
    }
    public void LowButtonPressed_low(VirtualButtonBehaviour Vb_low)
    {
        StartCoroutine(GetRequest(url_low));
        Debug.Log("FAN IS LOW");
    }

    public void OnButtonPressed_off(VirtualButtonBehaviour Vb_off)
    {
        StartCoroutine(GetRequest(url_off));
        Debug.Log("FAN IS OFF");
    }
}
```

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;
using UnityEngine.Networking;

public class LED : MonoBehaviour
{
    public VirtualButtonBehaviour Vb_on;
    public VirtualButtonBehaviour Vb_off;
    public string url_on;
    public string url_off;

    IEnumerator GetRequest(string uri)
    {
        using (UnityWebRequest webRequest = UnityWebRequest.Get(uri))
        {
            // Request and wait for the desired page.
            yield return webRequest.SendWebRequest();

        }
    }

    void Start()
    {
        Vb_on.RegisterOnButtonPressed(OnButtonPressed_on);

        Vb_off.RegisterOnButtonPressed(OnButtonPressed_off);

    }


    public void OnButtonPressed_on(VirtualButtonBehaviour Vb_on)
    {
        StartCoroutine(GetRequest(url_on));
        Debug.Log("LED IS ON");
    }

    public void OnButtonPressed_off(VirtualButtonBehaviour Vb_off)
    {
        StartCoroutine(GetRequest(url_off));
        Debug.Log("LED IS OFF");
    }

}
```

3- Motor target image code:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;
using UnityEngine.Networking;

public class Motor : MonoBehaviour
{
    public VirtualButtonBehaviour Vb_up;
    public VirtualButtonBehaviour Vb_down;
    public VirtualButtonBehaviour Vb_off;
    public string url_up;
    public string url_down;
    public string url_off;

    IEnumerator GetRequest(string uri)
    {
        using (UnityWebRequest webRequest = UnityWebRequest.Get(uri))
        {
            // Request and wait for the desired page.
            yield return webRequest.SendWebRequest();

        }
    }

    void Start()
    {
        Vb_up.RegisterOnButtonPressed(OnButtonPressed_up);
        Vb_down.RegisterOnButtonPressed(OnButtonPressed_down);
        Vb_off.RegisterOnButtonPressed(OnButtonPressed_off);

    }


    public void OnButtonPressed_up(VirtualButtonBehaviour Vb_up)
    {
        StartCoroutine(GetRequest(url_up));
        Debug.Log("UP IS ON");
    }
    public void OnButtonPressed_down(VirtualButtonBehaviour Vb_down)
    {
        StartCoroutine(GetRequest(url_down));
        Debug.Log("DOWN IS ON");
    }

    public void OnButtonPressed_off(VirtualButtonBehaviour Vb_off)
    {
        StartCoroutine(GetRequest(url_off));
        Debug.Log("STEPPER IS OFF");
    }

}
```

4.2.3 API URLs:

1- Fan URLs:

High speed signal URL:
https://blynk.cloud/external/api/update?token=LxTJ5uaHH_Qyp5X94a0SgmNyKeplKXF&v2=250

Medium speed signal URL:
https://blynk.cloud/external/api/update?token=LxTJ-5uaHH_Qyp5X94a0SgmNyKeplKXF&v2=125

Low speed signal URL:
https://blynk.cloud/external/api/update?token=LxTJ-5uaHH_Qyp5X94a0SgmNyKeplKXF&v2=25

OFF signal URL:
https://blynk.cloud/external/api/update?token=LxTJ-5uaHH_Qyp5X94a0SgmNyKeplKXF&v2=0

2- LED URLs:

ON signal URL:
https://blynk.cloud/external/api/update?token=LxTJ-5uaHH_Qyp5X94a0SgmNyKeplKXF&v4=1

OFF signal URL:
https://blynk.cloud/external/api/update?token=LxTJ-5uaHH_Qyp5X94a0SgmNyKeplKXF&v4=0

3- Motor URLs:

Up signal URL:
https://blynk.cloud/external/api/update?token=LxTJ-5uaHH_Qyp5X94a0SgmNyKeplKXF&v1=1

Down signal URL:
https://blynk.cloud/external/api/update?token=LxTJ-5uaHH_Qyp5X94a0SgmNyKeplKXF&v0=1

OFF signal URL:
https://blynk.cloud/external/api/update?token=LxTJ-5uaHH_Qyp5X94a0SgmNyKeplKXF&v3=1

4.3 Verification

Our project has successfully met all the specified criteria. After a verification process that involved testing, analyzing user feedback, checking compatibility, and ensuring functionality, we have verified that the system aligns with the project goals. Real-time data updates have been seamlessly integrated, triggering immediate responses whenever there are data updates. We rigorously tested compatibility with devices to confirm that the system can smoothly work with various devices while maintaining an easy-to-use interface, supported by positive *feedback from users. Functional tests have proven that the system can effectively control devices through AR interfaces, simplifying device management based on user testing results. Moreover, compatibility tests conducted on smartphones and tablets showed performance across a range of devices without any compatibility issues arising. With these verification outcomes, we are confident that all project requirements have been successfully met. Our system is now poised to provide users with real-time data insights, smooth device control, and intuitive interfaces to enhance their interaction with devices through AR experiences. Going forward, monitoring, upkeep, and user input will be crucial in ensuring our solution continues to perform and meet user expectations.*

4.4 Validation

In general our hardware parts worked perfectly, and these are the details:

4.4.1 Specification:

1- ESP32: we used it to control our devices via WIFI, and it worked correctly.

2- Relay: we used it to connect the ESP32 with high voltage devices like the 12V LED and fan, and it worked perfectly .

3- Jumper wires: we used them to connect our devices with the ESP32,and we faced no issue making them work perfectly.

4- LED: we used it as a main controllable device, and it worked as the specification suggested, it lights when we press ON and it stops when we press OFF, it is working correctly..

5- Stepper motor: we used it as a main part to make a controllable horizontal curtain, and it worked perfectly with the speed and limits of rotation that we used in code.

6- Fan: it is also one of the three main controllable devices, and we could use it for our purposes with no issues, but some fans does not stop when we send value=0 to the PWM fan pin, so we needed to connect it to a relay so that we can turn it off using ESP32.

7- Breadboard: we used it as a base to construct our circuit, and it worked correctly with no issues.

8- Battery: we used it to provide power to the 12V LED and the fan, and it is working correctly as it supposed to be.

4.4.2 Testing:

1- Unit testing:

To start unit testing, we first examined how well every part works along with there software on the Blynk platform. We tested each component thoroughly to ensure that it can work properly independently.

We first set up a Blynk project applying the LED widget and enabled control via the Blynk app or web interface in order to perform unit testing for a LED using Blynk. Then, test cases were created to evaluate multiple aspects of LED functionality, such as turning them on and off. We created firmware to communicate with the platform after connecting the LED to the ESP32 using Blynk. This allowed the LED to react to commands from the Blynk app. The test was executed by commanding the device from the app, watching how the LED behaved, and comparing the results to expectations.

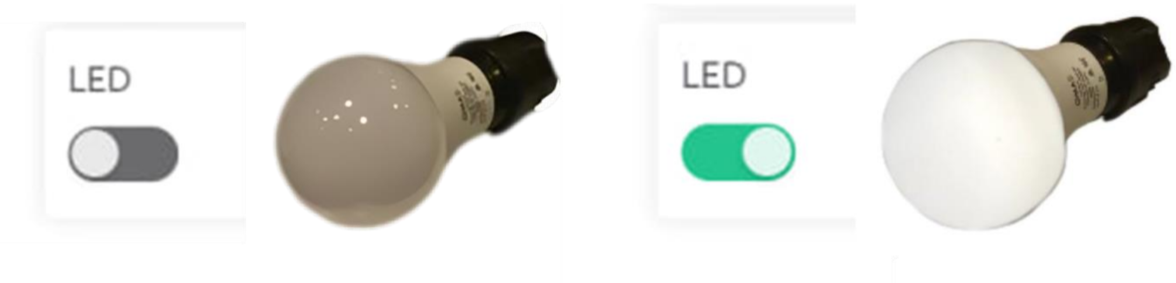


Figure 24. LED testing.

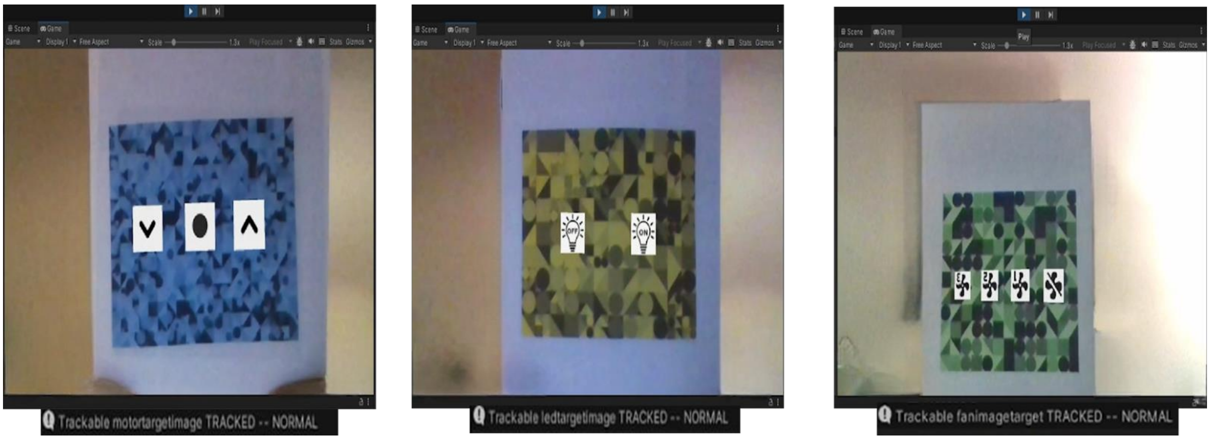
Similar to the LED test we started a Blynk project with a slider widget set up to remotely change the fan's speed using the Blynk app or web interface in order to do unit testing for a fan. Carefully thought-out test scenarios were created to assess the fan's performance at various speeds and under various operating circumstances. Firmware was created to establish communication with the platform after the fan was connected to the ESP32 microcontroller *that is compatible with Blynk. This allowed for the dynamic modification of the fan speed based on slider input.*

Figure 25. Fan testing.

As with the LED and fan tests, we started a Blynk project that included three widgets set up to remotely change the direction of a stepper motor: one widget moved the motor up, another down, and another for turning it off using the Blynk app or via the web browser. To assess the stepper motor's performance in directions and operating circumstances, we created test scenarios. Afterwards, firmware was created in order to connect to the platform after the stepper motor was connected to an ESP32, which communicates with Blynk. This allowed us to dynamically modify the motor's movement in response to input from the widgets.

Figure 26. Stepper motor testing.

Also, a unit test has been done in Unity to ensure the accuracy of image targets and AR tracking functionalities. This test was important to see if the AR virtual objects interact correctly to the real world with no errors.

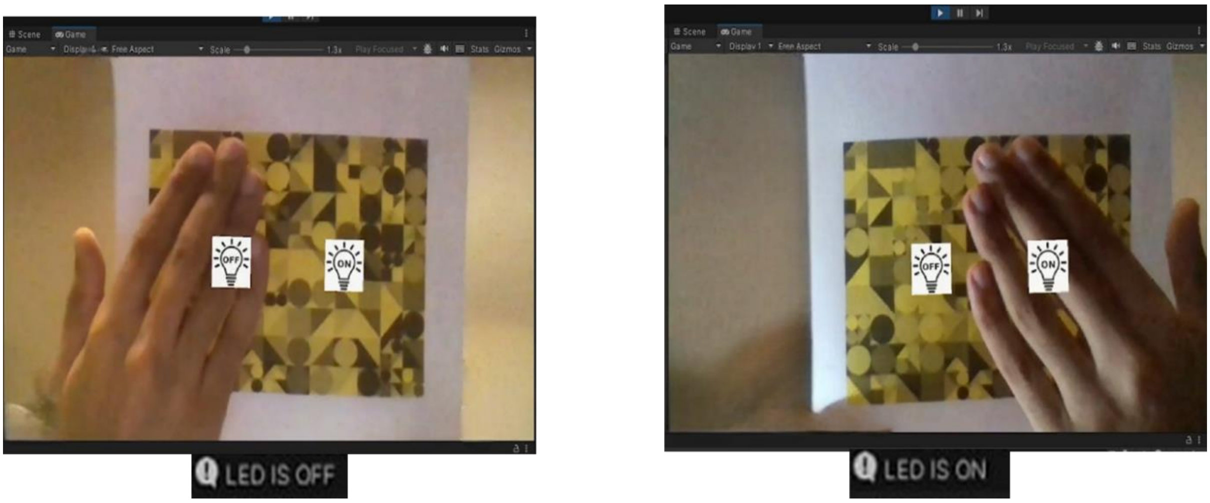


(a) ***(b)*** ***(c)***

Figure 27. (a) Motor Image target tracking, (b) LED target tracking, (c) fan Image target tracking.

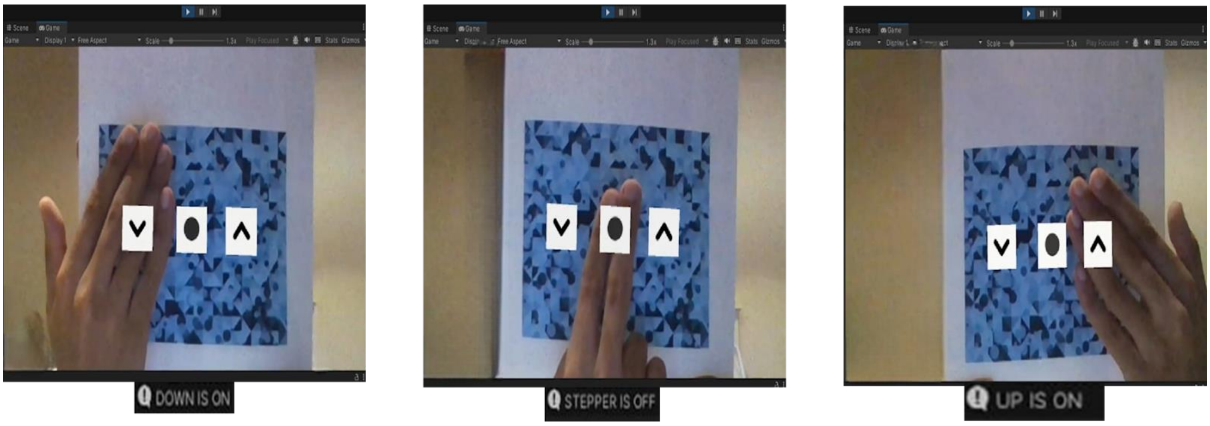
2- Integration testing:

After unit testing was successful, we moved on to the next step of testing, which is integration testing. We combined all the equipment that is in charge of controlling IoT devices together in Blynk to check if there are no problems and they are working smoothly. Similar to in Unity, an integration test has been done to ensure all objects and functions when grouped work as intended. Then we moved on to the most important test for our project, which involves testing the interaction between AR elements and IoT devices, or, in other words, the interaction between Unity and Blynk. To do an integration test between Unity and Blynk, first we need to create an AR scene in Unity that contains all the elements we need that allow users to interact with the real devices. To enable communication between Unity and Blynk, we need a script in each target image that contains a URL that interfaces with Blynk, which results in updating the state of the IoT device. During the test, we can ensure that user gestures translate correctly to Blynk commands and reflect on devices as desired. All three tests for the LED, fan, and curtain follow the same procedure.



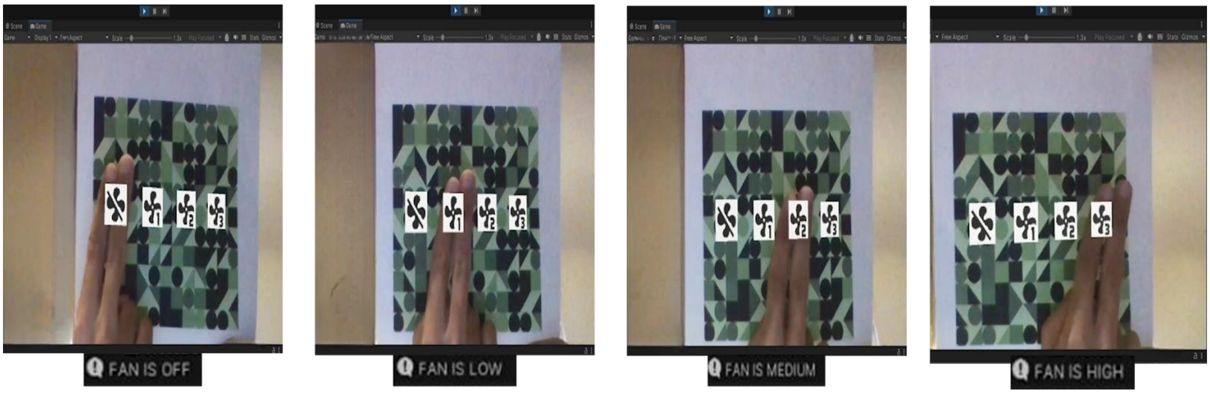
(a) ***(b)***

Figure 28. AR and IoT testing of LED states (a) OFF state, (b) ON state.



(a) ***(b)*** ***(c)***

Figure 29. AR and IoT testing of stepper motor states (a) DOWN state, (b) OFF state, (c) UP state.

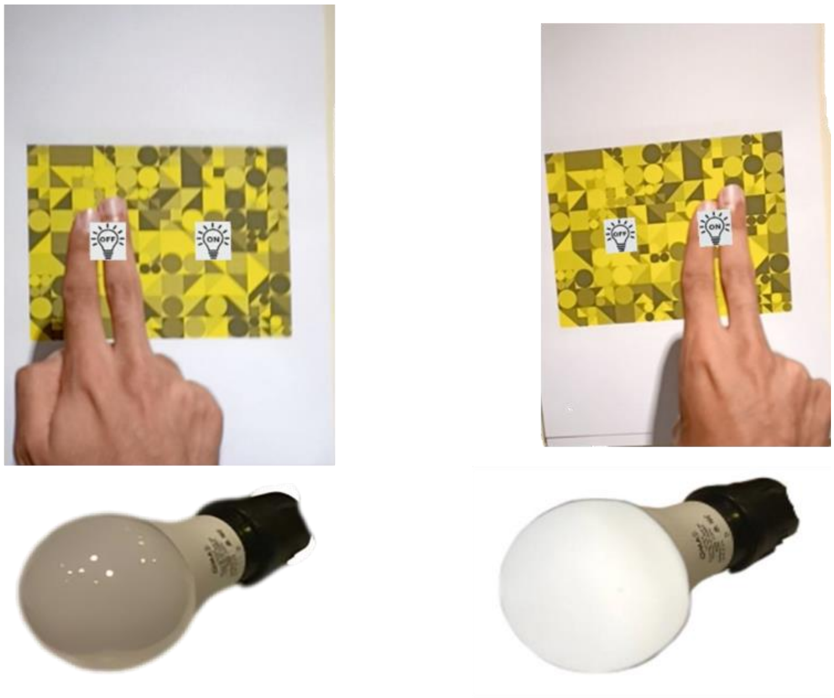


(a) *(b)* *(c)* *(d)*

Figure 30. AR and IoT testing of fan states (a) OFF speed state, (b) LOW speed state, (c) MEDIUM speed state, (d) HIGH speed state.

3- System testing:

A system test has been done to check that all aspects of the project work as intended towards achieving the functionality of the project and providing an intuitive, user-friendly experience. This test combines all the previous tests that have been done.



(a) *(b)*

Figure 31. Application testing of LED states (a) OFF state, (b) ON state.

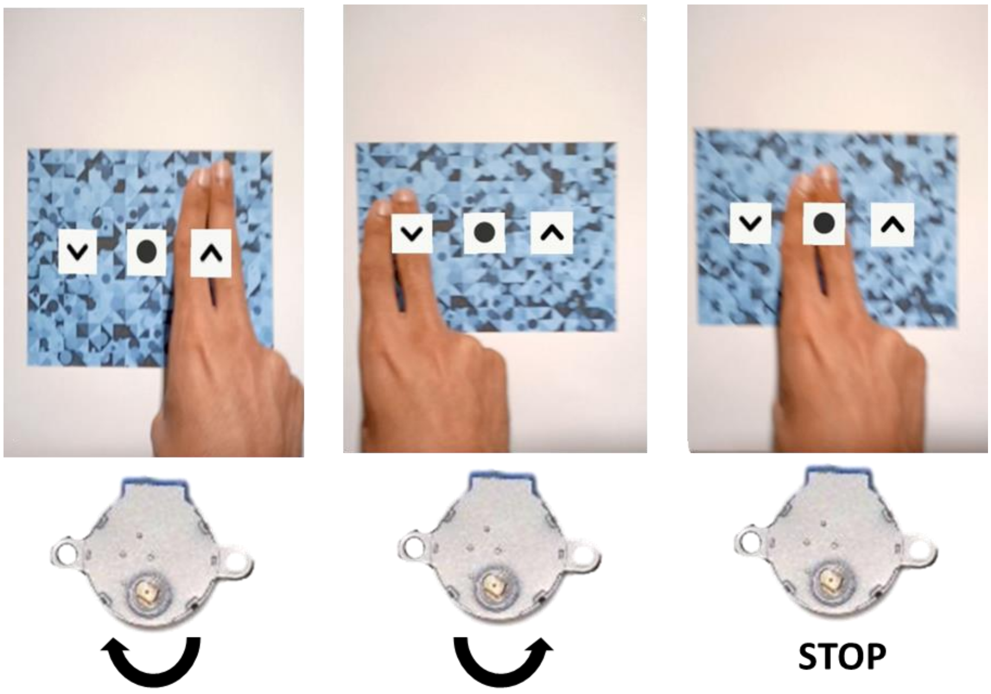


Figure 32. Application testing of stepper motor states (a) UP state, (b) DOWN state, (c) OFF state.

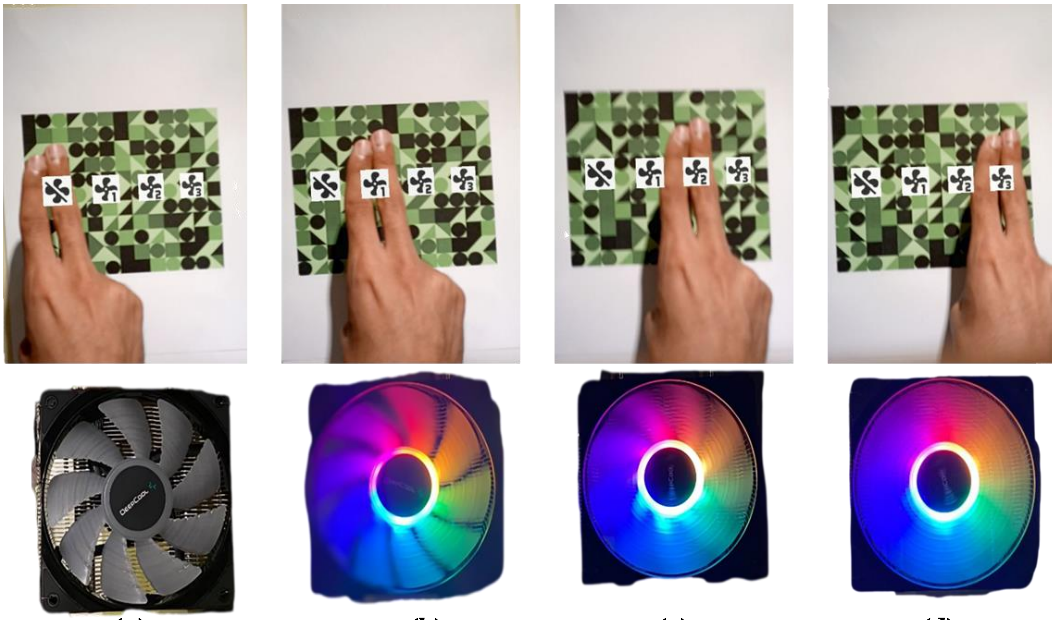


Fig 33. Application testing of fan states (a) OFF speed state, (b) LOW speed state, (c) MEDIUM state, (d) HIGH speed state.

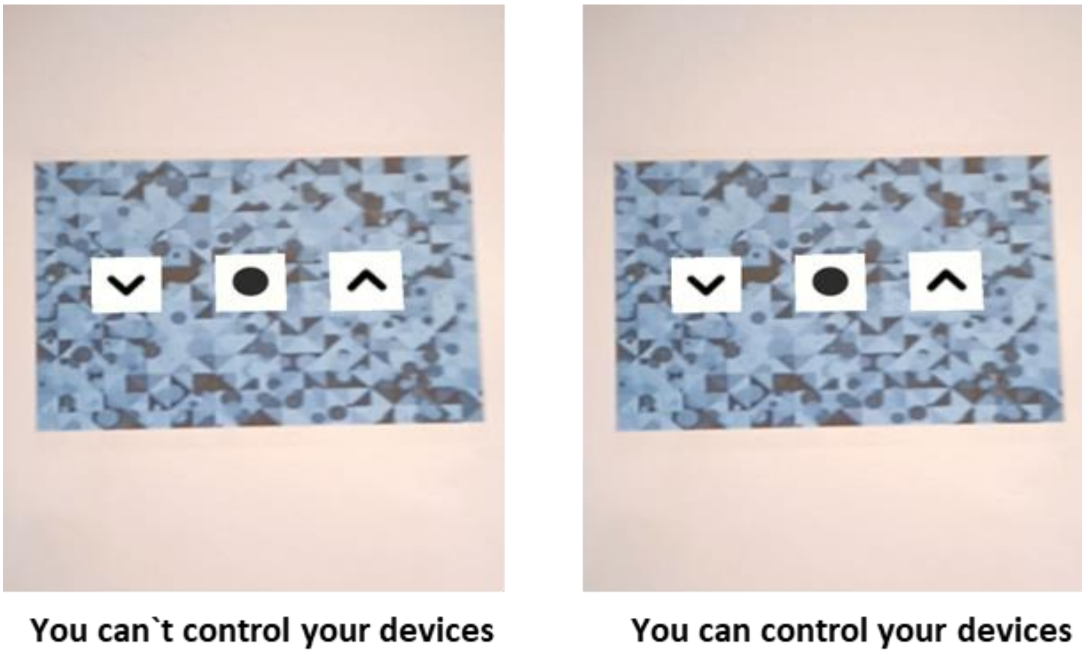
4- Permission testing:

Our app requires a camera to detect target images and read gestures from the user. Fig [30] illustrates two test cases. one that prevented camera access and another that allow camera access. When the camera is allowed, the app functions as expected with no problems, but when the permission is denied, the app will handle it without crashing and display a message that you need camera permission to run this app.

(a) ***(b)***

Figure 34. Camera Permission testing, (a) preventing camera access, (b) allowing camera access.

A network permission test has been done to check how the app will respond to two test cases. One that prevented network access and another that allowed network access. When the network is allowed, the app functions as expected with no problems, but when the permission is denied, the app will detect target images and display the virtual objects without being able to control the devices (see Fig 35).

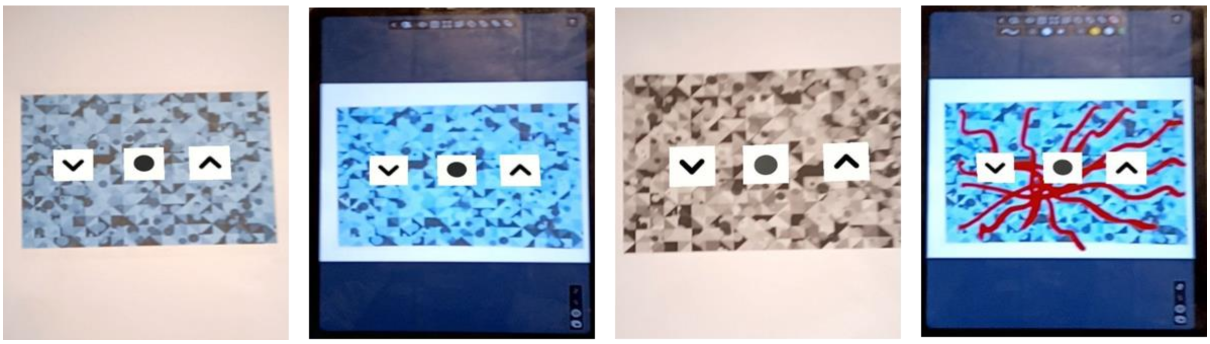


(a) *(b)*

Figure 35. Network Permission testing 2, (a) preventing network access, (b) allowing network access.

5- Performance testing:

Lastly, performance testing has been done to check how accurately the system will track the target image in four different scenarios: hard copy tracking, soft copy tracking, grayscale copy tracking, and soft copy with some noise tracking. and the result indicates that the system was able to continuously track accurately in all four scenarios (see Fig 36).



(a) *(b)* *(c)* *(d)*

Figure 36. Image target testing 2, (a) hard copy tracking, (b) soft copy tracking, (c) grayscale copy tracking, (d) soft copy with some noise tracking.

4.5 Evaluation

4.5.1 Compare to other systems

If we want to compare our system with other systems, we can see that there are just a few systems that use the IoT and AR concepts as our project, our project is characterized by more accuracy, ease of use and diversity than most of other projects related to IoT and AR, also there is no other project in the internet that looks the exact same project as ours; because we putted a lot of effort to make it different when it is compared to other projects.

4.5.2 Qualitative assessment of performance

We can measure qualitative assessment of performance by these 5 main standards:

1- User Experience: *the user experience plays a crucial role in determining the success of AR and IoT applications. The project should be evaluated based on how intuitive and seamless the interaction is between the AR interface and the physical devices. Factors such as the responsiveness of controls, ease of setup, and overall user satisfaction should be considered [56].*

2- Reliability: *the reliability of the system is essential to ensure consistent and error-free operation. The project should be assessed based on the stability of the AR and IoT components, including the stability of the connection between the AR device and the IoT devices [57].*

3- Latency: *refers to the delay between a user's action in the AR interface and the corresponding response from the physical devices. Low latency is crucial for providing a realistic and immersive experience. The project should be evaluated based on the perceived latency during different interactions, such as turning on/off the LED light, controlling the stepper motor, and adjusting the fan speed [58].*

4- Integration: *the successful integration of AR and IoT technologies is a key aspect of this project. The assessment should consider how effectively the AR interface communicates with the IoT devices, ensuring accurate and synchronized control. The compatibility of the project with different IoT platforms and protocols can also be considered [59].*

5- Scalability: *refers to the project's ability to handle an increasing number of connected devices and users. The assessment should consider how well the project can handle multiple LED lights, stepper motors, and fans simultaneously, without compromising performance or user experience [60].*

In conclusion, all these standards applied to our project in a good manner to make it work in an efficient and enjoyable way.

4.6 Economic

1- Human Capital: *Skilled individuals in electronics, programming, and possibly mechanical engineering are required for the project.These individuals contribute their expertise to design, implement, and test the device automation system.*

2- Financial Capital: *The project requires costs for component parts, software licenses, labor expenses, and any additional equipment needed.Financial capital is essential for purchasing materials, paying for licenses, and compensating team members for their time and expertise.*

3- Manufactured or Real Capital: *This includes physical components like the ESP32, relay, LED, stepper motor, and fan. The reliability and quality of these components impact the overall functionality and success of the project.*

4- Natural Capital: *While the project doesn't directly consume natural resources, it indirectly relies on the availability of resources for manufacturing components and powering devices. Energy consumption during the operation phase may have indirect environmental impacts.*

5- Costs and Benefits Accrual: *Costs accrue throughout the project's lifecycle, including component parts, and software licenses. Benefits appear.as the project progresses, such as improved automation capabilities, energy efficiency, and potential market value of the final product.*

6- Inputs and Costs: *The experiment requires various electronic components, software licenses, and possibly additional equipment for development. The project's cost includes the sum of component parts, software licenses, and any additional equipment costs.*

7- Earnings and Profits:*The project can generate income through device sales, technology licensing fees, or automation cost savings.Profitability depends on its innovative technology, with returns going to the team, stakeholders, or investors based on the business model and market acceptance.*

8- Timing: *The duration of products' existence depends on market demand, technological advancements, and product lifecycle management. Maintenance and operation costs may include software updates, component replacements, and technical support. These costs should be considered to ensure the long-term viability and sustainability of the project. The original estimated development time should be outlined in a Gantt or Pert chart at the start of the project.*

4.7 Manufacturability

Our project consists of two distinct parts. which are the IoT part and AR part. One challenge associated with manufacturing the IoT component is ensuring smooth integration of various electronic devices, such as the ESP32, relay, LED, stepper motor, and fan, into a functional system. Additionally, ensuring compatibility with existing IoT infrastructure such as blynk server. Similarly, manufacturing the AR component involves challenges related to software development and integration. we need to ensure compatibility with existing IoT infrastructure and it can provide the proper functionality.

Chapter 5: Discussion the Results

5.1 Summary of Work

Our system allows us to control a fan, LED lamp and stepper motor using AR gestures. In order to make the system functionable we constructed the circuit required for our system. This likely involved connecting the necessary components such as the fan, LED lamp, and stepper motor to the appropriate pins on our microcontroller board (ESP32).

After constructing the circuit, we proceeded to write the code for the ESP32 microcontroller. This code would be responsible for controlling the various components based on input received from the AR gestures. Once the code was written, we installed it onto the ESP32 board.

To enable the AR gesture recognition, we utilized Vuforia as a database to store your target images. By saving our target images in the Vuforia database, we set the foundation for recognizing and interacting with these images in our AR application.

In order to integrate the Vuforia database with our AR application, we connected it to Unity. By establishing a connection between Vuforia and Unity, we gained access to Vuforia's image recognition capabilities within our Unity project.

With the connection between Vuforia and Unity in place, we proceeded to design virtual buttons for each target image. These virtual buttons would serve as the interface for triggering actions related to the fan, LED lamp, and stepper motor.

To implement the desired functionality, we wrote C# scripts within Unity. These scripts would define the behavior and interactions associated with each virtual button and the corresponding target image. Once the scripts were written, we uploaded them to each target image within our AR application.

Finally, we conducted thorough testing to ensure the proper functioning of your entire system. This involved testing the AR gestures, verifying the control of the fan, LED lamp, and stepper motor through the virtual buttons, and making any necessary adjustments to optimize *performance. Once testing was completed and any issues were resolved, our system was considered finalized and ready for use.*

5.2 Development

On the journey of creating our project, we encountered many new tools, the most important of which are Blynk, Unity, and Vuforia, which led us to the technique of controlling IoT devices through AR. We did a literature search to find the best way to stay cutting-edge and improve overall performance and experience for our project. Check Section 2.1 for more details. Each tool mentioned earlier has a specific role in our project. Starting with Blynk, it is an IoT platform designed for web, iOS, and Android smartphones. Used to establish communication between devices like Arduino, Raspberry Pi, and ESP32 over the Internet and other hardware components [16]. Unity is a popular cross-platform game engine and development tool used for creating interactive 2D and 3D content, including video games, simulations, and other real-time experiences. It provides a comprehensive set of tools, a visual editor, and a powerful scripting API that allows developers to design and build games for various platforms, such as desktop, mobile, consoles, and virtual reality (VR) devices [21]. Vuforia enables developers to build AR experiences by recognizing and tracking images, objects, and environments in the real world and overlaying digital content on top of them. It also utilizes computer vision technology to detect and track visual targets, enabling virtual content to be effortlessly integrated with the physical world [17]. It was also used in our project as a database for the image targets.

5.3 Critical Appraisal of Work

The negative aspects of our project include the application's inactivity when there's no internet connection. Additionally, the limited options for widget controls in Unity software can result in functional constraints when selecting the appropriate widget for specific devices. Primarily, our project's main drawback lies in its mobile nature, as AR glasses offer a superior user experience compared to mobile applications. Furthermore, a critical aspect of AR projects is the image target, which must adhere to specific requirements such as size, colors, and shape. This adherence can directly impact project performance, as some image targets exhibit low response tracking, while others do not.

5.4 Proposal for Enhancement or Re-design

We suggest that there are a lot of other controllable devices can be added or removed to this system, also there are some options to use some different components in the hardware part that can improve and optimize the system, and some of the software issues we faced in AR part using Unity and Vuforia restricted us from increasing the usability of our system, but in the unreleased newer versions of these softwares, these issues will be fixed and the system can be improved. Finally the user interface can be enhanced to also improve the usability of the system.

5.5 Sustainability

Maintaining the completed device automation system may affect challenges related to software updates or the wearing out of hardware components, for example, when the battery runs out. In addition, with software updates, we need to ensure compatibility between platforms. Furthermore, the potential issues with sustainability may be impacted by the need for ongoing user training and support. Offering virtual experiences can enhance user satisfaction beyond what physical experiences can achieve, leading to enhanced sustainability outcomes. One potential solution to solve environmental issues is to integrate IoT sensors and devices into multiple applications. This can optimize energy usage, waste disposal, and enhance services for better resource use [54]. One obstacle slowing the advancement of sustainable IoT development is the inconsistency and incompatibility of devices, resulting in inefficiencies and increasing waste. Through the establishment of an integrated system, compatibility, and minimizing of potential inefficiencies and waste, our project actively addresses this obstacle. To improve the sustainability of the system, we suggest adding different hardware components in addition to the modification of different controlled devices in the system. Thus, adding more controlled devices may result in compatibility issues in both hardware and software, not to mention an increase in costs.

5.6 Challenges And Overcome

In this project we certainly faced some challenges especially with the hardware part of the project.

First of all, when we started constructing our circuit we were going to use Arduino UNO with ESP32, then we faced some hurdles during this operation, so we tried to construct our circuit without using the Arduino UNO, so the result was perfect and the whole system worked properly, and by doing this we saved more resources and constructing the circuit was much easier with same target achieved.

Another challenge we faced in connecting the LED lamp and fan to our circuit is that they need 12V to run properly, and the ESP32 cannot handle this amount of electricity, so we added two relays, one for the fan and another for the LED lamp.

Also a challenge we faced is that we could not control the fan speed, so we added L298N Motor Driver Module to control the speed of the fan, then we found that the fan is still just turning on and off without controlling the speed, so we removed the L298N Motor Driver Module and connected a wire from ESP32 pin to the fan PWM pin, and this allowed us to control the speed of the fan.

Also one of the possible challenges is that some fans will not stop when we send value (0) to the PWM pin, and this is also fixed by adding the relay between the fan and ESP32.

Qassim University
College of Computer | كلية الحاسب

References

[1] Sinha, S. (2023, May 24). State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally [Online]. Available: IoT Analytics. https://iot-analytics.com/number-connected-iot-devices/

[2] J. S. Yun, I. Y. Ahn, N. M. Sung, and J. H. Kim, "A device software platform for consumer electronics based on the Internet of Things," IEEE Trans. Consumer Electron., vol. 61, no. 4, pp. 564-571, Nov. 2015

[3] D. Jo and G. J. Kim, "Local context based recognition + Internet of Things: Complementary infrastructures for future generic mixed reality space," in Proc. 21st ACM Symposium on Virtual Reality Software and Technology, pp. 196, 2015

[4] Jo, D., & Kim, G. J. (2016). ARIoT: scalable augmented reality framework for interacting with Internet of Things appliances everywhere. IEEE Transactions on Consumer Electronics, 62(3), 334–340.

[5] K. R. Patel, R. Patel, P. Patel, and K. Thakkar, "A Review on Impact of Augmented Reality and Internet of Things on Smart Cities," 2019 International Conference on Computer Communication and Informatics (ICCCI), Rajkot, India, 2019, pp. 1-6.

[6] M. Lee, J. Lee, and S. Lee, "Augmented Reality-Based User Interface for IoT Appliances in Smart Home," in IEEE Access, vol. 7, pp. 81928-81938, 2019.

[7] Smart Home Automation using Augmented Reality and Internet of Things. R. Jaivignesh2, R. D. Janarthanan1 and V. Gnanalakshmi1 Published under license by IOP Publishing Ltd.

[8] Deyuan Zhang, Zhiqiang Cao, et al. "A Comprehensive Survey of Augmented Reality"
ACM Computing Surveys, 2020.

[9] Sherali Zeadally, Zubair Baig, et al. Security and Privacy in Internet of Things (IoT): Models, Applications, and Future Directions IEEE Communications Surveys & Tutorials, 2021

[10] T. Hasan, C. S. Hong, et al. Latency in the Internet of Things: Measurement and Mitigation
IEEE Communications Surveys & Tutorials, 2021

[11] Internet of Things and Cyber-Physical Systems. Volume 3, 2023, Pages 1-13 .Apostolos Gerodimos a, Leandros Maglaras b, Mohamed Amine Ferrag c, Nick Ayres d, Ioanna Kantzavelou e

[12] Communication Protocol Stack for Constrained IoT Systems Cheena Sharma;Naveen Kumar Gondhi
2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)

[13] Monica Aiswarya Ankireddy, Rajath AV, Ruthwik Ganesh M "Augmented Reality Rendered for IoT applications", Doi: 978-1-7281-2327-1/19/$31.00 ©2019 IEEE

[14] C. G. Coogan and B. He, "Brain-Computer Interface Control in a Virtual Reality Environment and Applications for the Internet of Things," in IEEE Access, vol. 6, pp. 10840-10849, 2018.

[15] M.Vasquez-Carbonell, "A Systematic Literature Review of Augmented Reality in Engineering Education: Hardware, Software, Student Motivation & Development Recommendations.," Digital Education Review - Number 41, June 2022.

[16] Dinkar R Patnaik Patnaikuni, A Comparative Study of Arduino, Raspberry Pi and ESP8266 as IoT Development Board, International Journal of Advanced Research in Computer Science, Volume 8, No. 5, pp. 2350-2352, May-June 2017

[17] Prema Srinivasan. Author. (2021, May 27). Vuforia Engine [Online]. Available: https://www.ptc.com/en/products/augmented-reality/vuforia-engine

[18] Rekimoto, J.; Ayatsuka, Y. CyberCode: Designing augmented reality environments with visual tags. In Proc. of DARE, Elsinore, Denmark, 2000. pp. 1–10.

[19] Jo, D.; Kim, G.J. IoT+AR: Pervasive and augmented environments for Digi-log shopping experience. Human-centric Comput. Inf. Sci. 2019, 9(1), Article No.1.

[20] Huang, B.-R.; Lin, C. H.; Lee, C.-H. Mobile augmented reality based on cloud computing. International Conference on Anti-Counterfeiting, Security and Identification (ASID), Taipei, Taiwan, 2012.

[21] N.D. Author. Unity Technologies [Online]. Available:_ https://unity.com

[22] Hunt, V., & Lakhani, K. R. (2020). Designing Better IoT Devices. Harvard Business Review Available:https://hbr.org/2020/11/designing-better-iot-devices

[23] Geeks Valley, Arduino UNO r3, Available: https://geeksvalley.com/en/product/arduino-uno-r3-atmega328/

[24] Probots, ESP 32S WIFI + BLE BLUETOOTH 4.0 IOT DEVELOPMENT NODEMCU BOARD 38 PIN(WROOM-32S),Available:https://probots.co.in/esp32-wifi-ble-bluetooth-4-0-iot-development-nodemcu-board-38-pin.html

[25] Circuits Electronics, Relay Module 4 Channels - 5V),Available: https://circuits-elec.com/products/relay-module-4-channels-5v

[26] Exploring Arduino, Available: https://www.exploringarduino.com/parts/jumper-wires/

[27] Draw.io Software Available: https://app.diagrams.net/

[28] Curtain Control Systems Development on Mesh Wireless Network of the Smart Home Trio Adiono, Sinantya Feranti Anindya, Syifaul Fuada, Maulana Yusuf Fathany University Center of Excellence on Microelectronics, Institut Teknologi Bandung, IC Design laboratory, PAU Building 4th floor, ITB Campus, Jln. Tamansari No.126, Bandung (40132), Indonesia

[29] https://learn.adafruit.com/adafruit-arduino-lesson-16-stepper-motors

[30] Szeliski, R. (2022). Computer Vision: Algorithms and Applications (2nd ed.). Springer.

[31] Christiano Gava, Gabriele Bleser. 2D projective transformations (homographies). Available: https://web.archive.org/web/20171226115739/https://ags.cs.uni-kl.de/fileadmin/inf_ags/3dcv-ws11-12/3DCV_WS11-12_lec04.pdf

[32] Lee, Lik-Hang & Braud, Tristan & Hosio, Simo & Hui, Pan. (2020). Towards Augmented Reality Driven Human-City Interaction: Current Research on Mobile Headsets and Future Challenges. ACM Computing Surveys. 54. 10.1145/3467963.

[33] Ngo, Trung-Thanh & Abdukhakimov, Asatilla & Kim, Dong-Seong. (2019). Long-Range Wireless Tethering Selfie Camera System Using Wireless Sensor Networks. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2933402.

[34] Vallino, James. (2023). Interactive Augmented Reality.

[35] Neshov, N & Manolova, Agata. (2021). Objects distance measurement in augmented reality for providing better user experience. IOP Conference Series: Materials Science and Engineering. 1032. 012020. 10.1088/1757-899X/1032/1/012020.

[36] Lillemor Blom. (2018). Impact of light on augmented reality. Linköping University Department of Computer Science, 16 ECTS Datateknik, LIU-IDA/LITH-EX-G--18/072--SE

[37] Andrew Makarov. 12 Augmented Reality Trends of 2023: New Milestones in Immersive Technology. Available: https://mobidev.biz/blog/augmented-reality-trends-future-ar-technologies

[39] "IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems." International Electrotechnical Commission, 2010.

[40] "IEC 61508 series - Functional safety of electrical/electronic/programmable electronic safety-related systems."IECWebstore,Available:https://webstore.iec.ch/iec/technical-committees-and-subcommittees/browse-by-iec-committee/iec-tc-56/

[41] "IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems - Part1:General requirements."IEC Webstore, Available: https://webstore.iec.ch/publication/2ddf0e7d-35b2-4863-9d38-716c3e1a6a46

[42] "Usability Engineering" by Jakob Nielsen (1993). This book provides guidelines and principles for designing user-friendly interfaces and improving user experience.

[43] "Internet of Things: Principles and Paradigms" by Rajkumar Buyya, Amir Vahid Dastjerdi, and Srirama Satish Kumar (2016). This book covers IoT device compatibility and communication protocols.

[44] "Architecting the Internet of Things" by Dieter Uckelmann, Mark Harrison, and Florian Michahelles (2011). This book discusses connectivity and reliability aspects of IoT systems.

[45] "Internet of Things Security: Fundamentals, Techniques, and Applications" by Shancang Li, Li Da Xu, and Liming Zhu (2017). This book provides insights into securing IoT systems, including AR and IoT integration.

[46] "Scalable Internet of Things: Architecture, Protocols, and Algorithms" by Naveen Chilamkurti (2017). This book covers scalability principles and techniques for IoT systems.

[47] "Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry" by Maciej Kranz (2016).

[48] "Pragmatic Project Automation: How to Build, Deploy, and Monitor Java Applications" by Mike Clark (2004).

[49] "Internet of Things: Architectures, Protocols, and Standards" by Pethuru Raj, Anupama C. Raman, and Anbu S. Dorairaj (2020).

[50] Arduino datasheets: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf

[51] ESP32 datasheets: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[52] Forti V, Balde CP, Kuehr R, Bel G. The Global E-waste Monitor 2020: quantities, flows and the circular economy potential. Bonn, Geneva, Vienna: United Nations University, International Telecommunication Union, International Solid Waste Association [Online]. Availabile: https://ewastemonitor.info/

[53] P.Y. Dibal, E.N. Onwuka, S. Zubair, E.I. Nwankwo, S.A. Okoh, B. A Salihu, H.B. Mustaphab, Processor power and energy consumption estimation techniques in IoT applications: A review, Internet of Things, Volume 21, 2023, 100655, ISSN 2542-6605, https://doi.org/10.1016/j.iot.2022.100655.

[54] Rosca, M.I., Nicolae, C., Sanda, E. and Madan, A., 2021.Internet of Things (IoT) and Sustainability. In: R.Pamfilie, V. Dinu, L. Tăchiciu, D. Pleşea, C. Vasiliu eds. 2021. 7th BASIQ International Conference on New Trends in Sustainable Business and Consumption. Foggia, Italy, 3-5 June 2021. Bucharest: ASE, pp. 346-352 DOI: 10.24818/BASIQ/2021/07/044

[55] Memić, Belma & Haskovic Dzubur, Adisa & Avdagić-Golub, Elma. (2022). Green IoT: Sustainability Environment and Technologies. Science, Engineering and Technology. 2. 24-29. 10.54327/set2022/v2.i1.25.

[56] F. Bellotti, A. De Gloria, and A. Primavera. (2019). Augmented Reality and Internet of Things: Two Promising Technologies for Industry 4.0. In International Journal of Online Engineering, Vol. 15, No. 7, pp. 165–173.

[57] W. Jiang, D. Y. Huang, and R. J. Chen. (2018). A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. In IEEE Internet of Things Journal, Vol. 4, No. 5, pp. 1125–1142.

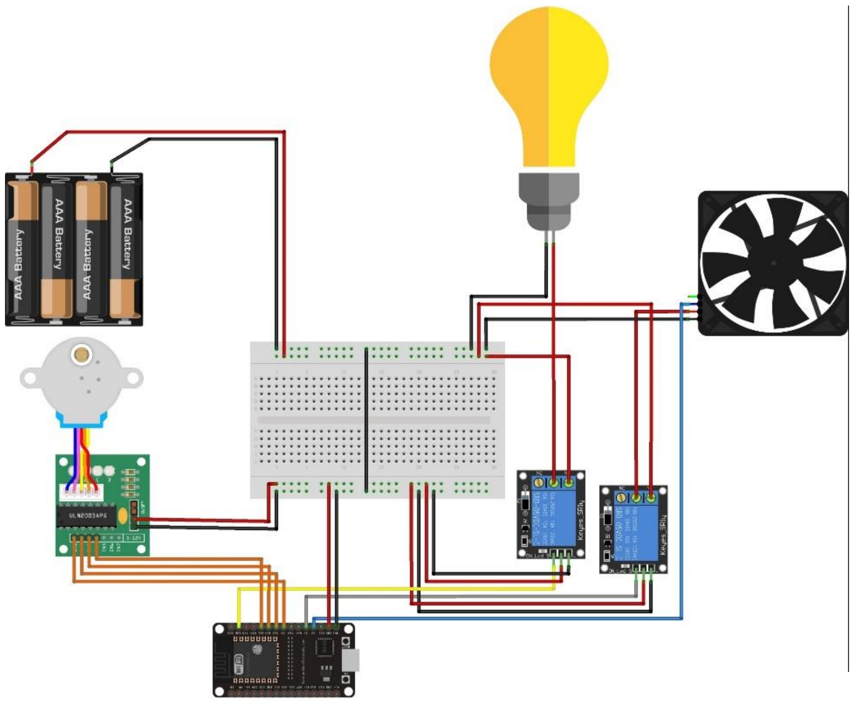
[58] M. Billinghurst and H. Kato. (2002). Collaborative Mixed Reality. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, pp. 261–262.

[59] D. Guinard and V. Trifa. (2016). Building the Web of Things: With Examples in Node.js and Raspberry Pi. Manning Publications.

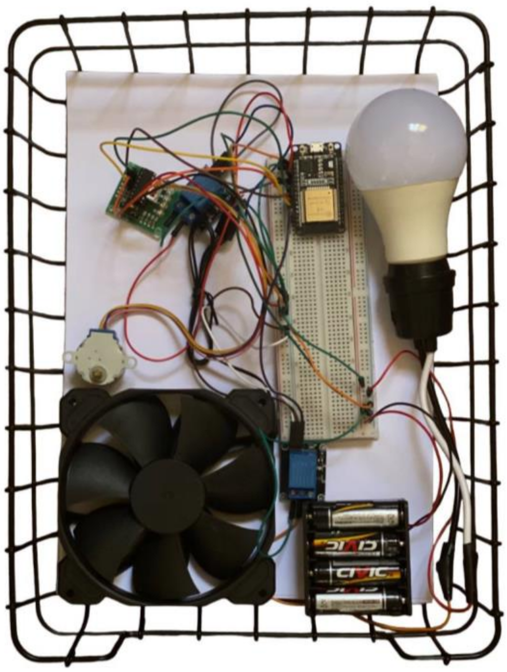
[60] S. S. S. P. Rao, S. R. Valluri, and S. E. Yilmaz. (2018). Internet of Things (IoT) and Its Impact on Energy Conservation. In Sustainable Cities and Communities Design Handbook, pp.

Appendices

Circuit diagrams:



Prototype circuit:



Final hardware:

Hardware box:

ESP32 code:

```
#define BLYNK_TEMPLATE_ID           "TMPL61h0AooTF"
#define BLYNK_TEMPLATE_NAME         "Quickstart Template"
#define BLYNK_AUTH_TOKEN            "LxTJ-5uaHH_Qyp5X94a0SgmNyKeplKXF"

#include <AccelStepper.h>
#include <BlynkSimpleEsp32.h>

const char auth[] = BLYNK_AUTH_TOKEN;         // Blynk authentication token
const char ssid[] = "**************";         // Your WiFi SSID
const char pass[] = "**************";         // Your WiFi password

const int speedPin = 2;    // Variable that control the speed of fan
const int LEDPin = 22;     // Variable that control the LED
const int FANPowerPin = 4; // Variable that switch pn/off fan


int currentPosition = 0;   // Variable that keep track the curnt position of
the motor
#define UPPER_LIMIT 13500  // Maximum upper limit
#define LOWER_LIMIT 0      // Maximum lower limit

// Flags indicator
bool clockwiseFlag = false;
bool counterclockwiseFlag = false;

// ULN2003 Motor Driver Pins
#define IN1 5
#define IN2 18
#define IN3 19
#define IN4 21

// Blynk virtual pins for the buttons
#define buttonClockwisePin V1         // Down
#define buttonCounterclockwisePin V0  // Up
#define buttonStepperOFFPin V3  // Up
#define LED V4                        // Led
#define FAN V2                        // FAN

// initialize the stepper library
AccelStepper stepper(AccelStepper::HALF4WIRE, IN1, IN3, IN2, IN4);

void setup() {
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  pinMode(LEDPin, OUTPUT);
  pinMode(speedPin, OUTPUT);
```

```
  pinMode(FANPowerPin, OUTPUT);
  // set the speed and acceleration
  stepper.setMaxSpeed(2000);
  stepper.setAcceleration(1000);
}

void loop() {
  Blynk.run();
  // Move the stepper motor (one step at a time)
  stepper.runSpeed();
    if (clockwiseFlag) {
    if (currentPosition < UPPER_LIMIT) {
      stepper.setSpeed(-1000);    // Set speed for clockwise rotation
      stepper.enableOutputs();  // Enable motor outputs
      stepper.runSpeed();       // Move the motor
      delay(2);
      currentPosition++;
    } else {
      stepper.setSpeed(0);        // Set speed to 0 to stop the motor
      stepper.disableOutputs();  // Disable motor outputs
    }
  } if (counterclockwiseFlag) {
    if (currentPosition > LOWER_LIMIT) {
      stepper.setSpeed(1000);  // Set speed for counterclockwise rotation
      stepper.enableOutputs();  // Enable motor outputs
      stepper.runSpeed();       // Move the motor
      delay(2);
      currentPosition--;
    } else {
      stepper.setSpeed(0);        // Set speed to 0 to stop the motor
      stepper.disableOutputs();  // Disable motor outputs
    }
  } if(counterclockwiseFlag == false && clockwiseFlag == false) {
      stepper.setSpeed(0);        // Set speed to 0 to stop the motor
      stepper.disableOutputs();  // Disable motor outputs
    }
}

BLYNK_WRITE(buttonClockwisePin) {
  // Set incoming value from pin V4 to a variable
  int buttonClockwiseState = param.asInt();
  if (buttonClockwiseState == HIGH) {
    clockwiseFlag = true;
    counterclockwiseFlag = false;
  } else {
    clockwiseFlag = false;
  }
}
```

```
BLYNK_WRITE(buttonCounterclockwisePin) {
  // Set incoming value from pin V1 to a variable
  int buttonCounterclockwiseState = param.asInt();
  if (buttonCounterclockwiseState == HIGH) {
    counterclockwiseFlag = true;
    clockwiseFlag = false;
  } else {
    counterclockwiseFlag = false;
  }
}

BLYNK_WRITE(buttonStepperOFFPin) {
  // Set incoming value from pin V1 to a variable
  int buttonStepperOFFState = param.asInt();
  if (buttonStepperOFFState == HIGH) {
    clockwiseFlag = false;
    counterclockwiseFlag = false;
    stepper.setSpeed(0);        // Set speed to 0 to stop the motor
    stepper.disableOutputs();  // Disable motor outputs
  }
}
BLYNK_WRITE(LED) {
  // Set incoming value from pin V4 to a variable
  int value = param.asInt();
  digitalWrite(LEDPin, value);
}

BLYNK_WRITE(FAN){
  // Set incoming value from pin V2 to a variable
int value = param.asInt();
 if(value <= 0){
  digitalWrite(FANPowerPin, LOW);
 }
 else{
  digitalWrite(FANPowerPin, HIGH);
  // Set the fan speed using PWM
  analogWrite(speedPin, value);
 }
}
```

Fan virtual buttons code:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;
using UnityEngine.Networking;

public class Fan : MonoBehaviour
{
    public VirtualButtonBehaviour Vb_high;
    public VirtualButtonBehaviour Vb_medium;
    public VirtualButtonBehaviour Vb_low;
    public VirtualButtonBehaviour Vb_off;
    public string url_high;
    public string url_medium;
    public string url_low;
    public string url_off;

    IEnumerator GetRequest(string uri)
    {
        using (UnityWebRequest webRequest = UnityWebRequest.Get(uri))
        {
            // Request and wait for the desired page.
            yield return webRequest.SendWebRequest();

        }
    }

    void Start()
    {
        Vb_high.RegisterOnButtonPressed(HighButtonPressed_high);

        Vb_medium.RegisterOnButtonPressed(MediumButtonPressed_medium);

        Vb_low.RegisterOnButtonPressed(LowButtonPressed_low);

        Vb_off.RegisterOnButtonPressed(OnButtonPressed_off);

    }
    public void HighButtonPressed_high(VirtualButtonBehaviour Vb_high)
    {
        StartCoroutine(GetRequest(url_high));
        Debug.Log("FAN IS HIGH");
    }
    public void MediumButtonPressed_medium(VirtualButtonBehaviour Vb_medium)
    {
        StartCoroutine(GetRequest(url_medium));
        Debug.Log("FAN IS MEDIUM");
    }
    public void LowButtonPressed_low(VirtualButtonBehaviour Vb_low)
    {
        StartCoroutine(GetRequest(url_low));
        Debug.Log("FAN IS LOW");
    }

    public void OnButtonPressed_off(VirtualButtonBehaviour Vb_off)
    {
        StartCoroutine(GetRequest(url_off));
        Debug.Log("FAN IS OFF");
    }
```

```
}
```

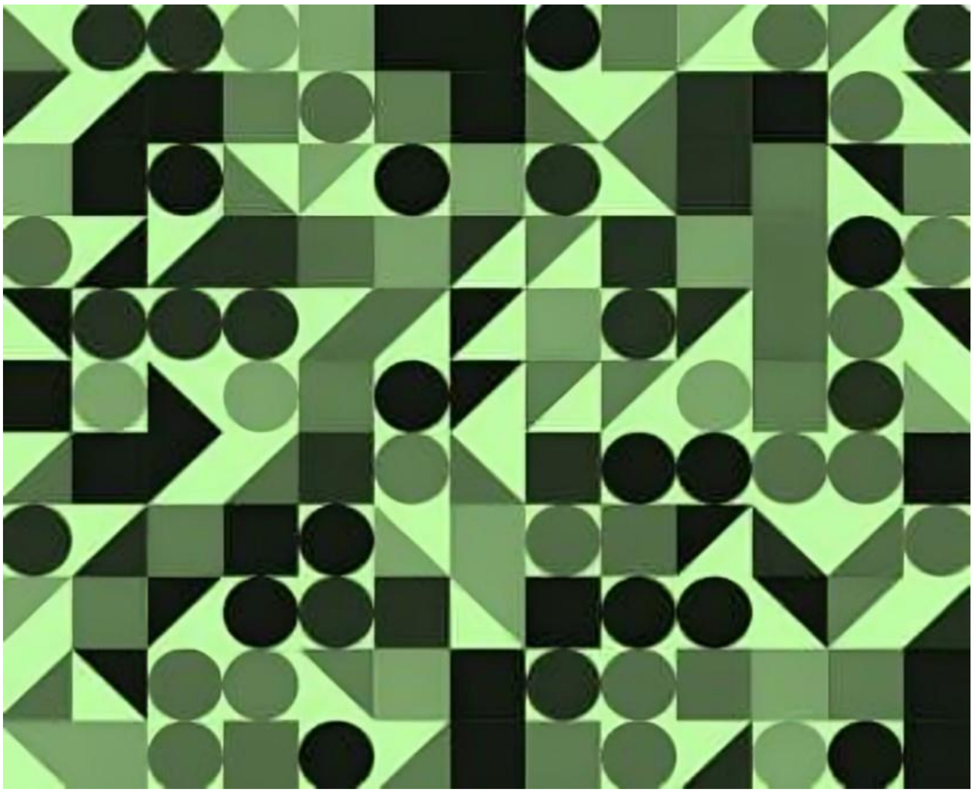
LED virtual buttons code:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;
using UnityEngine.Networking;

public class LED : MonoBehaviour
{
    public VirtualButtonBehaviour Vb_on;
    public VirtualButtonBehaviour Vb_off;
    public string url_on;
    public string url_off;

    IEnumerator GetRequest(string uri)
    {
        using (UnityWebRequest webRequest = UnityWebRequest.Get(uri))
        {
            // Request and wait for the desired page.
            yield return webRequest.SendWebRequest();

        }
    }

    void Start()
    {
        Vb_on.RegisterOnButtonPressed(OnButtonPressed_on);

        Vb_off.RegisterOnButtonPressed(OnButtonPressed_off);

    }


    public void OnButtonPressed_on(VirtualButtonBehaviour Vb_on)
    {
        StartCoroutine(GetRequest(url_on));
        Debug.Log("LED IS ON");
    }

    public void OnButtonPressed_off(VirtualButtonBehaviour Vb_off)
    {
        StartCoroutine(GetRequest(url_off));
        Debug.Log("LED IS OFF");
    }

}
```

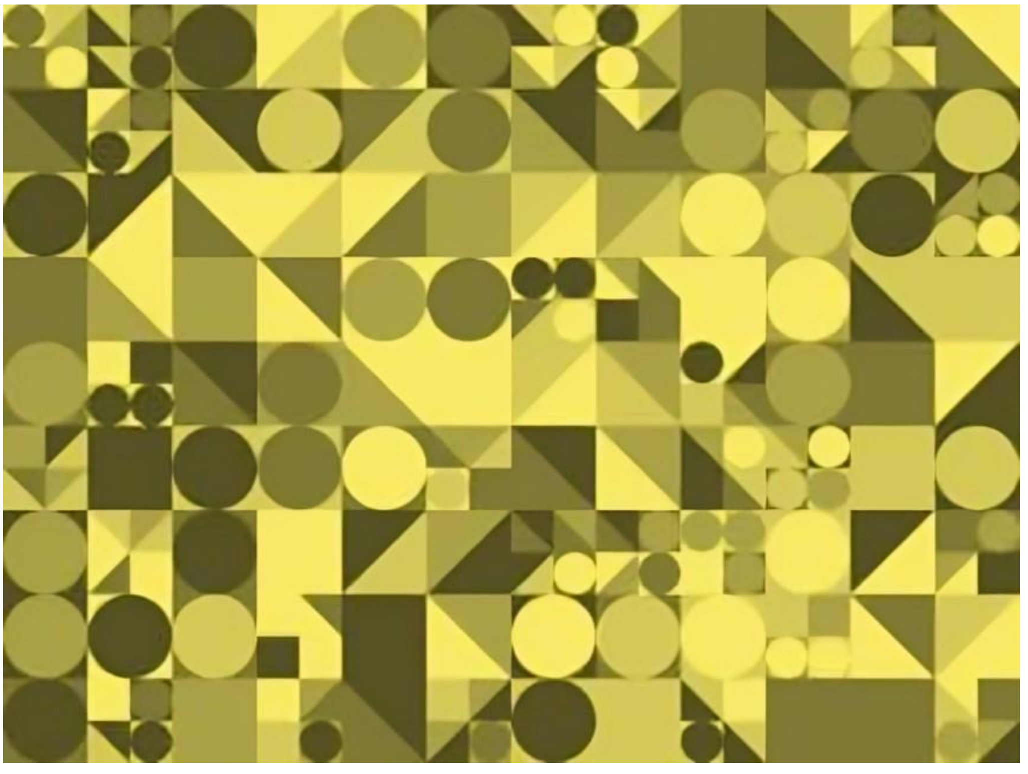
Stepper motor virtual buttons code:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;
using UnityEngine.Networking;

public class Motor : MonoBehaviour
{
    public VirtualButtonBehaviour Vb_up;
    public VirtualButtonBehaviour Vb_down;
    public VirtualButtonBehaviour Vb_off;
    public string url_up;
    public string url_down;
    public string url_off;

    IEnumerator GetRequest(string uri)
    {
        using (UnityWebRequest webRequest = UnityWebRequest.Get(uri))
        {
            // Request and wait for the desired page.
            yield return webRequest.SendWebRequest();

        }
    }

    void Start()
    {
        Vb_up.RegisterOnButtonPressed(OnButtonPressed_up);
        Vb_down.RegisterOnButtonPressed(OnButtonPressed_down);
        Vb_off.RegisterOnButtonPressed(OnButtonPressed_off);

    }


    public void OnButtonPressed_up(VirtualButtonBehaviour Vb_up)
    {
        StartCoroutine(GetRequest(url_up));
        Debug.Log("UP IS ON");
    }
    public void OnButtonPressed_down(VirtualButtonBehaviour Vb_down)
    {
        StartCoroutine(GetRequest(url_down));
        Debug.Log("DOWN IS ON");
    }

    public void OnButtonPressed_off(VirtualButtonBehaviour Vb_off)
    {
        StartCoroutine(GetRequest(url_off));
        Debug.Log("STEPPER IS OFF");
    }

}
```

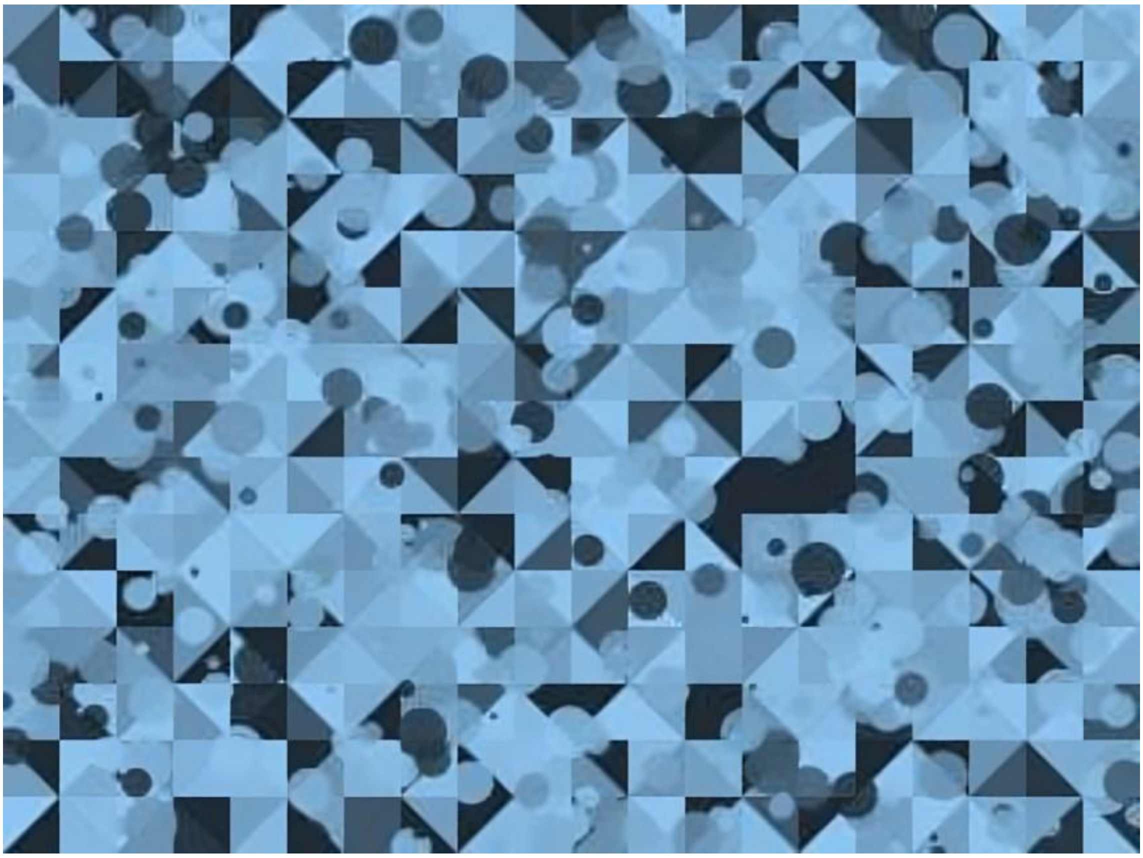
Fan target image:



LED target image:



Stepper motor target image:



Project logo:

Project poster: