

IoT switch based on Augmented Reality

*A Project report submitted in partial fulfillment
of the requirements for the degree of B. Tech in Electrical Engineering*

by

DEBAYAN CHAKRABORTY (11701619044)

RAJDEEP BASAK (11701619045)

GOURAB ROY (11700319080)

Under the supervision of

Mr. Budhaditya Biswas

Assistant Professor

Department of Electrical Engineering

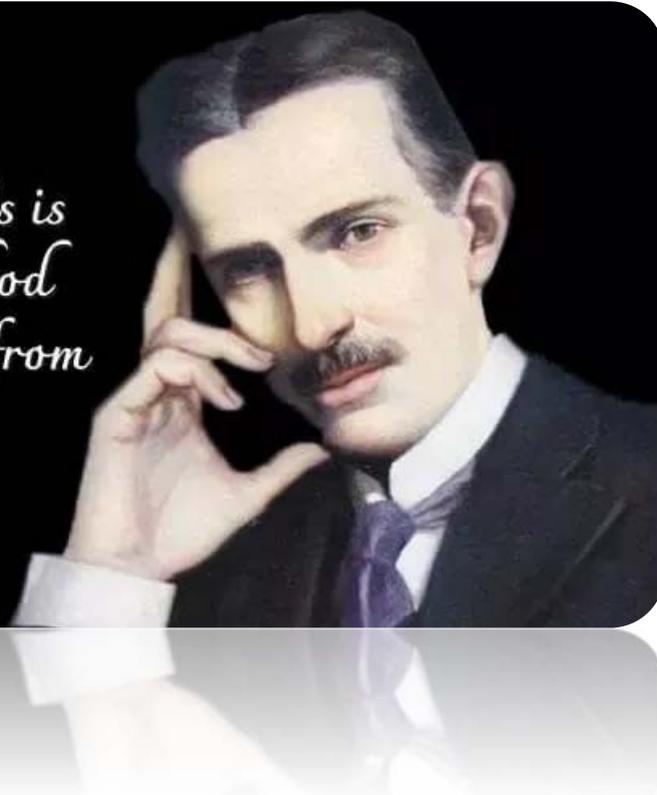


Department of Electrical Engineering
RCC INSTITUTE OF INFORMATION TECHNOLOGY
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015, WEST BENGAL
Maulana Abul Kalam Azad University of Technology (MAKAUT)

© 2023

*"The harness of waterfalls is
the most economical method
known for drawing energy from
the sun."*

- Nikola Tesla





श्रमम् विना न किमापि साध्यम्

Department of Electrical Engineering
RCC INSTITUTE OF INFORMATION TECHNOLOGY
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015, WEST BENGAL
PHONE: 033-2323-2463-154, FAX: 033-2323-4668
Email: hodeercciit@gmail.com, Website: <http://www.rcciit.org/academic/ee.aspx>

CERTIFICATE

To whom it may concern

This is to certify that the project work entitled **IoT switch based on Augmented Reality** is the bonafide work carried out by **DEBAYAN CHAKRABORTY (11701619044), RAJDEEP BASAK (11701619045), GOURAB ROY (11700319080)** the students of B.Tech in the Department of Electrical Engineering, RCC Institute of Information Technology (RCCIIT), Canal South Road, Beliaghata, Kolkata-700015, affiliated to Maulana Abul Kalam Azad University of Technology (MAKAUT), West Bengal, India, during the academic year 2022-23, in partial fulfillment of the requirements for the degree of Bachelor of Technology in Electrical Engineering and that this project has not submitted previously for the award of any other degree, diploma and fellowship.

(Budhaditya Biswas)
Assistant Professor

Department of Electrical Engineering
RCC Institute of Information Technology

Countersigned by

(Dr. Shilpi Bhattacharya)
HOD, Electrical Engineering Dept
RCC Institute of Information Technology

ACKNOWLEDGEMENT

It is our great fortune that we have got opportunity to carry out this project work under the supervision of **Mr. Budhaditya Biswas** in the Department of Electrical Engineering, RCC Institute of Information Technology (RCCIIT), Canal South Road, Beliaghata, Kolkata-700015, affiliated to Maulana Abul Kalam Azad University of Technology (**MAKAUT**), West Bengal, India. We express our sincere thanks and deepest sense of gratitude to our guide for his constant support, unparalleled guidance and limitless encouragement.

We would also like to convey our gratitude to all the faculty members and staffs of the Department of Electrical Engineering, RCCIIT for their whole hearted cooperation to make this work turn into reality.

We are very thankful to our department and to the authority of RCCIIT for providing all kinds of infrastructural facility towards the research work.

Thanks to the fellow members of our group for working as a team.

DEBAYAN CHAKRABORTY (11701619044)

RAJDEEP BASAK (11701619045)

GOURAB ROY (11700319080)

To

The Head of the Department
Department of Electrical Engineering
RCC Institute of Information Technology
Canal South Rd. Beliaghata, Kolkata-700015

Respected Sir,

In accordance with the requirements of the degree of Bachelor of Technology in the Department of Electrical Engineering, RCC Institute of Information Technology, we present the following thesis entitled “**IoT switch based on Augmented Reality**”. This work was performed under the valuable guidance of Mr. Budhaditya Biswas, Assistant Professor in the Dept. of Electrical Engineering.

We declare that the thesis submitted is our own, expected as acknowledge in the test and reference and has not been previously submitted for a degree in any other Institution.

Yours Sincerely,

DEBAYAN CHAKRABORTY
(11701619044)

Debayan Chakraborty

RAJDEEP BASAK (11701619045)

Rajdeep Basak

GOURAB ROY (11700319080)

Gourab Roy

Contents

Topic	Page No.
List of figures	i
List of tables	ii
Abbreviations and acronyms	iii
Abstract	iv
Chapter 1 (Introduction)	
Introduction	2
1.1 Home Automation	2
1.2 Overview & Benefits	3
1.3 Organization of Thesis	4
Chapter 2 (Literature Review)	5
Chapter 3 (Theory)	
3.1 Augmented Reality (AR)	11
3.1.1 How does augmented reality works	11
3.1.2 Difference between AR and VR	12
3.1.3 Top AR use cases	12
3.2 Overview of the project	13
3.3 Circuit Diagram	14
Chapter 4 (Hardware modeling)	
4.1 Main Features of the Prototype	16
4.2 Photographs of the prototype	16
4.3 Step by step operation of the prototype	16
4.4 Components Required	17
4.5 Hardware Interfacing	18

4.5.1	ESP32 OLED display with Arduino IDE	18
4.5.2	Relay Driver Interfacing with μ C	24
Chapter 5 (Logic & Operation)		
5.1	Introduction	26
5.2	Flow chart	26
5.3	Principle & operations	27
5.4	Advantages of the AG Load switching	27
5.5	Disadvantages	28
5.6	Cost estimation of the project	28
5.7	Photographs of the prototype	29
Chapter 6 (Conclusion & Future scope)		
6.1	Conclusion	33
6.2	Results	33
6.3	Future works	33
Chapter 7 (Reference)		34 – 35
Appendix A (Hardware Description)		36 – 44
Appendix B (AG App development using Unity)		45 – 72
Appendix C (Software Coding)		73 – 75
Appendix D (Datasheets)		76

List of Figures

Sl. No.	Figure	Page No.
1	Concept of home automation	2
2	Augmented Reality	11
3	Virtual Reality	12
4	Overview of the complete project	11
5	Complete circuit diagram of the project	14
6	Simulation of the project in Wokwi online simulator	14
7	Main prototype with OLED and relay driver	16
8	0.96 inch OLED display module	18
9	OLED interfacing with ESP 32	19
10	Text message display on OLD	22
11	ULN2003A interfacing with microcontroller	24
12	Main prototype and driver	29
13	Load is switch off and switch on condition	29
14	Different messages on the OLED	29
15	Making of Android app using unity and Vuforia	30
16	Target image and virtual layer	30
17	Using the AG app to control the load	31
18	Transformer less SMPS 5 V power supply	37
19	ESP32 development module	38
19	SPI and IIC OLED display module	40
20	ULN 2003a internal block diagram	40
21	Resistor	40
22	Colour code for resistance	41
23	6 Volt cube relay	42
24	Types of capacitors	43
25	Pizzo Buzzer	43
26	A blank glass epoxy PCB board	44

List of Tables

Sl. No.	Table	Page No.
1	Component Listing	17
2	OLED Pin Configuration	18
3	Cost estimation of the project	28

ABBREVIATIONS AND ACRONYMS

AG – Augmented Reality

IoT – Internet of Things

HVAC – Heating Ventilation and Air *Conditioning*

IC - Integrated Circuit

PCB – Printed Circuit Board

μ *C* – Micro Controller

BJT - Bi-polar Junction Transistor

SPDT - Single Pole Double Throw

NO - Normally Open

NC - Normally Closed

COM – Common

OLED – Organic Light Emitting Diode

LED - Light Emitting Diode

SMPS – Switch Mode Power Supply

ISM – Industrial, scientific and medical

USB – Universal serial bus

SPI – Serial Peripheral Interface

I²C – Inter-Integrated Circuit

TXD – Transmitter

RXD – Receiver

GND – Ground

ABSTRACT

Traditionally electrical appliances in a home are controlled via switches that regulate the electricity to these devices. As the world gets more and more technologically advanced, new technology coming in deeper and deeper into our personal lives even at home. Home automation is becoming more and more popular around the world and is becoming a common practice. The process of home automation works by making everything in the house automatically controlled using technology to control and do the jobs that we would normally do manually. Home automation takes care of a lot of different activities in the house.

IoT in other way could be a combination of physical objects with virtual depictions and services. The chief famous implementation of IOT involves Home Automation Systems. The most recent formation of IOT is now converging to IOE (Internet of Everything). Each home will become a smart home in future, where user is going to be having remote access to every and each appliance of his/her house. This project presents an affordable and lithe home control and milieu monitoring system. The work mainly concentrates on IOT based home automation using Augmented Reality acting between different home appliances based on wireless home automation system using IOT helps us control basic home appliances automatically by using smartphones. By using, this concept can be suitably combined to make it smarter, safer, and automated. The microcontroller used in the current project is the ESP32 for accessing and controlling devices and appliances remotely. The power obtained by selecting this system over similar existing systems is that the alerts and the status sent by the Wi-Fi-connected microcontroller managed system can be received by the user on his phone from any distance regardless of whether his mobile phone is connected to the internet. Lastly, Augmented Reality implementation in this project brings users executing direct interactions to manipulate all objects in the surrounding world. Thus, this project presents an idea or a concept for home automation using ESP32 with Blynk Cloud, Augmented Reality, and a switch to control the Relays with the help of internet.

The load status is showing in a 0.96" OLED inbuilt in the circuit. The whole system is power up through a 5V 1A adapter.

CHAPTER 1

(Introduction)

INTRODUCTION

The aim of the proposed system is to develop a cost-effective solution that will provide controlling of home appliances remotely and enable home security against intrusion in the absence of homeowner. The system provides availability due to development of a low-cost system. The home appliances control system with an affordable cost was thought to be built that should be mobile providing remote access to the appliances and allowing home security. Though devices connected as home and office appliances consume electrical power. These devices should be controlled as well as turn on/off if required. Most of the times it was done manually. Now it is a necessity to control devices more effectively and efficiently at anytime from anywhere.

In this system, we are going to develop a Augmented Reality controlled home/office appliance. This system is designed for controlling arbitrary devices, it includes a mobile app through which user can control the electrical load through a relay. The ESP32 microcontroller connected to the internet through the wifi. The microcontroller also connected to the blynk cloud. When a API command is initiated through the AG app, developed using Unity and Vuforia the relay is switch ON/OFF the electrical load connected. The load status is showing in a 0.96" OLED inbuilt in the circuit. The whole system is power up through a 5V 1A adapter.

1.1 HOME AUTOMATION

Home automation is the residential extension of building automation. It is automation of the home, housework or household activity. Home automation may include centralized

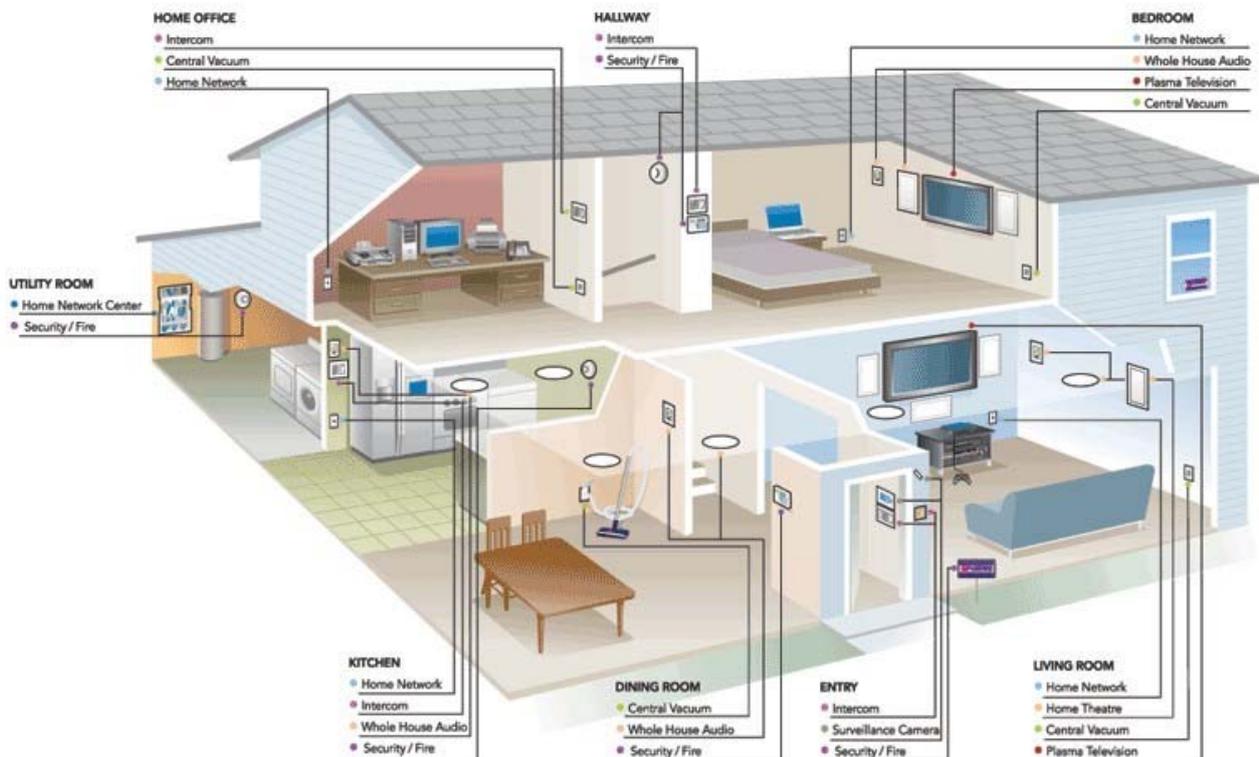


Figure 1: Concept of Home Automation

control of lighting, HVAC (heating, ventilation and air conditioning), appliances, security locks of gates and doors and other systems, to provide improved convenience, comfort, energy efficiency and security. Home automation for the elderly and disabled can provide increased quality of life for persons who might otherwise require caregivers or institutional care.

The popularity of home automation has been increasing greatly in recent years due to much higher affordability and simplicity through Smartphone and tablet connectivity. The concept of the "Internet of Things" has tied in closely with the popularization of home automation.

A home automation system integrates electrical devices in a house with each other. The techniques employed in home automation include those in building automation as well as the control of domestic activities, such as home entertainment systems, houseplant and yard watering, pet feeding, changing the ambiance "scenes" for different events (such as dinners or parties), lighting control system, and the use of domestic robots. Devices may be connected through a home network to allow control by a personal computer, and may allow remote access from the internet. Through the integration of information technologies with the home environment, systems and appliances can communicate in an integrated manner which results in convenience, energy efficiency, and safety benefits.

Automated "homes of the future" have been staple exhibits for World's Fairs and popular backgrounds in science fiction. However, problems with complexity, competition between vendors, multiple incompatible standards and the resulting expense have limited the penetration of home automation to homes of the wealthy, or ambitious hobbyists. Possibly the first "home computer" was an experimental home automation system in 1966.

1.2 Overview and benefits

Home automation refers to the use of computer and information technology to control home appliances and features (such as windows or lighting). Systems can range from simple remote control of lighting through to complex computer/microcontroller-based networks with varying degrees of intelligence and automation. Home automation is adopted for reasons of ease, security and energy efficiency.

In modern construction in industrialized nations, most homes have been wired for electrical power, telephones, TV outlets (cable or antenna), and a doorbell. Many household tasks were automated by the development of specialized automated appliances. For instance, automatic washing machines were developed to reduce the manual labor of cleaning clothes, and water heaters reduced the labor necessary for bathing.

The use of gaseous or liquid fuels, and later the use of electricity enabled increased automation in heating, reducing the labor necessary to manually refuel heaters and stoves. Development of thermostats allowed more automated control of heating, and later cooling.

As the number of controllable devices in the home rises, interconnection and communication becomes a useful and desirable feature. For example, a furnace can send an alert message when it needs cleaning or a refrigerator when it needs service. If no one is supposed to be home and the alarm system is set, the home automation system could call the owner, or the neighbors, or an emergency number if an intruder is detected.

In simple installations, automation may be as straightforward as turning on the lights when a person enters the room. In advanced installations, rooms can sense not only the presence of a person inside but know who that person is and perhaps set appropriate lighting, temperature,

music levels or television channels, taking into account the day of the week, the time of day, and other factors.

Other automated tasks may include reduced setting of the heating or air conditioning when the house is unoccupied, and restoring the normal setting when an occupant is about to return. More

sophisticated systems can maintain an inventory of products, recording their usage through bar codes, or an RFID tag, and prepare a shopping list or even automatically order replacements.

Home automation can also provide a remote interface to home appliances or the automation system itself, to provide control and monitoring on a Smartphone or web browser.

An example of remote monitoring in home automation could be triggered when a smoke detector detects a fire or smoke condition, causing all lights in the house to blink to alert any occupants of the house to the possible emergency. If the house is equipped with a home theater, a home automation system can shut down all audio and video components to avoid distractions, or make an audible announcement. The system could also call the home owner on their mobile phone to alert them, or call the fire department or alarm monitoring company.

1.3 Organisation of thesis

The thesis is organised into five chapters including the chapter of introduction. Each chapter is different from the other and is described along with the necessary theory required to comprehend it.

Chapter 2 deals with the literature reviews. From this chapter we can see before our project who else works on this topic and how our project is different and advance from those projects.

Chapter 3 deals with the theory required to do the project. The basic of augmented reality, connect ESP32 with Blynk cloud, built an AG app using unity and vuforia are described here. The overview of the project and software simulation of the project is also listed in this chapter.

Chapter 4 deals with the hardware modelling of the projects. The main features, photographs, step by step operation of the prototype, component listing and the hardware interfacing of the required components are described here.

Chapter 5 describes the basic operation of the circuit. A flow chart is presented on the actions that would take in the controller beginning from the AG app in android device with the API command to the switching on and off of loads. Advantages and disadvantages and cost estimation are listed in this chapter.

Chapter 6 concludes the work performed so far. The possible limitations in proceeding research towards this work are discussed. The future work that can be done in improving the current scenario is mentioned. The future potential along the lines of this work is also discussed.

Chapter 7 References are listed in this chapter

Appendix A, B & C Hardware description, software coding and datasheets are listed here.

CHAPTER 2

(Literature Review)

[1] Ritvik Khanna School of Information and Engineering Vellore Institute Of Technology Vellore, India, Vanitha M Associate Prof, School Of Information and Engineering Vellore Institute Of Technology Vellore, India Augmented Reality Based IOT Controller, 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)

Augmented reality enables computer generated graphics to be superimposed on real world objects. From 3D objects to animations, with the help of AR, one can create an interactive and assistance software applications. This project represents a framework for implementing an AR powered controller with interactive controlling of different devices that are connected to each other and communicate via the internet. The aim of this project to establish a more fundamental approach in integrating mobile AR with internet of things. The users are able to immerse themselves with a computer-generated reality and use it to control vast number of diverse objects and devices. The mobile application can be used with an AR headset for a better experience. This project produces three important contributions:

- A mobile AR based application to control IOT enabled devices.
- Implementation of the application using Unity 3D game engine.
- Future scope of blending AR with IOT.

[2] An Augmented Reality Prototype for supporting IoT-based Educational Activities for Energy-efficient School Buildings, Georgios Mylonas, Christos Triantafyllis, Dimitrios Amaxilatis, Computer Technology Institute and Press “Diophantus” Research Unit 1 Patras, 26504, Greece

The use of Augmented Reality (AR) technology is currently being investigated in numerous and diverse application domains. In this work, they discuss the ways in which they are integrating AR into educational in-class activities for the GAIA project, aiming to enhance existing tools that target behavioral changes towards energy efficiency in schools. They combine real-time IoT data from a sensing infrastructure inside a fleet of school buildings with AR software running on tablets and smartphones, as companions to a set of educational lab activities aimed at promoting energy awareness in a STEM context. They also utilize this software as a means to ease access to IoT data and simplify device maintenance. They report on the design and current status of our implementation, describing functionality in the context of our target applications, while also relaying our experiences from the use of such technologies in this application domain

[3] AR Enabled IoT for a Smart and Interactive Environment: A Survey and Future Directions 1 Department of Digital Contents Engineering, Wonkwang University 2 Department of Computer Science and Engineering, Korea University

Accompanying the advent of wireless networking and the Internet of Things (IoT), traditional augmented reality (AR) systems to visualize virtual 3D models of the real world are evolving into smart and interactive AR related to the context of things for physical objects. They propose the integration of AR and IoT in a complementary way, making AR scalable to cover objects everywhere with an acceptable level of performance and interacting with IoT in a more intuitive manner. They identify three key components for realizing such a synergistic integration:

- Distributed and object-centric data management (including for AR services).
- IoT object-guided tracking.

- Seamless interaction and content interoperability. They survey the current state of these respective areas and herein discuss research on issues about realizing a future smart and interactive living environment.

[4] Augmented and Virtual Reality Based Monitoring and Safety System: A Prototype IoT Platform Md Fasiul Alam, Serafeim Katsikas, Olga Beltramello , Stathes Hadjiefthymiades

This paper presents an Augmented and Virtual Reality (AR/VR) based IoT prototype system. Performing maintenance tasks in a complex environment is quite challenging and difficult due to complex, and possibly, underground facilities, uneasy access, human factors, heavy machineries, etc. Current technology is not acceptable because of significant delays in communication and data transmission, missing multi-input interfaces, and simultaneous supervision of multiple workers who are working in the extreme environment. The aim is to technically advance and combine several technologies and integrate them as integral part of a personnel safety system to improve safety, maintain availability, reduce errors and decrease the time needed for scheduled or ad hoc interventions. They emphasize on the aspects that were made “feasible” on the worker’s side due to the equipment used (mobile computing equipment). They present that the demanding tasks that previously were simply undertaken on the fixed infrastructure are now possible on the mobile end. The research challenges lie in the development of real-time data transmission, instantaneous analysis of data coming from different inputs, local intelligence in low power embedded systems, interaction with multiple on-site users, complex user interfaces, portability and wearability. This work is part EDUSAFE, a Marie Curie ITN (Initial Training Network) project focusing on research into the use of Augmented and Virtual Reality (AR/VR) during planned and adhoc maintenance in extreme work environments.

[5] Augmented Reality based Mobile Application for Energy Monitoring and IoT Device Control Abishek R, D. Vaishali, Adhitya Narayan R, Vignesh Sundar M International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075 (Online), Volume-11 Issue-1, November, 2021

IoT has become an integrated part of our lives changing ways in which they operate our everyday appliances. In addition to making our home appliances smart, it has become a common trend for companies to adopt industry 4.0, which uses various sensors to monitor the equipment, machinery, and the work environment. They often come across multiple brands which make smart appliances but each brand comes with its separate mobile application for the appliance's operation. This requires us to switch between Apps to control these appliances if they at all remember which App controls which appliance. They intend to solve these two major inconveniences by creating a single mobile application that can control all these appliances using Augmented Reality technology. All they have to do is point our camera at the appliance that they need to operate and the App will display control options in real-time AR.

[6] Augmented Reality Enabled IoT Services for Environmental Monitoring Utilising Serious Gaming Concept Boris Pokric, Srdjan Kr, Dejan Draji, Maja Pokri , Vladimir Rajs, Živorad Mihajlovic, Petar Kneževi and Dejan Jovanovi Dunav, 21000 Novi Sad, Serbia

This paper describes an approach to integration of physical and digital worlds through aggregation of Internet of Things (IoT) service with the Augmented Reality (AR) platform, AR Genie. The

solution is generally applicable, but concretely in this paper the IoT service is provided by ekoNET platform which provides the environmental data through AR based applications. The ekoNET solution is developed for a real-time monitoring of air quality and other atmospheric condition parameters such as temperature, air pressure and humidity. AR Genie is **Software as a Service (SaaS) and Platform as a Service (PaaS)** solution offering features and functionalities for creation and deployment of AR applications using simple and intuitive drag and drop interface requiring no programming knowledge. By extending the **AR Genie platform with the ekoNET IoT** service they were able to demonstrate usage of a real-time environmental data within AR mobile applications as well as to enable a new, more engaging way of IoT data visualization utilizing serious gaming and AR technologies.

- [7] **Augmented Reality with Internet of Things Suresh kumar S Assistant Professor Dept of computer Science Sri Krishna College of Engineering and Technology Coimbatore, India Agash C P PG Student Dept of computer Science Sri Krishna College of Engineering and Technology Coimbatore, India Ramya S Assistant Professor Department of Computer Science and Engineering Paavai Engineering College, Namakkal. Kaviyaraj R Research Scholar Department of Computer Science and Engineering SRM Institute of Science and Technology Chennai, India. Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021) IEEE Xplore Part Number: CFP21OAB-ART; ISBN: 978-1-7281-9537-7**

In Today's World technological changes make the probability of more complex things made into simple tasks with more accuracy in major areas and mostly in Manufacturing Industry. Internet of things contributes its major part in automation which helps human to make life easy by monitoring and directed to a related person with in a fraction of second. Continuous advances and improvement in computer vision, mobile computing and tablet screens have led to a revived interest in Augmented Reality the Augmented Reality makes the complex automation into an easier task by making more realistic real time animation in monitoring and automation on **Internet of Things (eg like temperature, time, object information, installation manual, real time testing)**. Inorder to identify and link the augmented content, like object control of home appliances, industrial appliances. The AR-IoT will have a much cozier atmosphere and enhance the overall Interactivity of the IoT environment. Augmented Reality applications use a myriad of data generated by IoT devices and components, AR helps workers become more competitive and productive with the realistic environment in IoT. Augmented Reality and Internet of Things together plays a critical role in the development of next generation technologies. This paper describes the concept of how Augmented Reality can be integrated with industry (AR-IoT)4.0 and how the sensors are used to monitoring objects/things contiguously round the clock, and make the process of converting real-time physical objects into smart things for the upcoming new era with AR-IoT.

- [8] **Interaction Paradigm of IoT using Augmented Reality Siddharth Bhokare International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 p-ISSN: 2395-0072**

IoT in other way could be a combination of physical objects with virtual representations and services. The foremost famous implementation of IOT involves Home Automation Systems. The most recent formation of IOT is now converging to **IOE (Internet of Everything)**. While using this technology, one can access each and every device, equipment and machinery within its surrounding

through any application or “middleware system”. Each home will become a smart home in future, where user is going to be having remote access to every and each appliance of his/her house. For this technological enhancement The Augmented Reality (AR) is among top trending technology of today i.e creating visual solutions of digital problems and experiencing and visualizing them within the world. Augmented reality integrates the unique abilities of kinsfolk that can't be replicated by AI, and Bonafede concerns about AI are raised by some visionaries

[9] IoT based Home Automation System using Augmented Reality Aaditya Auti¹, Bharat Kolekar², Saylee Surve³, Tejas Lad⁴, Prof. Shrikrishna Patil⁵ 1,2,3,4Student, Rajiv Gandhi Institute of Technology, Mumbai 5 Faculty, Rajiv Gandhi Institute of Technology, Mumbai ISSN: 2349-6002

This project presents a low-cost and flexible home control and environment monitoring system. The work mainly concentrates on IOT based home automation using Augmented Reality acting as GUI between different home appliances based on wireless home automation system using IOT helps us control basic home appliances automatically by using smartphones. When it comes to our house, this concept can be aptly incorporated to make it smarter, safer, and automated. The leverage obtained by preferring this system over similar existing systems is that the alerts and the status sent by the Wi-Fi-connected microcontroller managed system can be received by the user on his phone from any distance irrespective of whether his mobile phone is connected to the internet. The microcontroller used in the current project is the ESP32 for accessing and controlling devices and appliances remotely. Lastly, Augmented Reality implementation in this project brings users executing direct interactions to manipulate all objects in the surrounding world. Thus, this project presents an idea or a concept for home automation using **ESP32 with Blynk App, Augmented Reality**, and a switch to control four Relays with and without internet.

CHAPTER 3

(Theory)

3.1 Augmented Reality (AR)

Augmented reality (AR) is the integration of digital information with the user's environment in real time. Unlike virtual reality (VR), which creates a totally artificial environment, AR users experience a real-world environment with generated perceptual information overlaid on top of it.

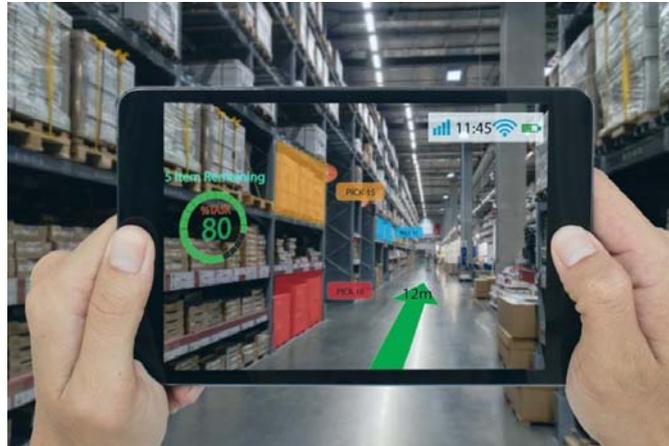


Figure 2: Augmented Reality

Augmented reality is used to either visually change natural environments in some way or to provide additional information to users. The primary benefit of AR is that it manages to blend digital and three-dimensional (3D) components with an individual's perception of the real world. AR has a variety of uses, from helping in decision-making to entertainment.

AR delivers visual elements, sound and other sensory information to the user through a device like a smartphone or glasses. This information is overlaid onto the device to create an interwoven experience where digital information alters the user's perception of the real world. The overlaid information can be added to an environment or mask part of the natural environment.

3.1.1 How does augmented reality work?

Augmented reality can be delivered in a variety of formats, including within smartphones, tablets and glasses. AR delivered through contact lenses is also being developed. The technology requires hardware components, such as a processor, sensors, a display and input devices. Mobile devices already typically have this hardware available, with sensors including cameras, accelerometers, Global Positioning System (GPS) and solid-state compasses. This helps make AR more accessible to the everyday user. A GPS is used to pinpoint the user's location, and its compass is used to detect device orientation, for example.

Sophisticated AR programs used by the military for training can also include machine vision, object recognition and gesture recognition. AR can be computationally intensive, so if a device lacks processing power, data processing can be offloaded to a different machine.

Augmented reality apps are written in special 3D programs that enable developers to tie animation or contextual digital information in the computer program to an *augmented reality marker* in the real world. When a computing device's AR app or browser plugin receives digital information from a known marker, it begins to execute the marker's code and layer the correct image or images.

3.1.2 Differences between AR and VR

VR is a virtual environment created with software and presented to users in such a way that their brain suspends belief long enough to accept a virtual world as a real environment. Virtual reality is primarily experienced through a headset with sight and sound.



Figure 3: Virtual Reality

The biggest difference between AR and VR is that augmented reality uses the existing real-world environment and puts virtual information on top of it, whereas VR completely immerses users in a virtually rendered environment. While VR puts the user in a new, simulated environment, AR places the user in a sort of mixed reality.

The devices used to accomplish this are different, too. VR uses VR headsets that fit over the user's head and present them with simulated visual and audio information. AR devices are less restrictive and typically include devices like phones, glasses, projections and HUDs in cars.

In VR, people are placed inside a 3D environment in which they can move around and interact with the generated environment. AR, however, keeps users grounded in the real-world environment, overlaying virtual data as a visual layer within the environment.

3.1.3 Top AR use cases

AR can be used in the following ways:

- **Retail.** Consumers can use a store's online app to see how products, such as furniture, will look in their own homes before buying.
- **Entertainment and gaming.** AR can be used to overlay a virtual game in the real world or enable users to animate their faces in different and creative ways on social media.
- **Navigation.** AR can be used to overlay a route to the user's destination over a live view of a road. AR used for navigation can also display information about local businesses in the user's immediate surroundings.
- **Tools and measurement.** Mobile devices can use AR to measure different 3D points in the user's environment.
- **Architecture.** AR can help architects visualize a building project.

- **Military.** Data can be displayed on a vehicle's windshield that indicates destination directions, distances, weather and road conditions.
- **Archaeology.** AR has aided archaeological research by helping archaeologists reconstruct sites. 3D models help museum visitors and future archaeologists experience an excavation site as if they were there.

3.2 Overview of the project

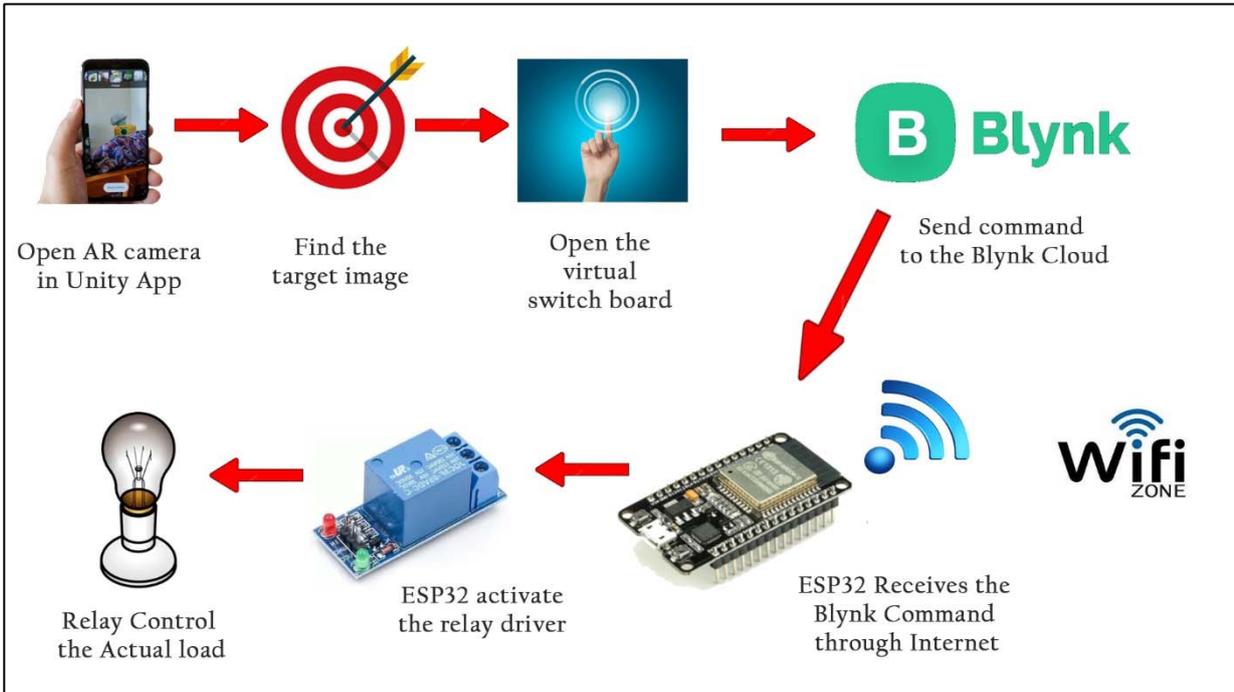


Figure 4: Overview of the complete project

According to fig. No. 4 i.e. overview of the project, first we need to place the ESP32 in the wifi zone and make sure that the microcontroller must be connected to the wifi. Open the augmented reality application in the android mobile designed by the Unity and Vuforia software. Find the target image using the AR camera within the application. Once the target image found, the app add some digital layer of a virtual switchboard on the target image. Press the ON/OFF button on the virtual switchboard. Once the switch is pressed the Blynk cloud API command will be activated and the Blynk server communicate with the ESP32 through internet. Receiving the signal from the Blynk server ESP32 send ON/OFF command to the relay driver. The relay activates the actual load based on the command received from the controller. The real time status of the load will be display over a 0.96 inch colour OLED screen which is install on the prototype. The small buzzer also generates some audio feedback during load switching to ensure the load is turn on or off properly.

3.3 Circuit Diagram

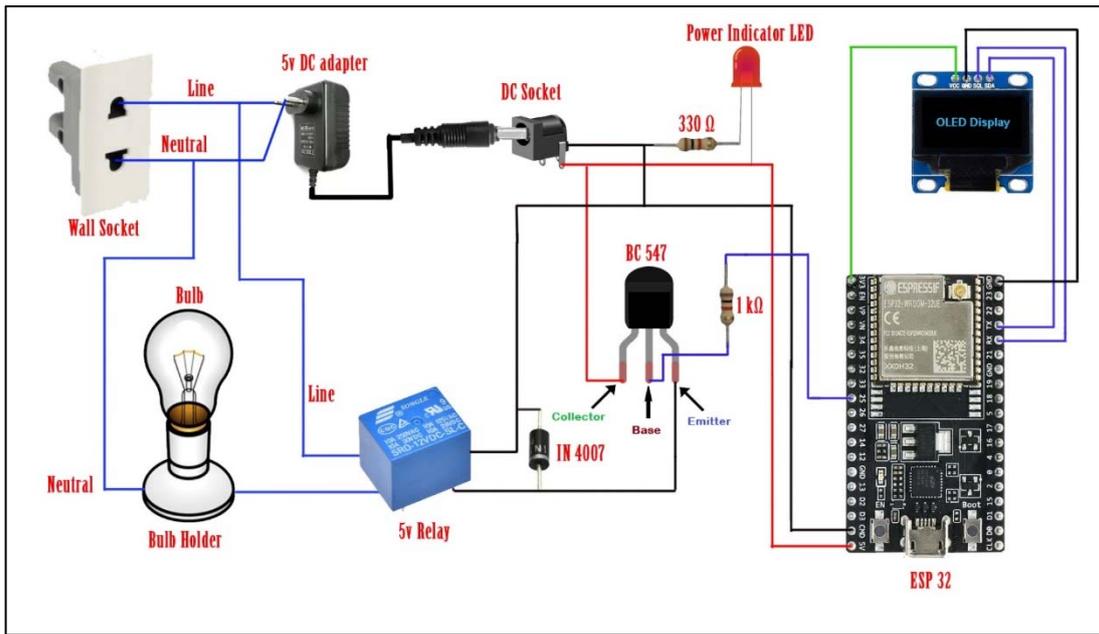


Figure 5: Complete circuit Diagram of the Project

The complete circuit is first simulated in the Wokwi simulator, the screenshot of the Wokwi simulation is shown in figure 6. The circuit is running well in this software. The software simulation includes microcontroller (ESP32), OLED and LED. Here we incorporate 1 load which can be controlled as per user choice. The load status is showing in a 0.96" OLED inbuilt in the circuit. The whole system is power up through a 5V 1A adapter.

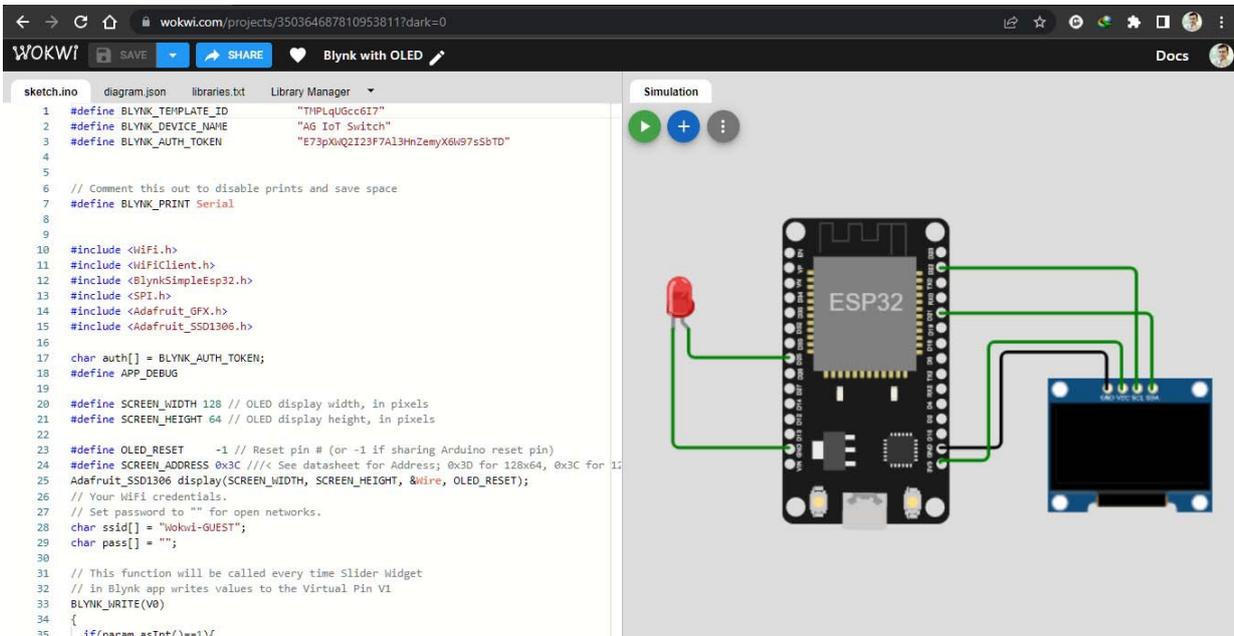


Figure 6: Simulation of the Project in Wokwi online simulator

CHAPTER 4

(Hardware Modeling)

4.1 Main features of the prototype

The features of the developed prototype are:

- Secure (AG based virtual switchboard)
- OLED display (showing the condition of the load status and the status of the circuit)
- 1 electrical load controls (250 volt, 7 amp max, ON/OFF control)
- Inbuilt relay driver
- Buzzer indication during load switching
- Power indication LEDs
- 5 Volt operation (both control board and relay board)
- Virtual switchboard can be generated through the AG app only on the target image

4.2 Photographs of the prototype

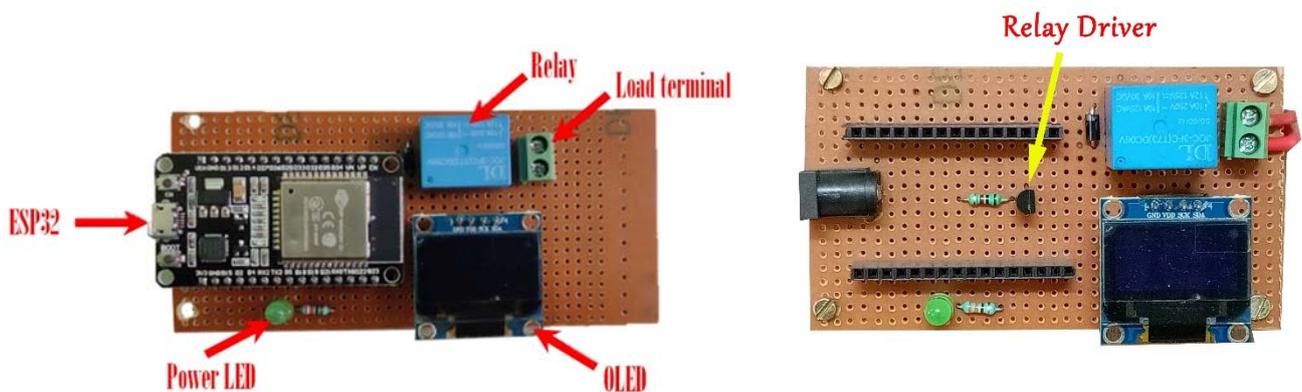


Figure 7: Main prototype with OLED and relay driver

4.3 Step by step operation of the prototype

1. Connect the DC adapter (5V, 1A) to the DC jack.
2. Power On the circuit
3. Connect the wifi and make sure that the device is connected to the wifi.
4. Open the AG app in the android mobile to scan the target image



5. On the detection of the target image the app will create a virtual digital layer on the target image.
6. Switch on the load by pressing the ON switch on the virtual switchboard.
7. Switch off the load by pressing the OFF switch on the virtual switchboard.
8. The status of the load will be seen in the OLED screen.



4.4 Components required

Sl. No.	Components	Quantity
1	ESP32 Microcontroller	1
2	330 Ω resistance	1
3	OLED Display	1
4	5 volt static Relay	1
5	Female pin header	1
6	Piezo Buzzer	1
7	Single stand wire	3m
8	Wire nipper	1
9	Wire striper	1
10	Soldering Iron	1
11	Soldering material	1
12	De-soldering pump	1
13	BC 547	1
14	DC Socket	1
15	IN 4007	1
16	Bulb Holder	1
17	Bulb	1

Table 1: Component listing

4.5 Hardware interfacing

4.5.1 ESP32 OLED Display with Arduino IDE

The combination of OLED with ESP32 is so popular that there are some boards of ESP32 with the OLED integrated. We'll, however, assume that you will be using a separate OLED module with your ESP32 board. If you have an OLED module, it perhaps looks like the image below.

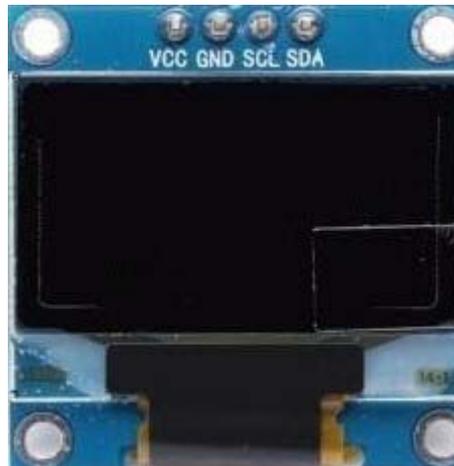


Figure 8: 0.96 inch OLED display module

The OLED (Organic Light Emitting Diode) display that we'll use in this tutorial is the SSD1306 model: a mono color, 0.96-inch display with 128×64 pixels as shown in the above figure.

The OLED display doesn't require backlight, which results in a very nice contrast in dark environments. Additionally, its pixels consume energy only when they are on, so the OLED display consumes less power when compared to other displays.

The model we're using has four pins and communicates with any microcontroller using I2C communication protocol. There are models that come with an extra RESET pin or that communicate using SPI communication protocol.

OLED Display SSD1306 Pin Wiring

Because the OLED display uses I2C communication protocol, wiring is very simple. Use the following table as a reference.

Pin	ESP32
Vin	3.3V
GND	GND
SCL	GPIO 22
SDA	GPIO 21

Table 2: OLED pin configuration

Alternatively, it can follow the next schematic diagram to wire the ESP32 to the OLED display.

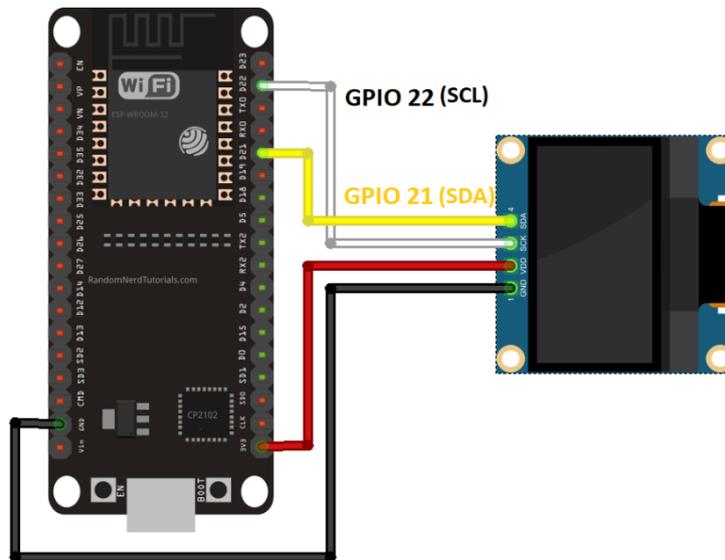


Figure 9: OLED interfacing with ESP32

In this example, we're using I2C communication protocol. The most suitable pins for I2C communication in the ESP32 are GPIO 22 (SCL) and GPIO 21 (SDA).

If you're using an OLED display with SPI communication protocol, use the following GPIOs.

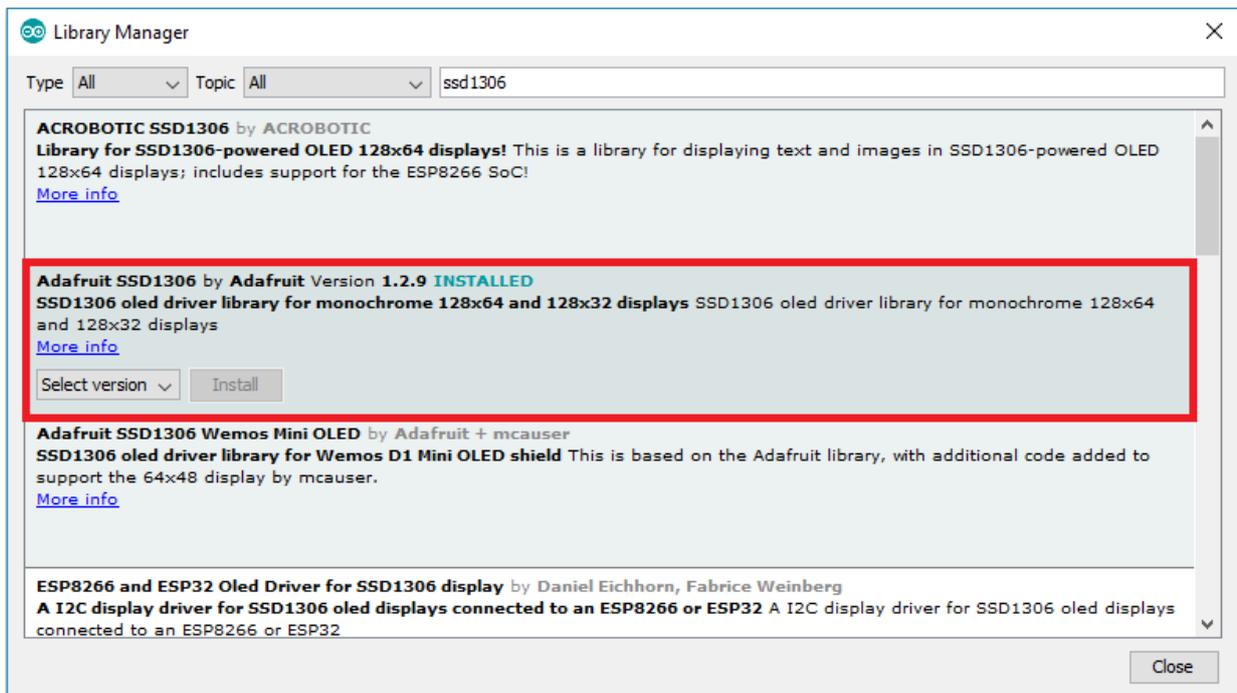
- GPIO 18: CLK
- GPIO 19: MISO
- GPIO 23: MOSI
- GPIO 5: CS

Installing SSD1306 OLED Library – ESP32

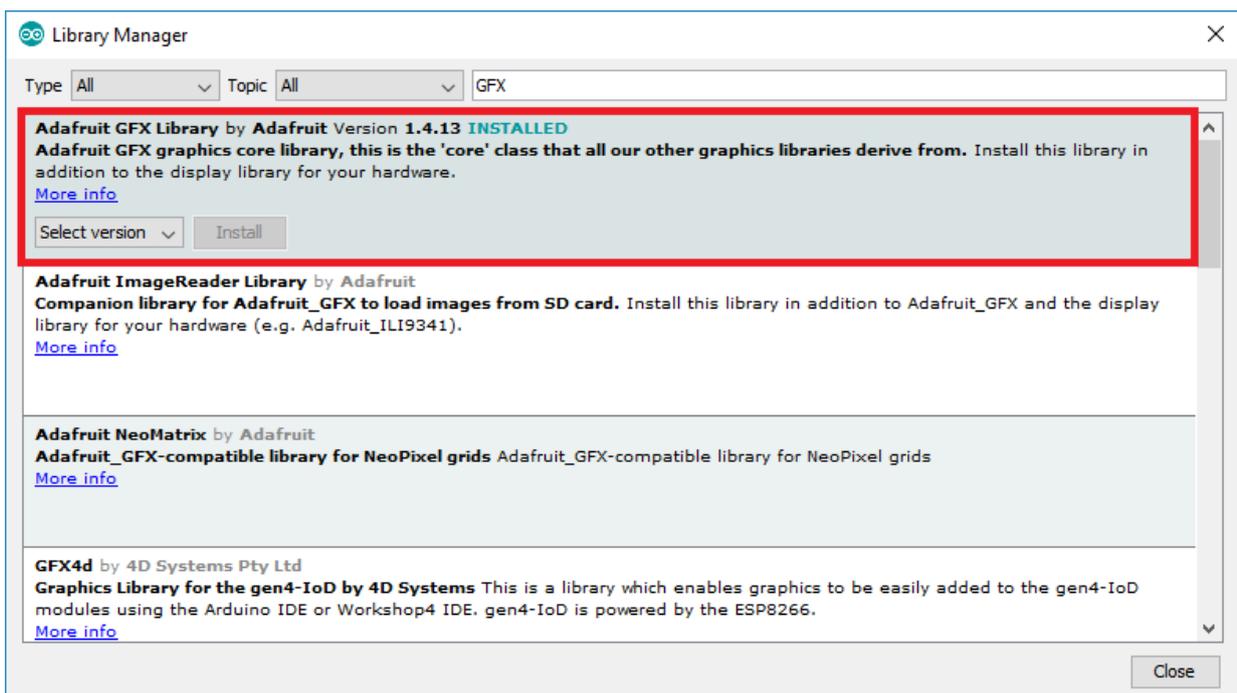
There are several libraries available to control the OLED display with the ESP32. In this project we'll use two Adafruit libraries: Adafruit_SSD1306 library and Adafruit_GFX library.

Follow the next steps to install those libraries.

1. Open your Arduino IDE and go to Sketch > Include Library > Manage Libraries. The Library Manager should open.
2. Type "SSD1306" in the search box and install the SSD1306 library from Adafruit.



3. After installing the SSD1306 library from Adafruit, type “GFX” in the search box and install the library.

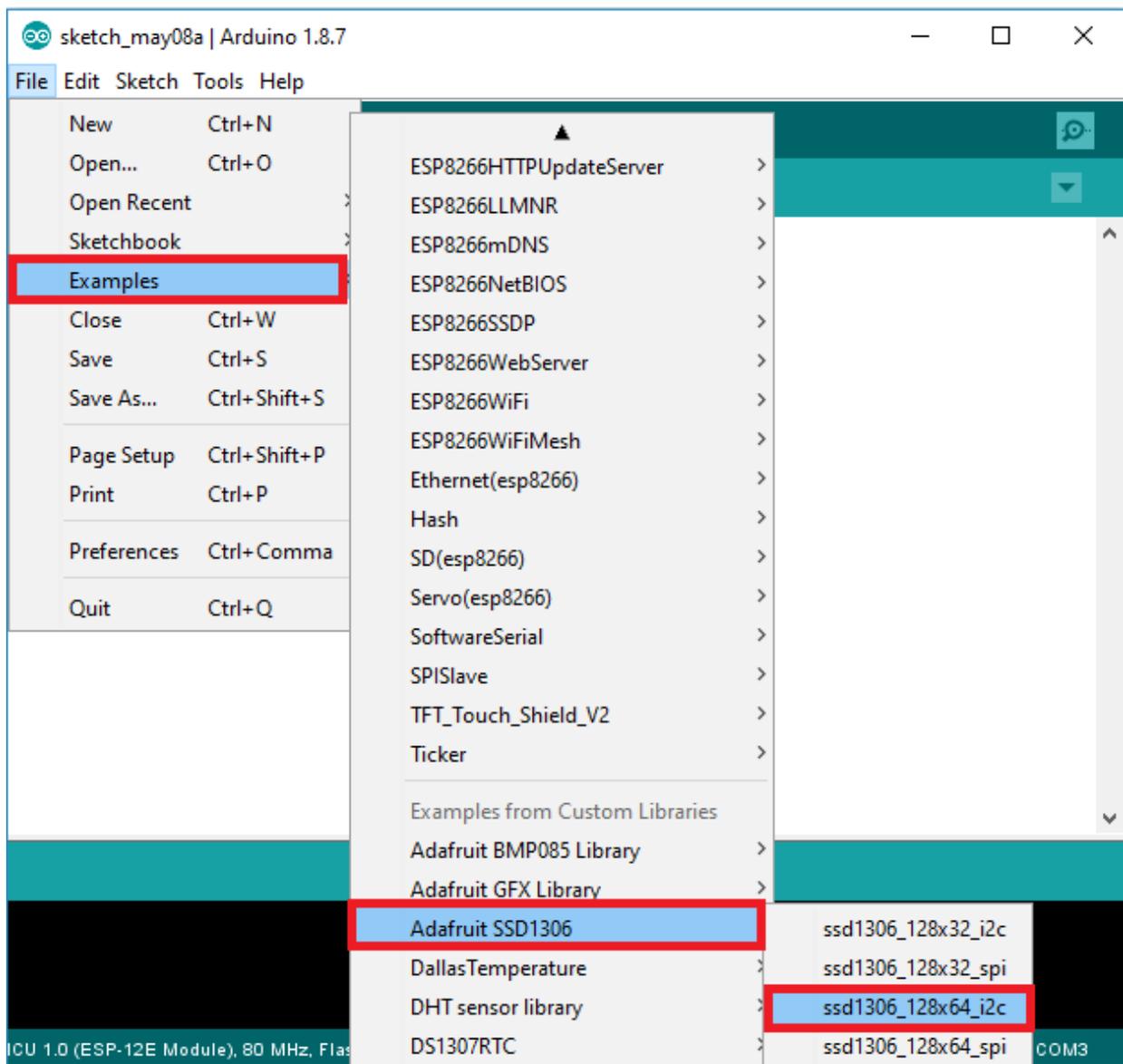


4. After installing the libraries, restart your Arduino IDE.

Testing OLED Display with ESP32

After wiring the OLED display to the ESP32 and installing all required libraries, you can use one example from the library to see if everything is working properly.

In your Arduino IDE, go to File > Examples > Adafruit SSD1306 and select the example for the display you’re using.



Write Text – OLED Display

The Adafruit library for the OLED display comes with several functions to write text. In this section, you’ll learn how to write and scroll text using the library functions.

“Hello, world!” OLED Display

The following sketch displays Hello, world! message in the OLED display.

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

```
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
```

```
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

```
void setup() {
  Serial.begin(115200);
```

```

if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for 128x64
  Serial.println(F("SSD1306 allocation failed"));
  for(;;);
}
delay(2000);
display.clearDisplay();

display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 10);
// Display static text
display.println("Hello, world!");
display.display();
}

void loop() {

}

```

After uploading the code, this is what you'll get in your OLED:



Figure 10: Text message display on OLED

Let's take a quick look on how the code works.

Importing libraries

First, you need to import the necessary libraries. The Wire library to use I2C and the Adafruit libraries to write to the display: Adafruit_GFX and Adafruit_SSD1306.

```

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

```

Initialize the OLED display

Then, you define your OLED width and height. In this example, we're using a 128×64 OLED display. If you're using other sizes, you can change that in the SCREEN_WIDTH, and SCREEN_HEIGHT variables.

```
#define SCREEN_WIDTH 128 // OLED display width, in pixels
```

```
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
```

Then, initialize a display object with the width and height defined earlier with I2C communication protocol (&Wire).

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

The (-1) parameter means that your OLED display doesn't have a RESET pin. If your OLED display does have a RESET pin, it should be connected to a GPIO. In that case, you should pass the GPIO number as a parameter.

In the setup(), initialize the Serial Monitor at a baud rate of 115200 for debugging purposes.

```
Serial.begin(115200);
```

Initialize the OLED display with the begin() method as follows:

```
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {  
  Serial.println("SSD1306 allocation failed");  
  for(;;); // Don't proceed, loop forever  
}
```

This snippet also prints a message on the Serial Monitor, in case we're not able to connect to the display.

```
Serial.println("SSD1306 allocation failed");
```

In case you're using a different OLED display, you may need to change the OLED address. In our case, the address is 0x3C.

```
if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
```

After initializing the display, add a two second delay, so that the OLED has enough time to initialize before writing text:

```
delay(2000);
```

Clear display, set font size, color and write text

After initializing the display, clear the display buffer with the clearDisplay() method:

```
display.clearDisplay();
```

Before writing text, you need to set the text size, color and where the text will be displayed in the OLED.

Set the font size using the setTextSize() method:

```
display.setTextSize(1);
```

Set the font color with the `setTextColor()` method:

```
display.setTextColor(WHITE);
```

WHITE sets white font and black background.

Define the position where the text starts using the `setCursor(x,y)` method. In this case, we're setting the text to start at the (0,0) coordinates – at the top left corner.

```
display.setCursor(0,0);
```

Finally, you can send the text to the display using the `println()` method, as follows:

```
display.println("Hello, world!");
```

Then, you need to call the `display()` method to actually display the text on the screen.

```
display.display();
```

4.5.2 Relay Driver interfacing with microcontroller

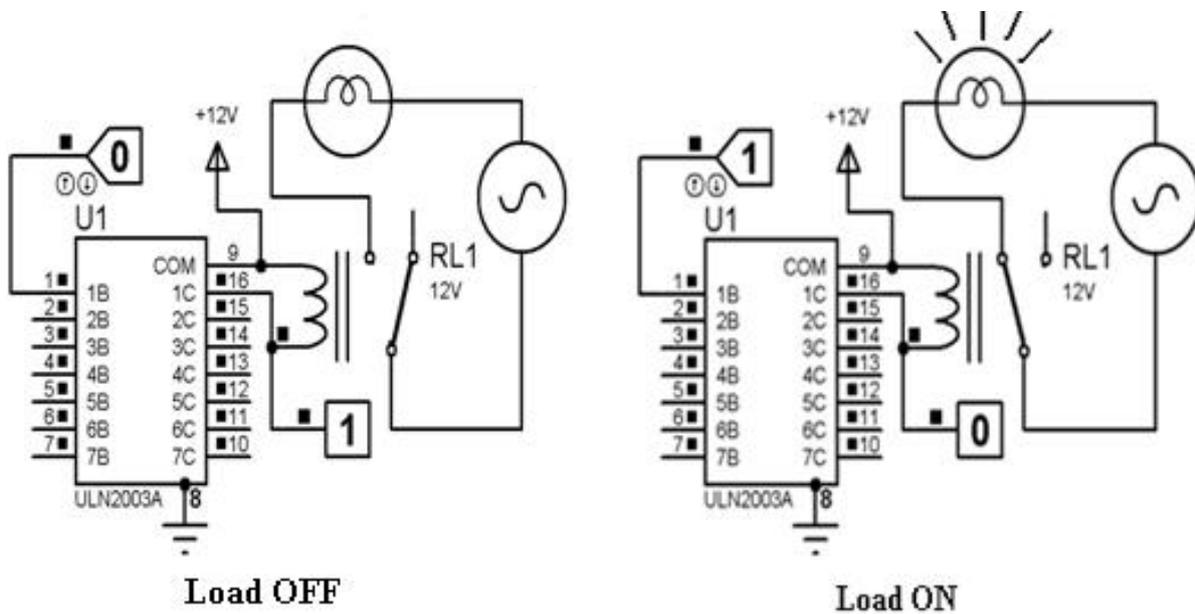


Figure 11: ULN2003A interfacing with microcontroller

The ULN2003A is a active high relay driver. 7 relays are controlled by this relay driver. Pin 1-7 are for controlling the relay which are connected to pin 10-16. For a '0' from microcontroller the corresponding relay is turned off and a '1' from microcontroller is turned on the relay.

CHAPTER 5

(Logic & Operation)

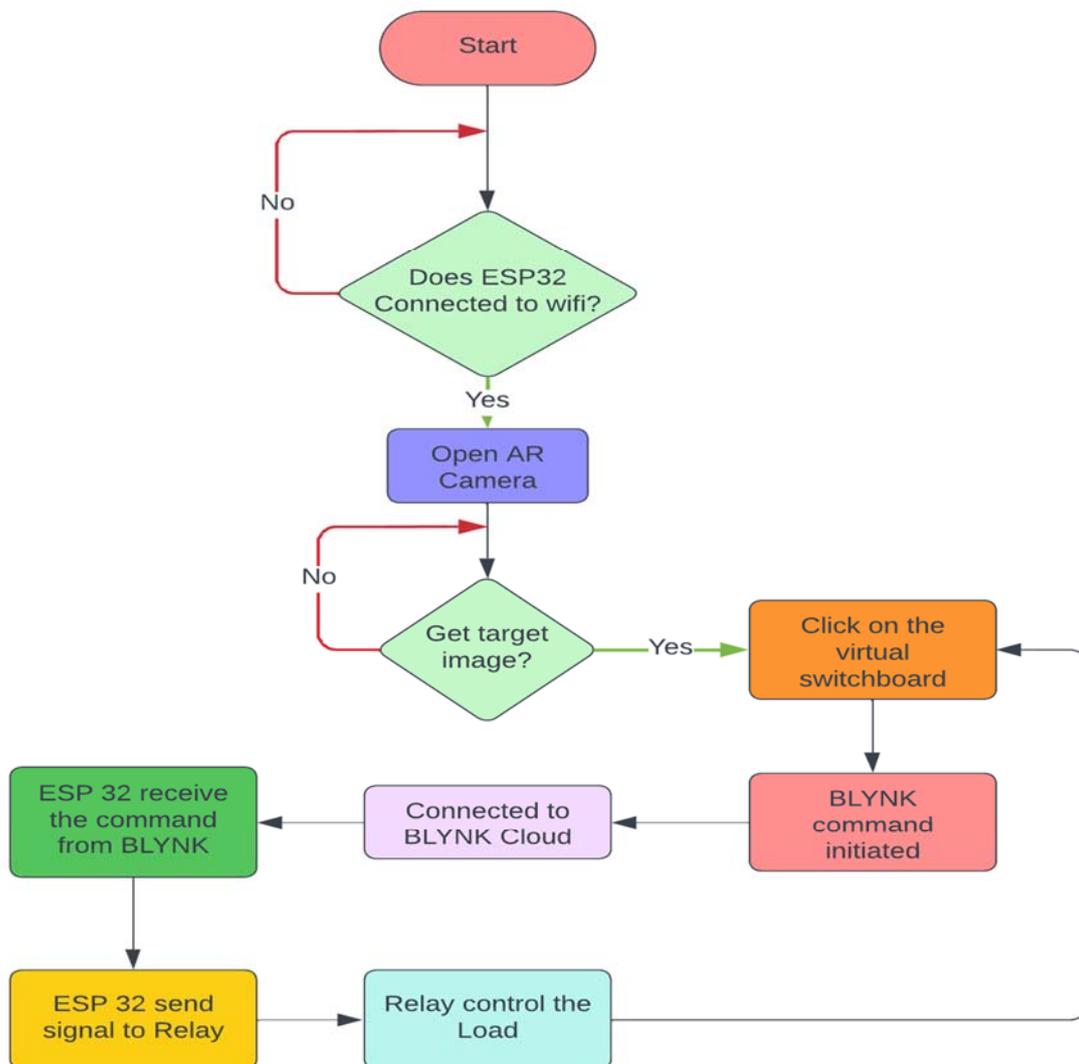
5.1 INTRODUCTION

After assembling the system, what remains is to observe its operation and efficiency of the system. The total system is divided in several sub systems, like

- Blynk cloud section
- Microcontroller section
- OLED section
- Audio feedback section
- Relay section
- AG app section

The operation of the whole circuit is depending on every sections performance.

5.2 Flow Chart



5.3 Principle & Operations

First, we need to give supply to the prototype. The supply will be taken from 5volt, 1amp adapter. As soon as we power up the circuit the indicator LED will glow to indicates that the board is powered. Now the OLED will show the message “Please connect the wifi...” if the wifi is not connected to the circuit. Here we need to connect the microcontroller ESP32 to the wifi. Now the device is connected to the wifi.

After a successful connection user need to open the app which has been built using UNITY software using Vuforia engine. Opening the app, the AR camera looks for the specified target image which has been specified while developing the app. When the AR camera gets the target image, it will generate a virtual layer upon it. This layer consists of two switches i.e, ON and OFF. By clicking the switches on the virtual switchboard, the API commands will be transmitted to the Blynk server which will further communicate with ESP32 to execute the commands and controlling the load.

The on-API command is

<https://blr1.blynk.cloud/external/api/update?token=E73pXWQ2I23F7A13HnZemyX6W97sSbTD&V0=1>

The off-API command is

<https://blr1.blynk.cloud/external/api/update?token=E73pXWQ2I23F7A13HnZemyX6W97sSbTD&V0=0>

receiving the API command from the cloud the ESP32 microcontroller will instruct the relay to switch ON or OFF respectively. The status of the relay will also be shown on a 0.96-inch OLED screen on board.

The AG based IoT switch is very much secure because the API command for controlling the load is generated only through the app, it will show the control screen only when it finds the target image. Except the developer it is impossible for others to find the target image.

5.4 Advantages of the AG load switching

A. Maintenance: It is an economical system that requires very less maintenance as compared to conventional system as it has no complicated circuits and delicate mechanisms. This saves the additional maintenance cost.

B. Cost : The main advantage of this project is it has very low cost than the conventional one available in markets. For example, some commercial controllers use microcontrollers which alone costs around Rs.900. Some controllers even have a price range of Rs.2000-Rs. 4000. But for our system, the components used are less in number and easily available. Hence losses will be less leading to a better efficiency.

C. Construction: The construction of a AG based IoT load switching system is very simple as it requires only a few components. The circuit involved is also relatively simpler. The space and power requirement to operate this system is very less.

D. Skill Required: Since the system we implement is simpler than the ones conventionally available, it can be easily made at home. The controller can also be easily operated by anyone.

5.5 Disadvantages

- The actual status of the load is unknown.
- No backup action will take for any false switching by controller itself.

5.6 Cost estimation of the project

Sl. No.	Components	Quantity	Cost (in ₹,
1	ESP32 Microcontroller	1	370
2	330 Ω resistance	1	2
3	OLED Display	1	450
4	5-volt static Relay	1	25
5	Female pin header	1	10
6	Piezo Buzzer	1	15
7	Single stand wire	3m	30
8	Wire nipper	1	180
9	Wire striper	1	100
10	Soldering Iron	1	250
11	Soldering material	1	50
12	De-soldering pump	1	130
13	BC 547	1	2
14	DC Socket	1	10
15	IN 4007	1	2
16	Bulb Holder	1	35
17	Bulb	1	75
18	Blank PCB (KS 100)	1	40
Total			1776/-

Table 3: Costing of the projects

5.7 Photographs of the prototype

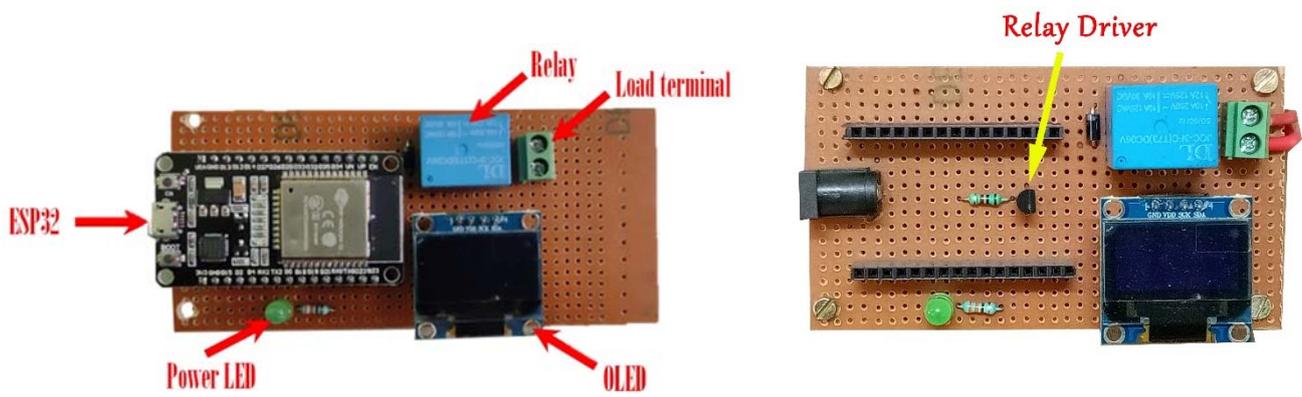


Figure 12: Main prototype and relay driver



Figure 13: Load is switch off and switch on condition



Figure 14: Different messages on the OLED screen

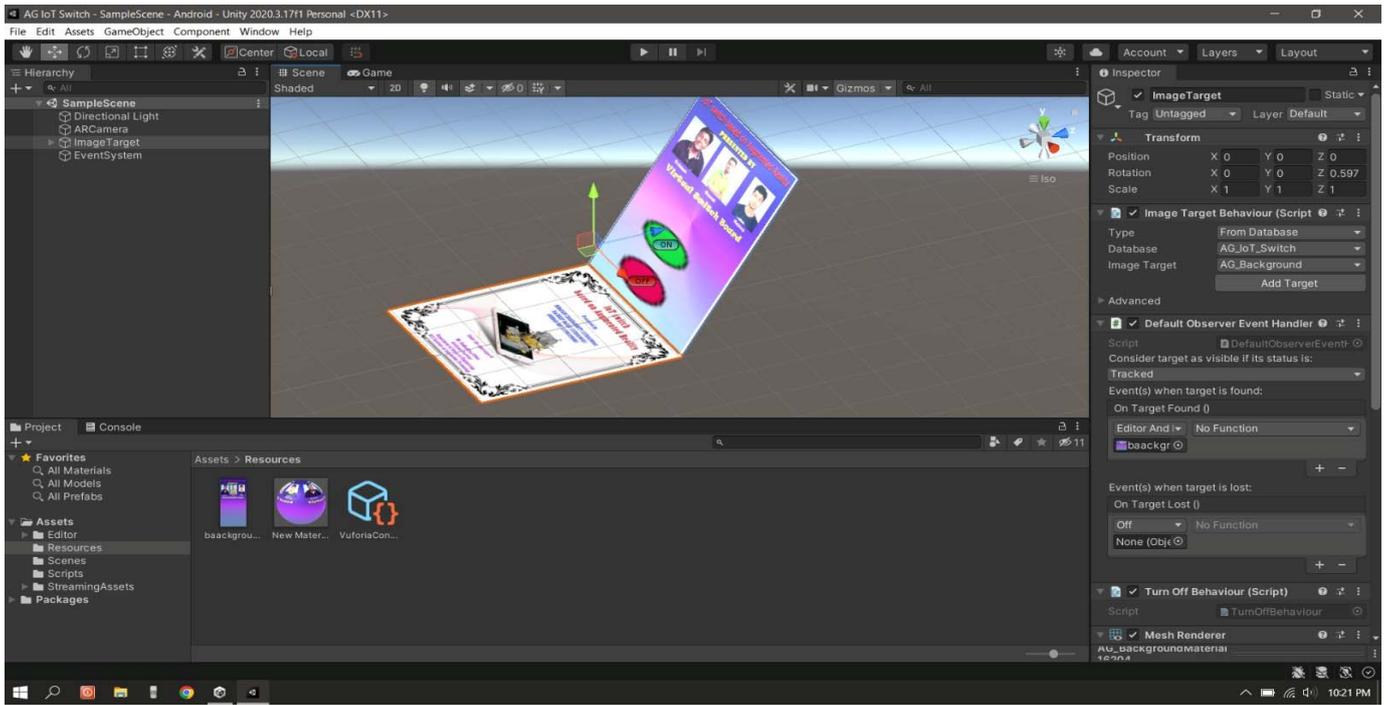


Figure 15: Making of android AG app using Unity and Vuforia

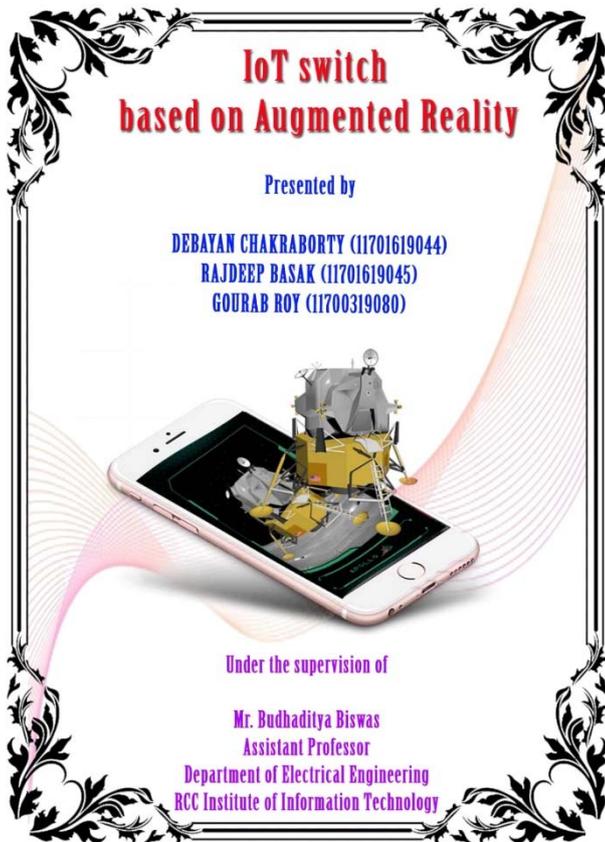


Figure 16: Target image and virtual layer



Figure 17: Using the AG app to control the load

Chapter 6

(Conclusion & Future Scope)

6.1 CONCLUSION

Here we are developed a ‘Augmented Reality based IoT switch’ circuit which could be used for domestic load switching. It’s a secure way to control the load. The circuit mainly consists of four parts such as IoT section, AG app development, audio and visual feedback, switching on and off the loads. When a command is given to the microcontroller ESP32 through blynk cloud it immediately changes the state of the load. The commands that are being passed through the microcontroller are generated through an AG app build using Unity and Vuforia software. When the AR camera finds the target image it will add an extra digital layer on the target image. This digital layer consists of 2 switches ON and OFF. By pressing the switch ON, an API command will be initiated which will communicate through blynk server and then blynk cloud will communicate to the ESP 32 microcontroller. By receiving the command from blynk the controller switch ON/OFF the load.

6.2 RESULTS

The prototype was made according to the circuit diagram and the results were as expected. The loads are switched on when the API command is received by the controller through the AG app. The loads are switched off only when the respective buttons of the on the app are pressed.

6.3 FUTURE WORK

In our present prototype the actual status of the load is not showing. It only shows the status of the ON/OFF command is generated by the controller. In future we will connect a feedback system to give the actual status of the load to the remote user. Also, the current control is ON OFF control. In future we also develop the continuous control of the load.

Chapter 7

(References)

1. **Augmented Reality Based IOT Controller**, Ritvik Khanna School of Information and Engineering Vellore Institute Of Technology Vellore, India, Vanitha M Associate Prof, School Of Information and Engineering Vellore Institute Of Technology Vellore, India 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)
2. **An Augmented Reality Prototype for supporting IoT-based Educational Activities for Energy-efficient School Buildings**, Georgios Mylonas, Christos Triantafyllis, Dimitrios Amaxilatis, Computer Technology Institute and Press “Diophantus” Research Unit 1 Patras, 26504, Greece
3. **AR Enabled IoT for a Smart and Interactive Environment: A Survey and Future Directions 1 Department of Digital Contents Engineering**, Wonkwang University 2 Department of Computer Science and Engineering, Korea University
4. **Augmented and Virtual Reality Based Monitoring and Safety System: A Prototype IoT Platform** Md Fasiul Alam, Serafeim Katsikas, Olga Beltramello , Stathes Hadjiefthymiades
5. **Augmented Reality based Mobile Application for Energy Monitoring and IoT Device Control** Abishek R, D. Vaishali, Adhitya Narayan R, Vignesh Sundar M International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075 (Online), Volume-11 Issue-1, November, 2021
6. **Augmented Reality Enabled IoT Services for Environmental Monitoring Utilising Serious Gaming Concept** Boris Pokric, Srdjan Kr, Dejan Draji, Maja Pokri , Vladimir Rajs, Živorad Mihajlovic, Petar Kneževi and Dejan Jovanovi Dunav, 21000 Novi Sad, Serbia
7. **Augmented Reality with Internet of Things** Suresh kumar S Assistant Professor Dept of computer Science Sri Krishna College of Engineering and Technology Coimbatore, India Agash C P PG Student Dept of computer Science Sri Krishna College of Engineering and Technology Coimbatore, India Ramya S Assistant Professor Department of Computer Science and Engineering Paavai Engineering College, IEEE Xplore Part Number: CFP21OAB-ART; ISBN: 978-1-7281-9537-7
8. **Interaction Paradigm of IoT using Augmented Reality** Siddharth Bhokare International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 p-ISSN: 2395-0072
9. **IoT based Home Automation System using Augmented Reality** Aaditya Autil, Bharat Kolekar², Saylee Surve³, Tejas Lad⁴, Prof. Shrikrishna Patil⁵ 1,2,3,4Student, Rajiv Gandhi Institute of Technology, Mumbai 5 Faculty, Rajiv Gandhi Institute of Technology, Mumbai ISSN: 2349-6002

Appendix A

(Hardware description)

Transformer less AC to DC power supply circuit using dropping capacitor

Production of low voltage DC power supply from AC power is the most important problem faced by many electronics developers and hobbyists. The straight forward technique is the use of a step down transformer to reduce the 230 V or 110V AC to a preferred level of low voltage AC. But *SMPS* power supply comes with the most appropriate method to create a low cost power supply by avoiding the use of bulky transformer. This circuit is so simple and it uses a voltage dropping capacitor in series with the phase line. Transformer less power supply is also called as capacitor power supply. It can generate 5V, 6V, 12V 150mA from 230V or 110V AC by using appropriate zener diodes.

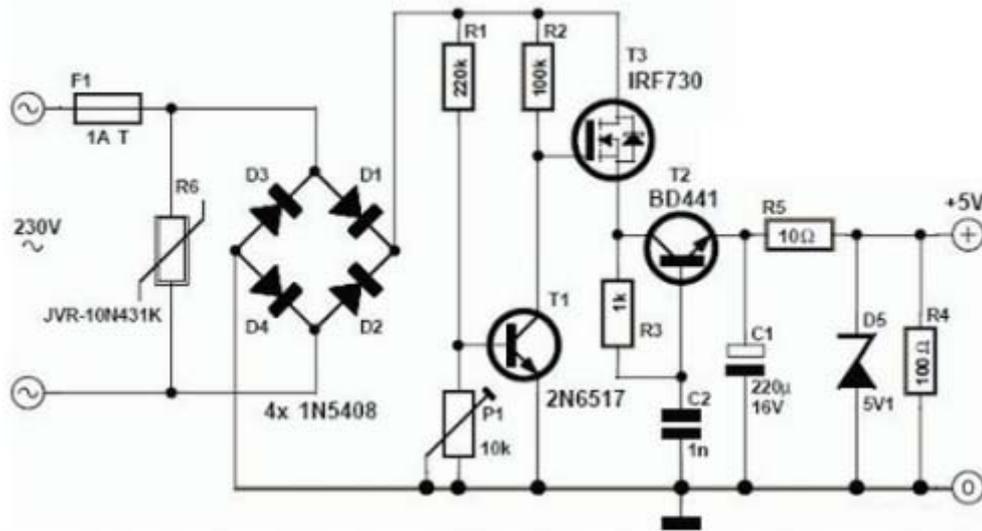


Figure 18: Transformer less SMPS 5-volt power supply

Working of Transformer less capacitor power supply

- This transformer less power supply circuit is also named as capacitor power supply since it uses a special type of AC capacitor in series with the main power line.
- A common capacitor will not do the work because the mains spikes will generate holes in the dielectric and the capacitor will be cracked by passing of current from the mains through the capacitor.
- X rated capacitor suitable for the use in AC mains is vital for reducing AC voltage.
- A X rated dropping capacitor is intended for 250V, 400V, 600V AC. Higher voltage versions are also obtainable. The dropping capacitor is non polarized so that it can be connected any way in the circuit.
- The 470kΩ resistor is a bleeder resistor that removes the stored current from the capacitor when the circuit is unplugged. It avoids the possibility of electric shock.
- Reduced AC voltage is rectified by bridge rectifier circuit. We have already discussed about bridge rectifiers. Then the ripples are removed by the 1000µF capacitor.
- This circuit provides 24 volts at 160 mA current at the output. This 24 volt DC can be regulated to necessary output voltage using an appropriate 1 watt or above zener diode.

- Here we are using 6.2V zener. You can use any type of zener diode in order to get the required output voltage.

ESP32 microcontroller

ESP32 is the SoC (System on Chip) microcontroller which has gained massive popularity recently. Whether the popularity of ESP32 grew because of the growth of IoT or whether IoT grew because of the introduction of ESP32 is debatable. If you know 10 people who have been part of the firmware development for any IoT device, chances are that 7–8 of them would have worked on ESP32 at some point. So what is the hype all about? Why has ESP32 become so popular so quickly? Let's find out.

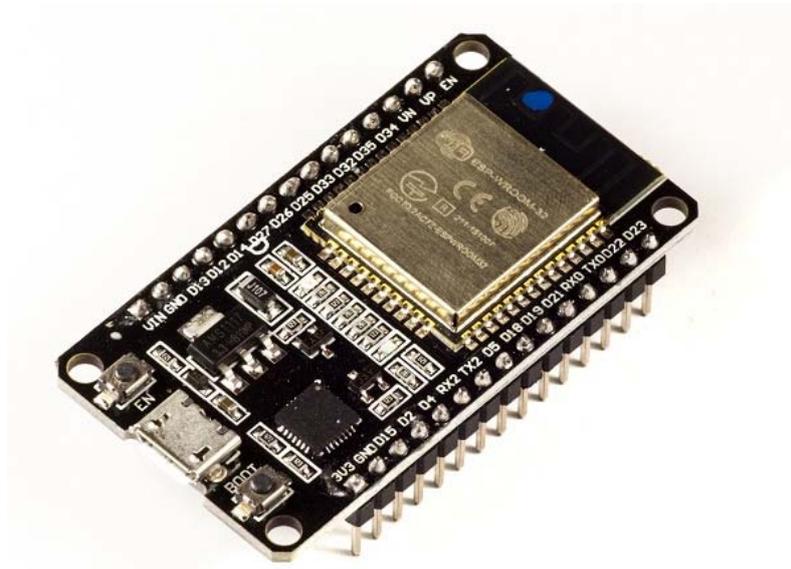


Figure 19: ESP32 Development module

Before we delve into the actual reasons for the popularity of ESP32, let's take a look at some of its important specifications. The specs listed below belong to the [ESP32 WROOM 32](#) variant.–

- Integrated Crystal– 40 MHz
- Module Interfaces– UART, SPI, I2C, PWM, ADC, DAC, GPIO, pulse counter, capacitive touch sensor
- Integrated SPI flash– 4 MB
- ROM– 448 KB (for booting and core functions)
- SRAM– 520 KB
- Integrated Connectivity Protocols– WiFi, Bluetooth, BLE
- On–chip sensor– Hall sensor
- Operating temperature range– –40 – 85 degrees Celsius
- Operating Voltage– 3.3V
- Operating Current– 80 mA (average)

With the above specifications in front of you, it is very easy to decipher the reasons for ESP32's popularity. Consider the requirements an IoT device would have from its microcontroller (μ C).

If you've gone through the previous chapter, you'd have realized that the major operational blocks of any IoT device are sensing, processing, storage, and transmitting. Therefore, to begin with, the μ C should be able to interface with a variety of sensors. It should support all the common communication protocols required for sensor interface: UART, I2C, SPI. It should have ADC and pulse counting capabilities. ESP32 fulfills all of these requirements. On top of that, it also can interface with capacitive touch sensors. Therefore, most common sensors can interface seamlessly with ESP32.

Secondly, the μ C should be able to perform basic processing of the incoming sensor data, sometimes at high speeds, and have sufficient memory to store the data. ESP32 has a max operating frequency of 40 MHz, which is sufficiently high. It has two cores, allowing parallel processing, which is a further add-on. Finally, its 520 KB SRAM is sufficiently large for processing a large array of data onboard. Many popular processes and transforms, like FFT, peak detection, RMS calculation, etc. can be performed onboard ESP32. On the storage front, ESP32 goes a step ahead of the conventional microcontrollers and provides a file system within the flash. Out of the 4 MB of onboard flash, by default, 1.5 MB is reserved as SPIFFS (SPI Flash File System). Think of it as a mini-SD Card that lies within the chip itself. You can not only store data, but also text files, images, HTML and CSS files, and a lot more within SPIFFS. People have displayed beautiful Webpages on WiFi servers created using ESP32, by storing HTML files within SPIFFS.

Finally, for transmitting data, ESP32 has integrated WiFi and Bluetooth stacks, which have proven to be a game-changer. No need to connect a separate module (like a GSM module or an LTE module) for testing cloud communication. Just have the ESP32 board and a running WiFi, and you can get started. ESP32 allows you to use WiFi in Access Point as well as Station Mode. While it supports TCP/IP, HTTP, MQTT, and other traditional communication protocols, it also supports HTTPS. Yep, you heard that right. It has a crypto-core or a crypto-accelerator, a dedicated piece of hardware whose job is to accelerate the encryption process. So you cannot only communicate with your web server, you can do so securely. BLE support is also critical for several applications. Of course, you can interface LTE or GSM or LoRa modules with ESP32. Therefore, on the 'transmitting data' front as well, ESP32 exceeds expectations.

With so many features, ESP32 would be costing a fortune, right? That's the best part. ESP32 dev modules cost in the ballpark of ₹ 500. Not only that, the chip dimensions are quite small (25 mm x 18 mm, including the antenna area), allowing its use in devices requiring a very small form factor.

Finally, ESP32 can be programmed using the Arduino IDE, making the learning curve much less steep. Isn't that great? Are you excited to get your hands dirty with ESP32? Then let's start by installing the ESP32 board in the Arduino IDE in the next chapter. See you there.

OLED

OLED displays are available in a range of sizes (such as 128×64, 128×32) and colors (such as white, blue, and dual-color OLEDs). Some OLED displays have an I2C interface, while others have an SPI interface.

One thing they all have in common, however, is that at their core is a powerful single-chip CMOS OLED driver controller – SSD1306, which handles all RAM buffering, requiring very little work from your Arduino.

In this tutorial, we'll be using both I2C and SPI 0.96-inch 128×64 OLED displays. Don't worry if your module is a different size or color; the information on this page is still useful.

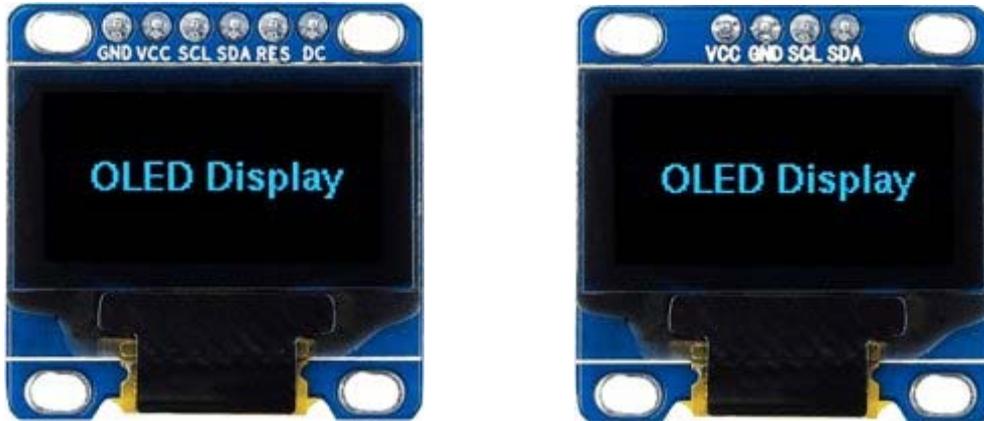


Figure 19: SPI and IIC OLED display module

Relay Driver

- The ULN2003 is a monolithic high voltage and high current Darlington transistor arrays.
- It consists of seven NPN Darlington pairs that features high-voltage outputs with common-cathode clamp diode for switching inductive loads.
- The collector-current rating of a single Darlington pair is 500mA.
- The ULN functions as an inverter.
- If the logic at input 1B is high then the output at its corresponding pin 1C will be low.

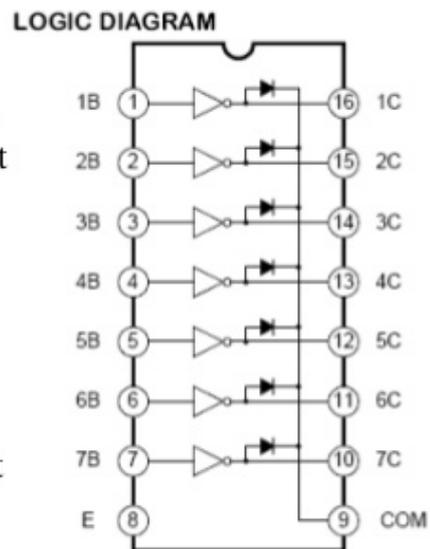


Figure 20: ULN2003A Internal Block Diagram

Resistor

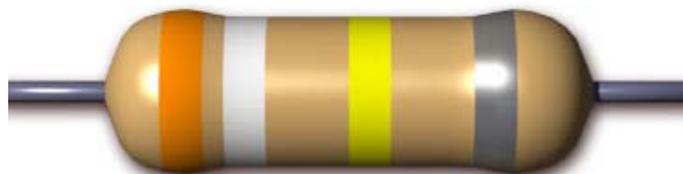


Figure 21: Resistor

Resistance is the opposition of a material to the current. It is measured in Ohms Ω . All conductors represent a certain amount of resistance, since no conductor is 100% efficient. To control the electron flow (current) in a predictable manner, we use resistors. Electronic circuits use calibrated lumped resistance to control the flow of current. Broadly speaking, resistor can be divided into two groups viz. fixed & adjustable (variable) resistors. In fixed resistors, the value is fixed & cannot be varied. In variable resistors, the resistance value can be varied by an adjuster knob. It can be divided into (a) Carbon composition (b) Wire wound (c) Special type. The most common type of resistors used in our projects is carbon type. The resistance value is normally indicated by color bands. Each resistance has four colors, one of the band on either side will be gold or silver, this is called fourth band and indicates the tolerance, others three band will give the value of resistance (see table). For example if a resistor has the following marking on it say red, violet, gold. Comparing these colored rings with the color code, its value is 27000 ohms or 27 kilo ohms and its tolerance is $\pm 5\%$. Resistor comes in various sizes (Power rating). The bigger the size, the more power rating of 1/4 watts. The four color rings on its body tells us the value of resistor value.

Color Code of the resistor

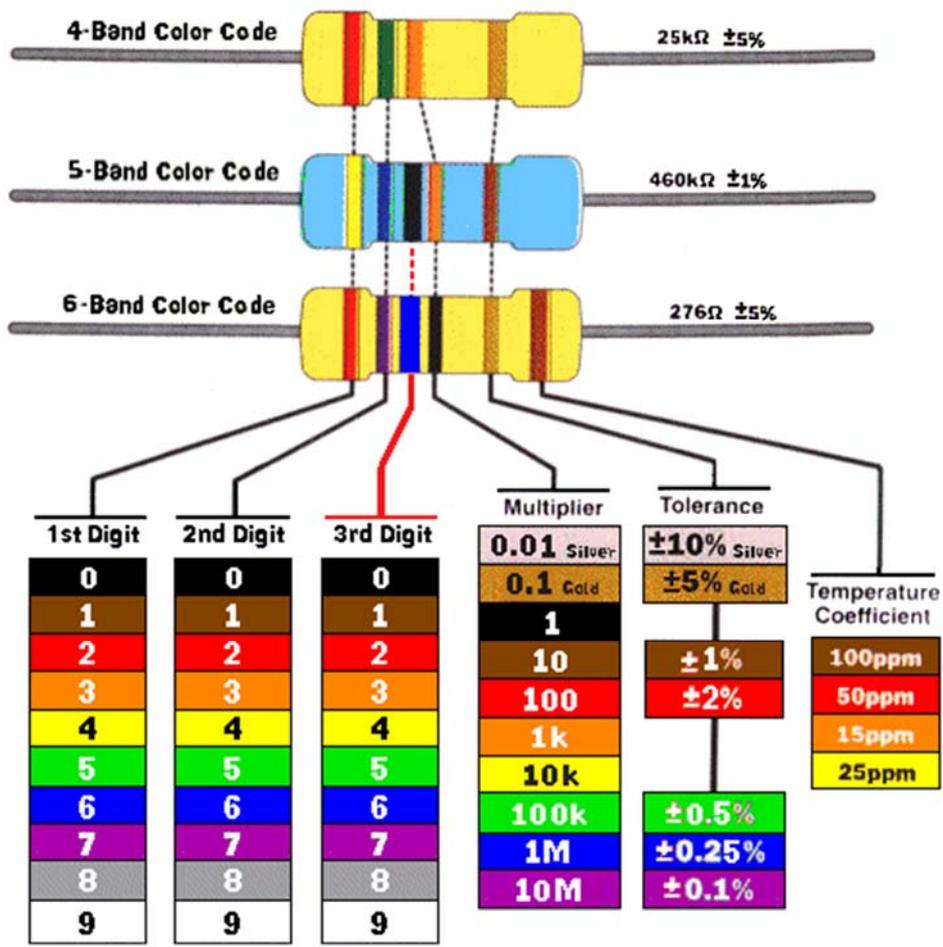


Figure 22: Color Code for resistance

RELAY

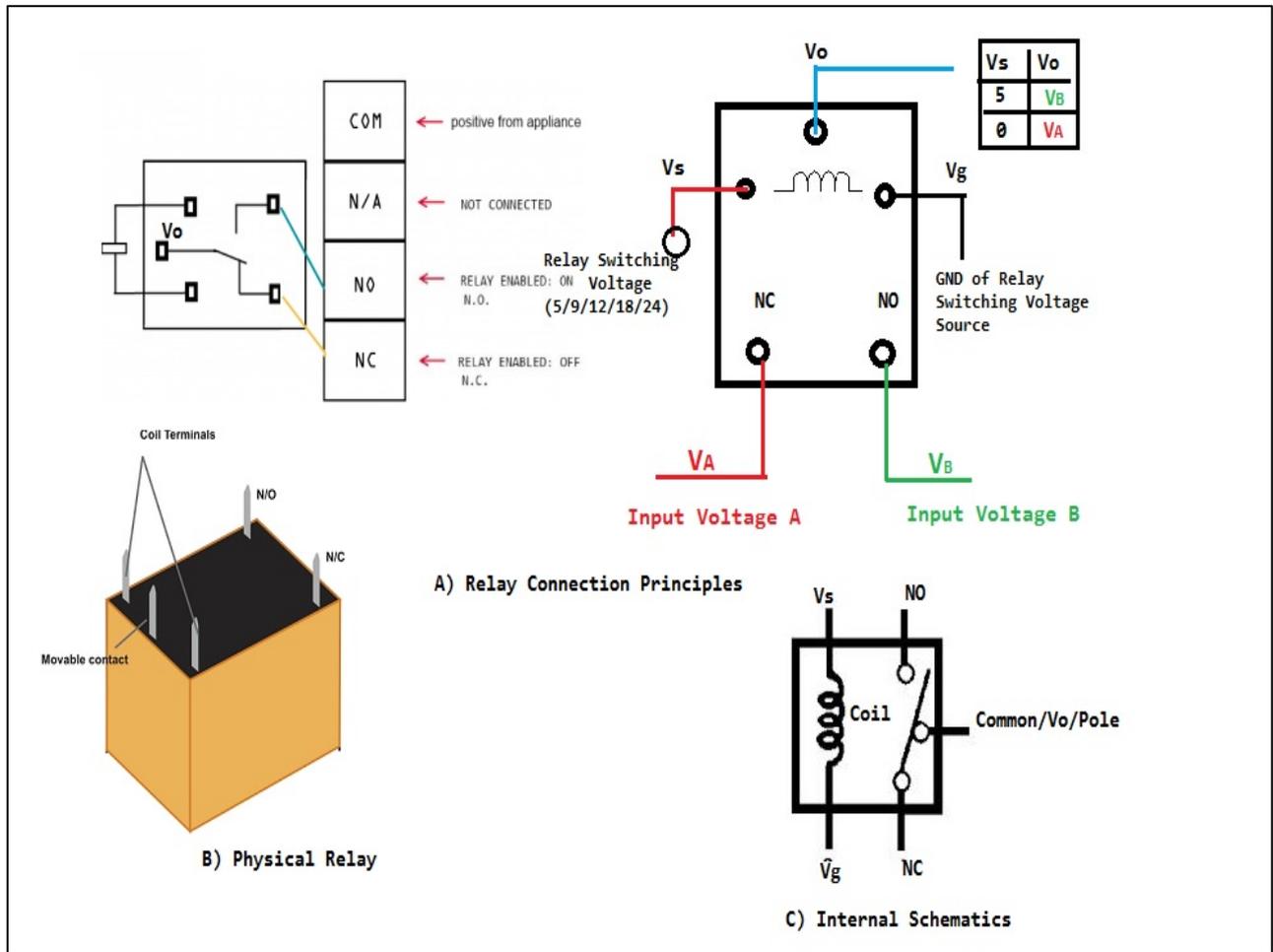


Figure 23: 6 volt Cube Relay

A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts. The coil current can be on or off so relays have two switch positions and they are double throw (changeover) switches.

The relay's switch connections are usually labeled COM (POLE), NC and NO:

COM/POLE= Common, NC and NO always connect to this, it is the moving part of the switch.

NC = Normally Closed, COM/POLE is connected to this when the relay coil is not magnetized.

NO = Normally Open, COM/POLE is connected to this when the relay coil is MAGNETIZED and vice versa.

Capacitors

It is an electronic component whose function is to accumulate charges and then release it.

To understand the concept of capacitance, consider a pair of metal plates which are placed near to each other without touching. If a battery is connected to these plates the positive pole to one and the negative pole to the other, electrons from the



Figure 24: Types of capacitors

battery will be attracted from the plate connected to the positive terminal of the battery. If the battery is then disconnected, one plate will be left with an excess of electrons, the other with a shortage, and a potential or voltage difference will exist between them. These plates will be acting as capacitors. Capacitors are of two types: - (1) **fixed type** like ceramic, polyester, electrolytic capacitors - these names refer to the material they are made of aluminum foil. (2) **Variable type** like gang condenser in radio or trimmer. In fixed type capacitors, it has two leads and its value is written over its body and variable type has three leads. Unit of measurement of a capacitor is farad denoted by the symbol F. It is a very big unit of capacitance. Small unit capacitors are pico-farad denoted by pf ($1\text{pf}=1/1000,000,000,000\text{ f}$) Above all, in case of electrolytic capacitors, its two terminals are marked as (-) and (+).

Piezo buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke. A piezoelectric element may be driven by an oscillating electronic circuit or other audio signal source, driven with a piezoelectric audio amplifier. Sounds commonly used to indicate that a button has been pressed are a click, a ring or a beep.



Figure 25: Piezo Buzzer

Blank PCB

A **printed circuit board (PCB)** mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. PCBs can be *single sided* (one copper layer), *double sided* (two copper layers) or *multi-layer* (outer and inner layers). Multi-layer PCBs allow for much higher component density. Conductors on different layers are connected with plated-through holes called vias. Advanced PCBs may contain components - capacitors, resistors or active devices - embedded in the substrate.

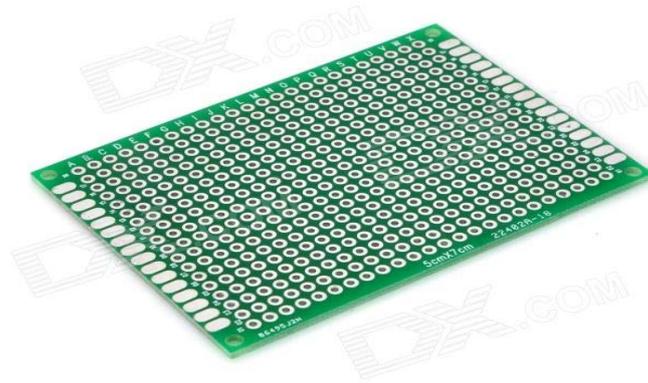


Figure 26: Blank glass epoxy PCB Board

FR-4 glass epoxy is the primary insulating substrate upon which the vast majority of rigid PCBs are produced. A thin layer of copper foil is laminated to one or both sides of an FR-4 panel. Circuitry interconnections are etched into copper layers to produce printed circuit boards. Complex circuits are produced in multiple layers.

Printed circuit boards are used in all but the simplest electronic products. Alternatives to PCBs include wire wrap and point-to-point construction. PCBs require the additional design effort to lay out the circuit, but manufacturing and assembly can be automated. Manufacturing circuits with PCBs is cheaper and faster than with other wiring methods as components are mounted and wired with one single part. Furthermore, operator wiring errors are eliminated.

Appendix B

**(AG App development using
UNITY and Vuforia)**

AG app development using Unity and Vuforia

In this project, we will use unity along with Vuforia to Make an **AR-based IoT Switch** which can **control any device** connected to a ESP32 microcontroller through an **API** (Application programming interface) request.

Structure/Work Flow:

- **STEP1:** Setup the **ESP32** along with a **LED** indicating any device.
- **STEP2:** Download and install **Unity Hub** from <https://unity.com/download>. Also, install **unity latest version** from unity hub.
- **STEP3:** Download **Vuforia Engine** for unity from <https://developer.vuforia.com/downloads/sdk>.
- **STEP4:** Setup **Target Image** which will be used to Augment buttons.
- **STEP5:** Add **buttons** over the target image in unity.
- **STEP6:** Write a **script for the buttons** to create an **HTTP request**.
- **STEP7:** Add script to **Autofocus AR Camera**.
- **STEP8:** Build a **.apk file** and test it on any **Android Device**.

Hardware and Software Requirement:

- ESP32 microcontroller
- LED with Resistor
- Unity
- Vuforia

STEP1: Setup the ESP32 along with a LED indicating any device.

First, we need to setup our target device to which we will connect our devices. Here we use blynk.cloud to control the GPIO of ESP32. The actual LED or Relay is connected to the GPIO 25 of the ESP32 and using virtual pin V0 the ESP32 is connected to Blynk server.

AT the end of this step we have two URLs to turn on and Off Our device.

1.To Turn Device ON:

<https://blr1.blynk.cloud/external/api/update?token=E73pXWQ2I23F7A13HnZemyX6W97sSbTD&V0=1>

2.To Turn Device OFF:

<https://blr1.blynk.cloud/external/api/update?token=E73pXWQ2I23F7A13HnZemyX6W97sSbTD&V0=0>

These urls are called Blynk APIs

STEP2: Download and install Unity Hub. Also, install unity latest version from unity hub.

- First goto <https://unity.com/download> to download Unity Hub.



Products

Solutions

Made with Unity

Learning

Support & Services

Download Unity

Welcome! You're here because you want to download Unity, the world's most popular development platform for creating 2D and 3D multiplatform games and interactive experiences.

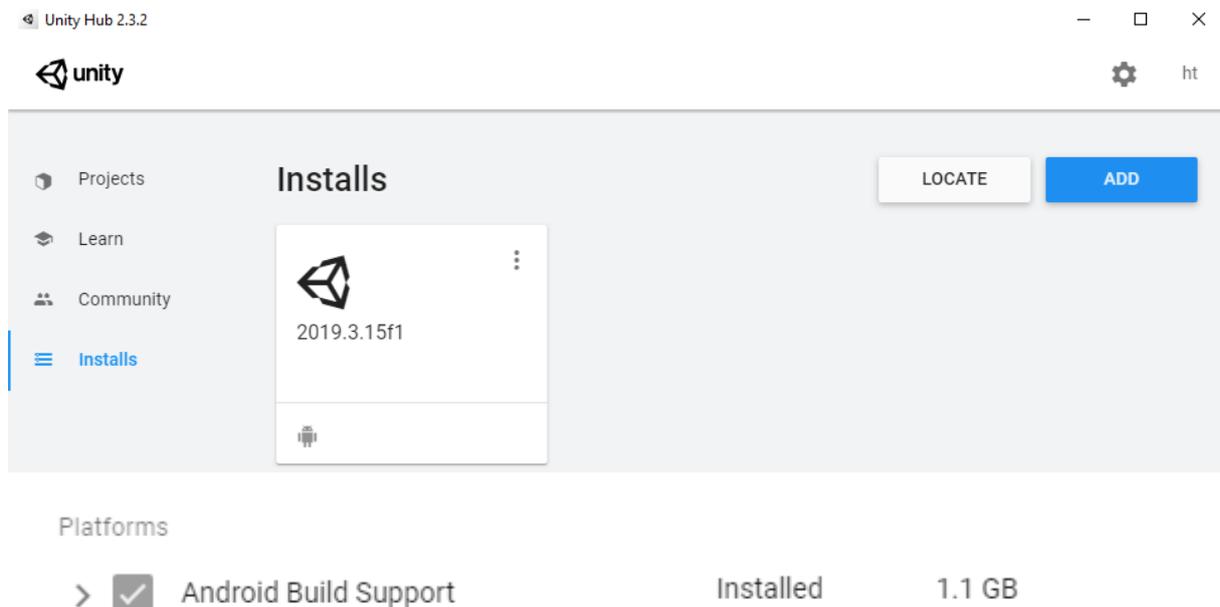
Before you download choose the version of Unity that's right for you.

Choose your Unity + download

Download Unity Hub

[Learn more about the new Unity Hub here.](#)

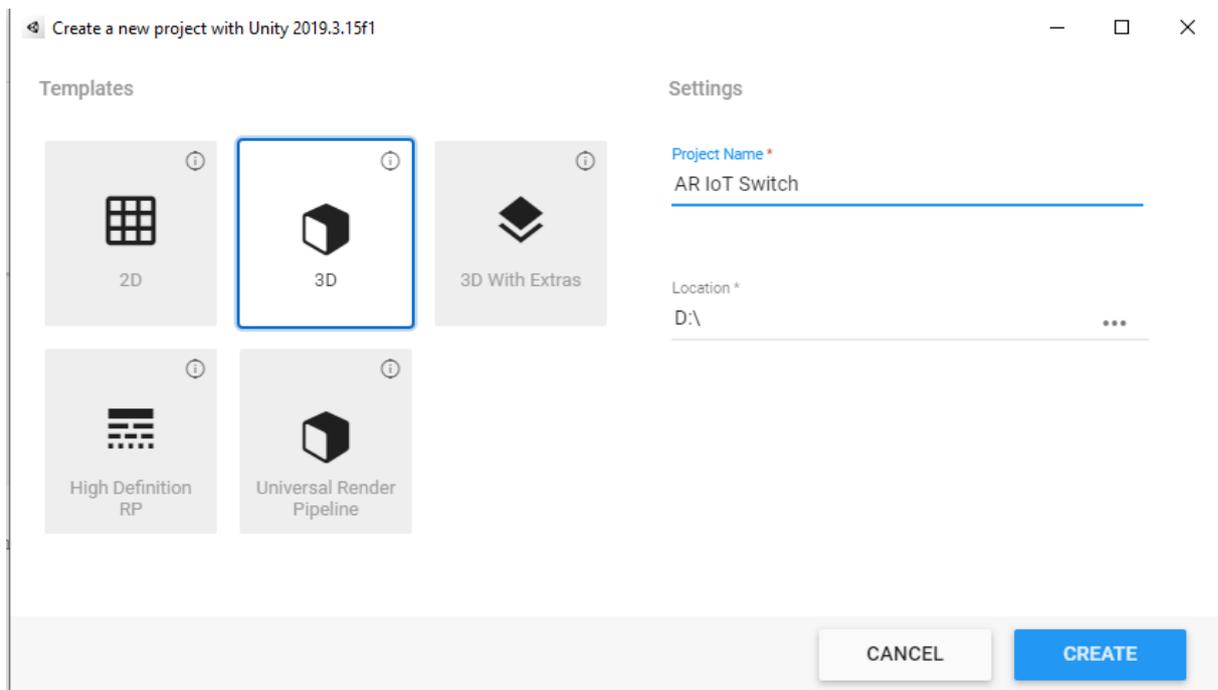
- Once Unity Hub is installed open it and install unity latest version with android build support to build apk files.



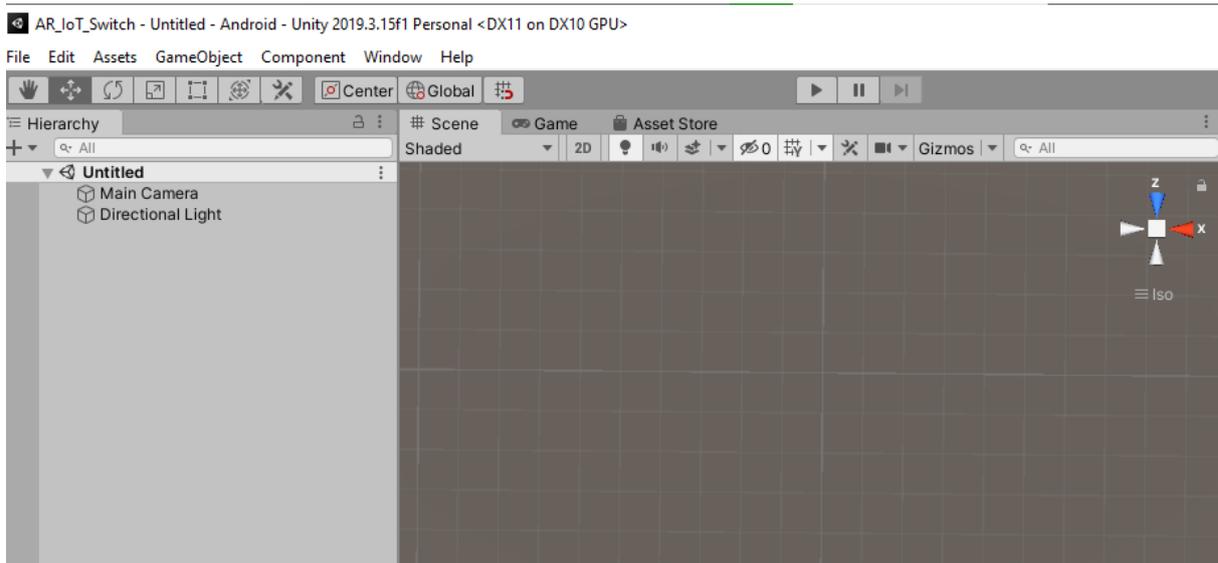
- Create a new project:



- Give a project name, select 3D template option and Click on CREATE



- You will Get a screen like this.



- Goto file and **save scene** with desired name (say “AG IoT Switch”).
- **STEP3: Download Vuforia Engine for unity from Here.**
- First Visit this link <https://developer.vuforia.com/downloads/sdk> and download.

Vuforia Engine 9.1

Use Vuforia Engine to build Augmented Reality Android, iOS, and UWP applications for mobile devices and AR glasses. Apps can be built with Unity, Android Studio, Xcode, and Visual Studio. Vuforia Engine can also be accessed through the Unity Package Manager by adding Vuforia's package repository with the script below.

```

Add Vuforia Engine to a Unity Project or upgrade to the latest
version
add-vuforia-package-9-1-7.unitypackage (2.57 KB)
    
```



- Login and agree the terms.

vuforia engine™ developer portal Home Pricing Downloads Library Develop Support Log In | Register

Login

Login with your Vuforia developer account to download software, get license keys and participate in the Vuforia community.

Not Registered? Create account now

Forgot password?

Login

PTC, Inc. Resources Community AR Products Stay cutting edge. Get the Vuforia newsletter.

- **Double click** the in unity
- Click

Software License

In order to continue downloading, you need to agree to the terms below.

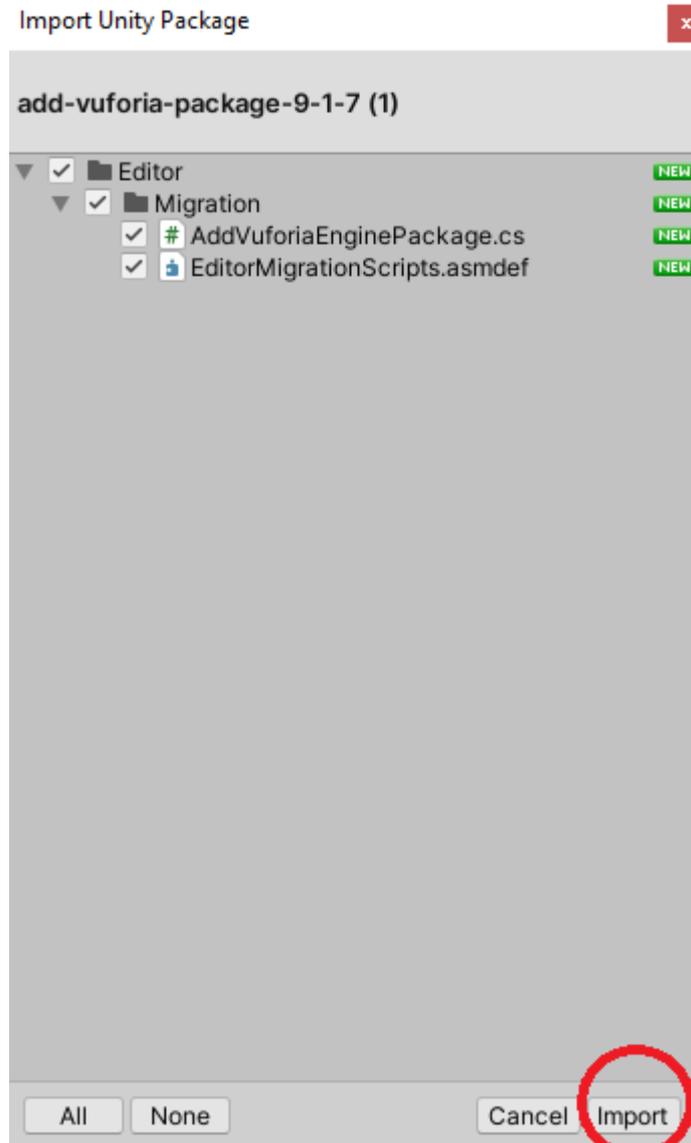
Use of this Vuforia™ Software Development Kit or extension (the "Software") is subject in all respects to all of the terms and conditions of the Vuforia Developer Agreement, which may be found at <https://developer.vuforia.com/legal/vuforia-developer-agreement>. If you do not agree to all the terms and conditions of the Vuforia Developer Agreement, then you must not retain or in any manner use any of the Software.

Print

Decline

I Agree

downloaded file to open on **Import**.

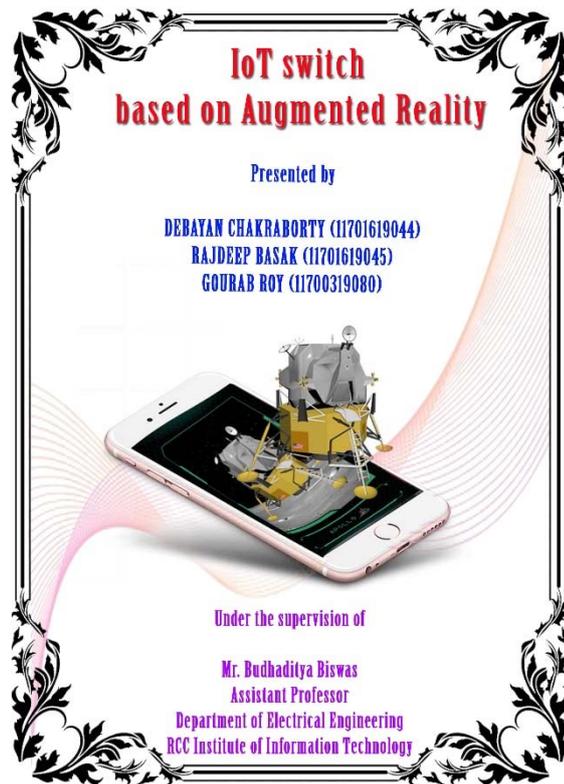


Now your unity Software is ready along with Vuforia Engine.

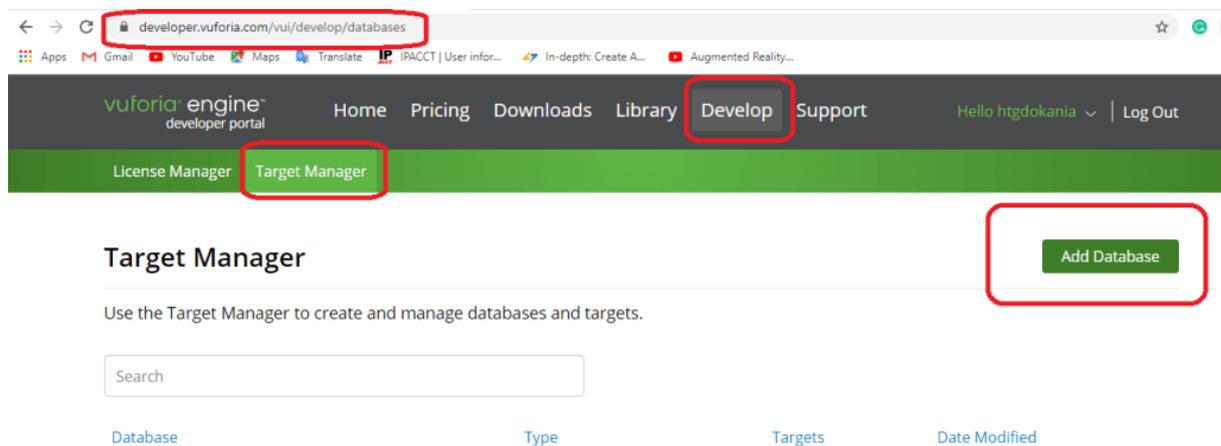
STEP4: Setup Target Image

Next, we will need a target image. Whenever camera will detect this image, it will Augment our ON/OFF buttons over this image to control any Device.

We Will use the following image as our target image. You can use any image of your choice. (make sure to fill the entire image area)



- Now Goto <https://developer.vuforia.com/vui/develop/databases> and click on Add Database under Target Manager tab.



- Give a name. select Type: Device and click on Create:

Create Database

Database Name *
IoT

Type:

- Device
- Cloud
- VuMark

Cancel Create

- Click on Add target and Browse Your target image as shown.

The screenshot shows the 'Target Manager' interface. At the top, there are two tabs: 'License Manager' and 'Target Manager'. Below the tabs, the breadcrumb 'Target Manager > IoT' is visible. The main content area displays 'IoT' with an 'Edit Name' link, and 'Type: Device'. Below this, there is a 'Targets (1)' section. At the bottom of the interface, there are two buttons: 'Add Target' and 'Download Database (All)'. A table header is visible at the bottom with columns: 'Target Name', 'Type', 'Rating', 'Status', and 'Date Modified'.

- Give a **Width** of your choice (say 4) and a **name** (say bulb). Finally **Click on Add**.

Add Target

Type:



File:

WhatsApp Image 2020-06-12 at 4.17.29 PM.jpeg

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

- The Better the **Star Rating** You get the better will be the results.(In my case i got **4 stars** for the above image).

License Manager Target Manager

Target Manager > IoT

IoT [Edit Name](#)
Type: Device

Targets (1)

<input type="checkbox"/>	Target Name	Type	Rating ⓘ	Status ▾	Date Modified
<input type="checkbox"/>	Bulb	Single Image	★★★★☆	Active	Jun 12, 2020 16:18

- Click on download Database (All). Select Unity Editor and Download.

Download Database

1 of 1 active targets will be downloaded

Name:

IoT

Select a development platform:

Android Studio, Xcode or Visual Studio

Unity Editor

Cancel

Download

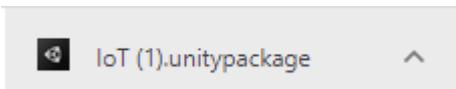
Compiling Database



Compiling a database with Object targets may take several minutes

Cancel

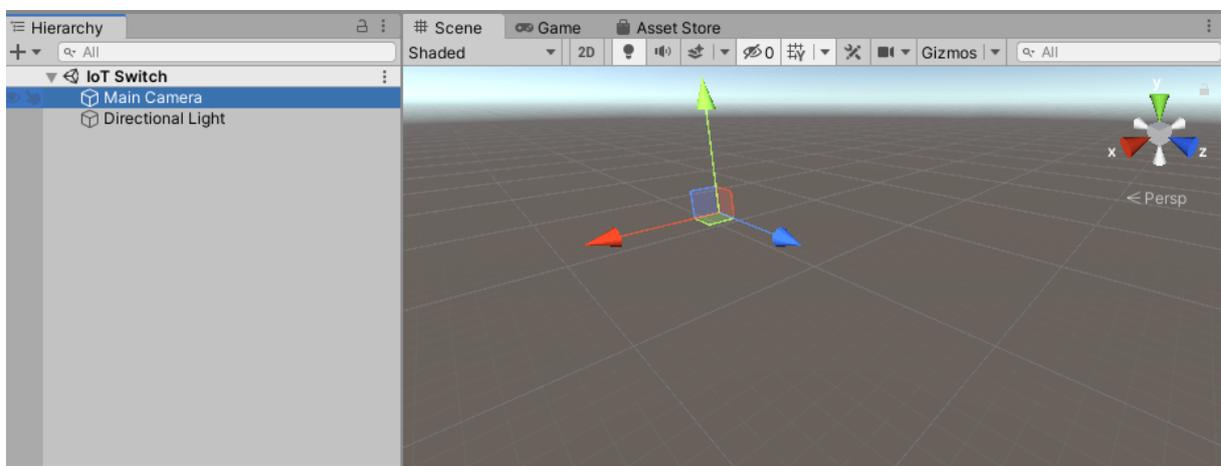
- Double click the Package Downloaded and import in unity Just like we did Previously.



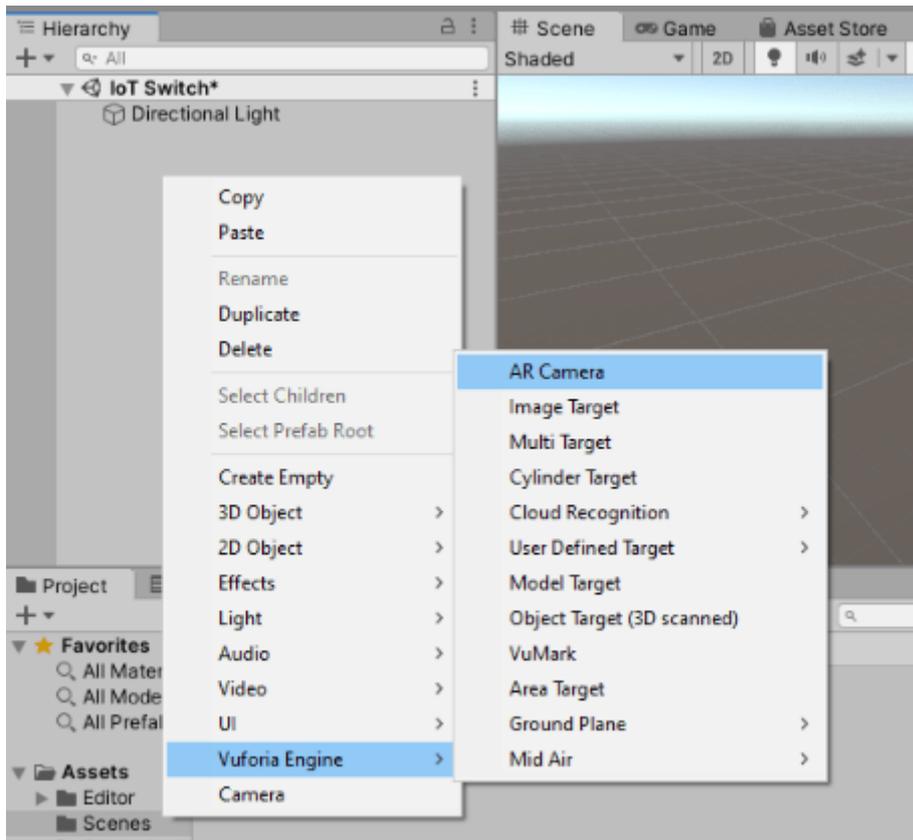


STEP5: Add buttons over the target image in unity.

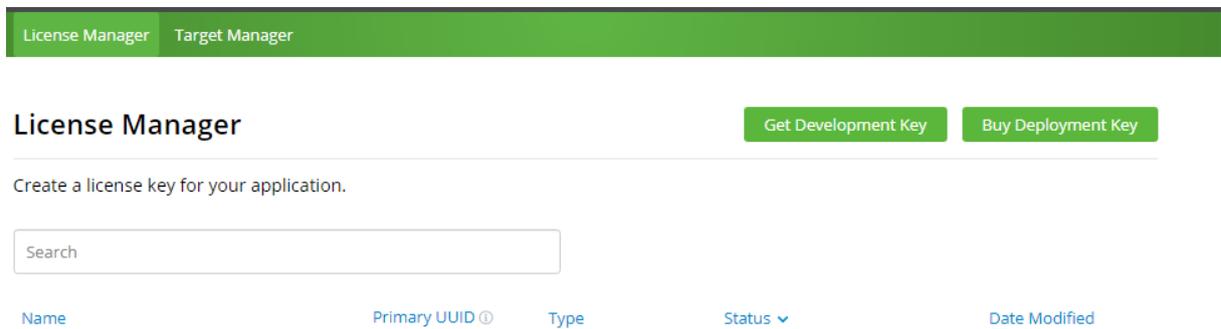
- First Open the scene and Delete the main camera.(simply select it and press delete key.)



- Add AR Camera from Vuforia Engine.



- Next we will add licence key. Goto Licence Manager Tab and click on Get Development Key.



- Give a name, Tick the checkbox and Confirm

[Back To License Manager](#)

Add a free Development License Key

License Name *

IoT

You can change this later

License Key

Develop

Price: No Charge

Reco Usage: 1,000 per month

Cloud Targets: 1,000

VuMark Templates: 1 Active

VuMarks: 100

By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#).

Cancel

Confirm

- Next click on the New Licence.

License Manager

[Get Development Key](#)[Buy Deployment Key](#)

Create a license key for your application.

Search

Name	Primary UUID [ⓘ]	Type	Status [▼]	Date Modified
IoT	N/A	Develop	Active	Jun 12, 2020

- Copy the licence

License Manager > IoT

IoT

[Edit Name](#) [Delete License Key](#)

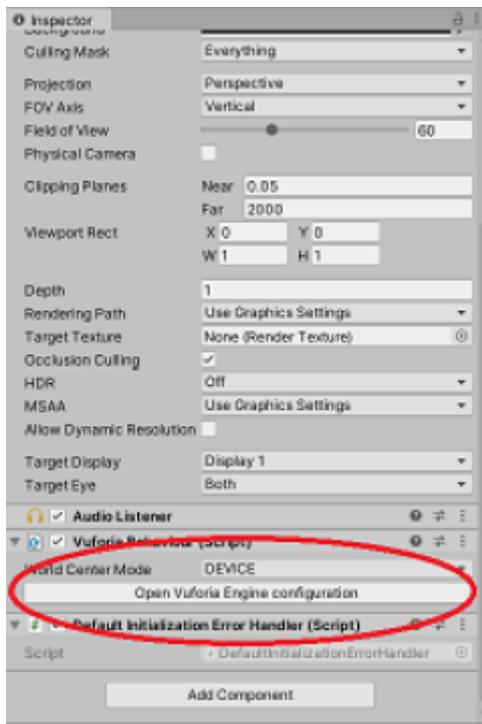
License Key

Usage

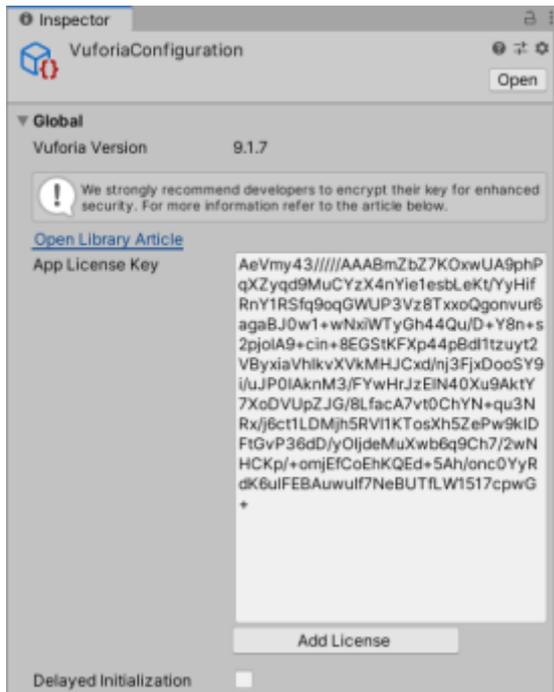
Please copy the license key below into your app

```
AeVmy49////AAABmZbZ7K0xwUA9phPqXZyqd9MuCYeX4nYie1esbLeKt/VyHifRnY1RSEq9oqGWUP9Vz8TxxoQgonvur6agaBJ0w1+wNx
iWiyGh44Qu/D+Y8n+s2pjo1A9+cin+8EGStKFXp44pBdIltzuyt2VByxiaVh1kvXVkmHJCxd/nj9FjxDoo8Y9i/uJP0IAknM9/FYwHzJzE
1N40Xu9AktY7XoDVUp2JG/8LfacA7vt0ChYN+qu3NRx/j6ct1LDMjhSRV11KToeXh52ePw9k1DFtGv936dD/yOIjdeMuXwb6q9Ch7/2wNH
CKp/+omjEeCoEhKQEEd+8Ah/one0YyRdK6ulFEBAuwuIf7NeBUIfLW1517cpwG+
```

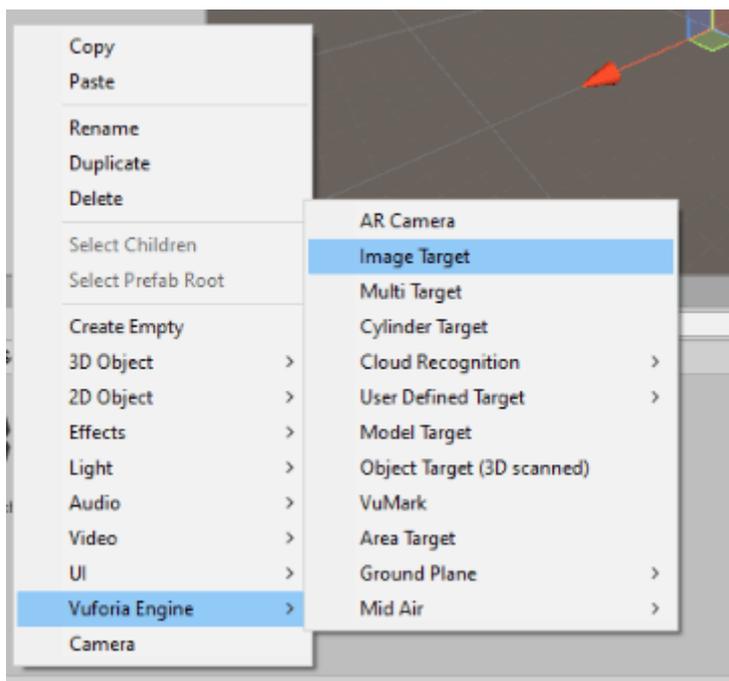
- open unity ,select AR camera and in the inspector panel click on Open Vuforia Engine Config.



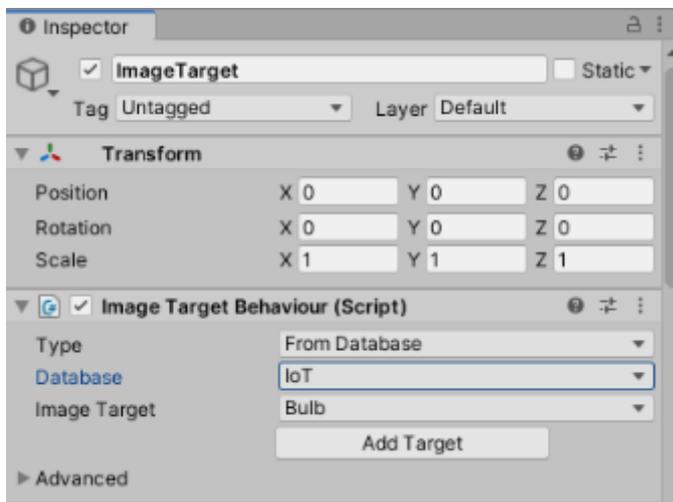
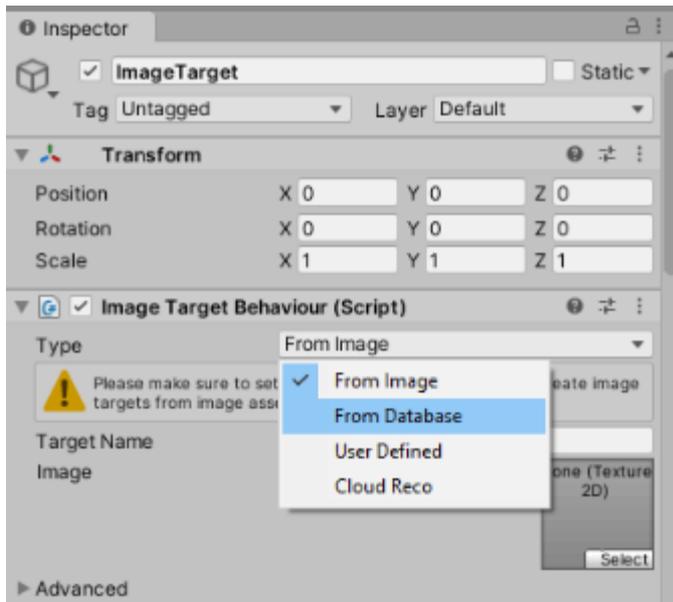
- Paste the licence key you copied in earlier step.



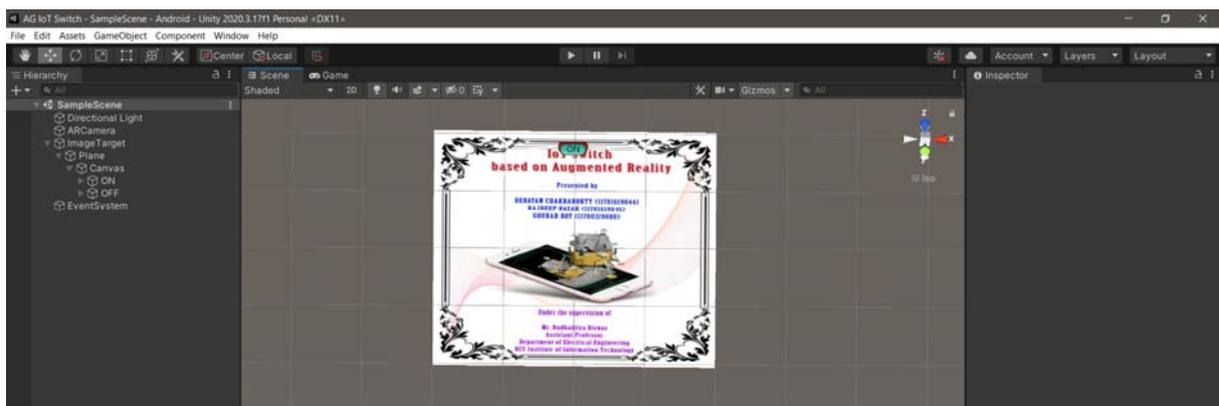
- Next add image target from Vuforia Engine.



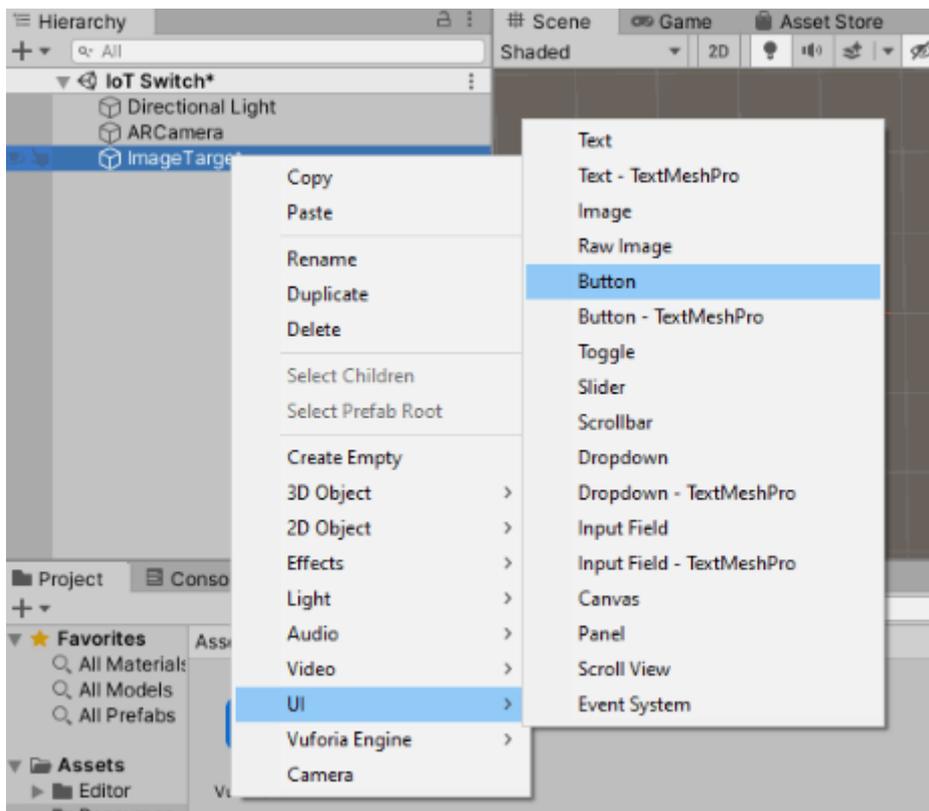
- Next , Select Image target and in the inspector panel on the right side of the screen load the image from database.



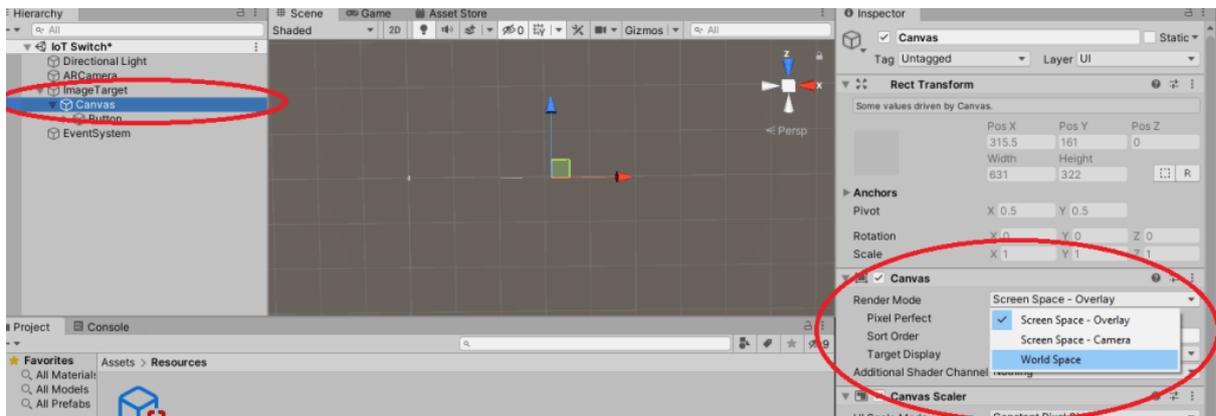
- once this is done Our screen will look Something like this



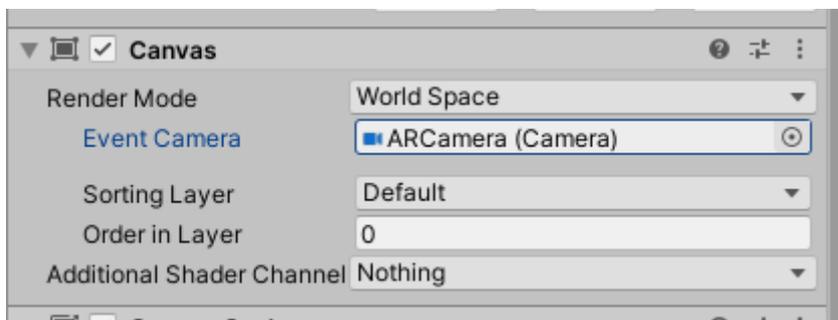
- Next Add a button from UI. Right click on Image Target and select button from UI.



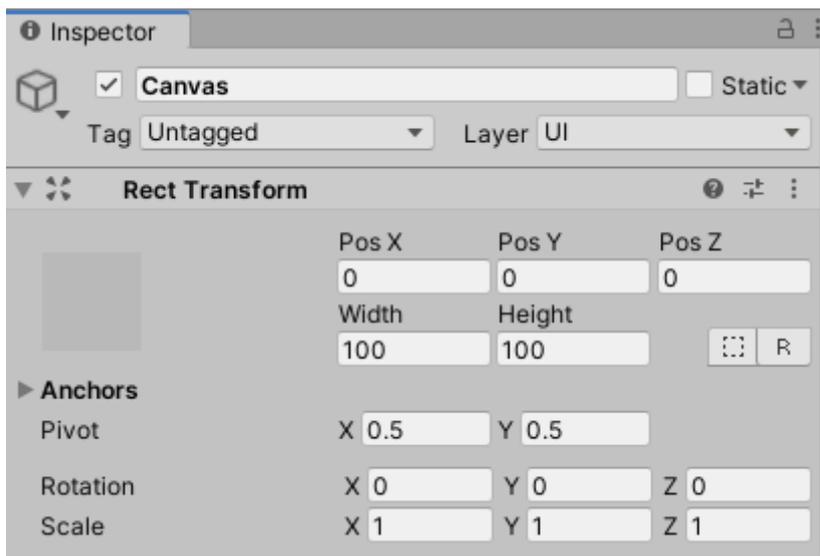
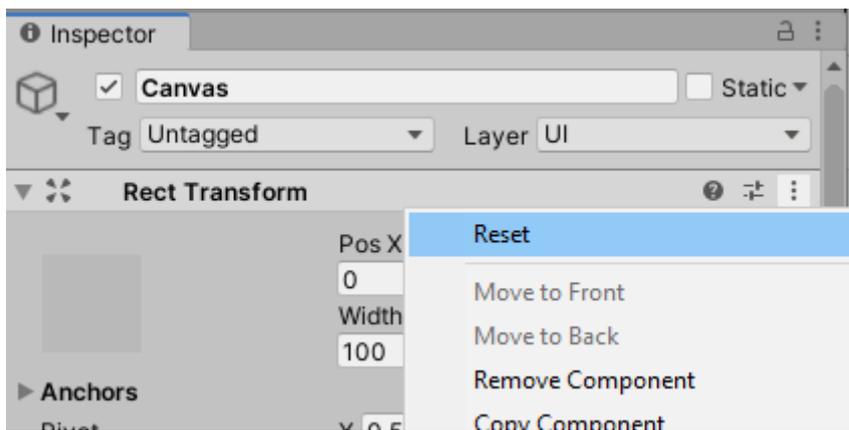
- Next click on canvas and Change Render Mode to World Space.



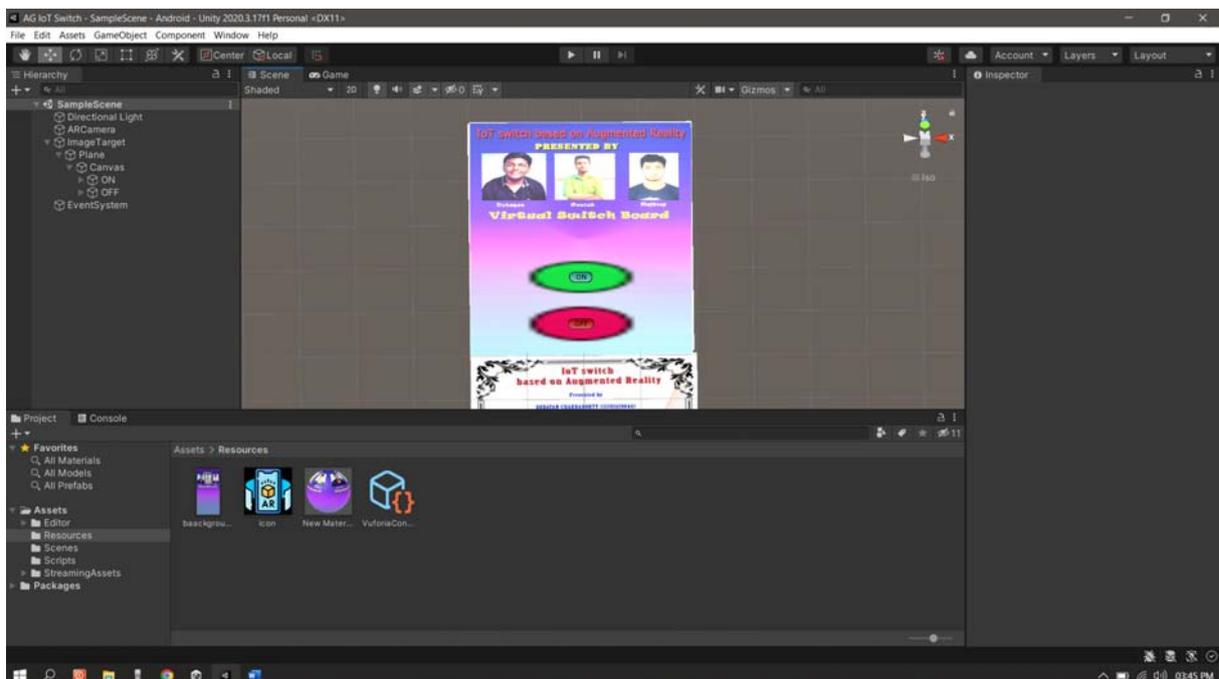
- Also set Event Camera as AR camera.(Simply drag and drop)



- Also Reset the canvas position to (0,0,0). so that the buttons will be at **same position as the target image**.



- Next Adjust the button on top of the Target image. (Rotate and resize to fit like this)



- Right click and duplicate the button
- Rename the Buttons and Also its text as shown to ON and OFF.

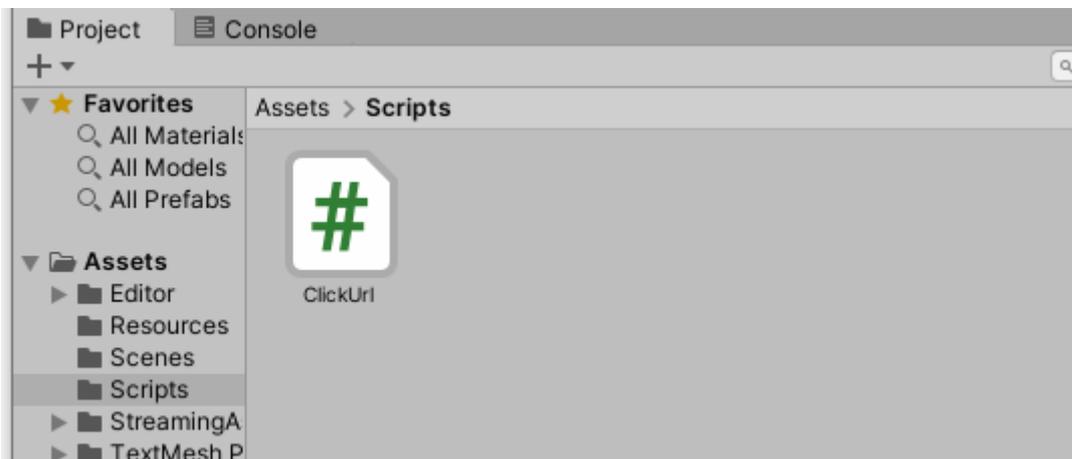
- Now if you run and bring the target image in front of the PC's camera You will see the buttons on top of it.



STEP6: Write a script for the buttons to create an HTTP request.

So Now finally we need some code which will create a http request to our Webserver whenever we press the On /OFF buttons.

- First inside assets Folder we create a new folder “Scripts” and inside this folder we create a **new C# file** by name “ClickUrl”.



Add the following code to the C# file.

- First add following namespaces
 - 1 **using** System.Collections;
 - 2 **using** System.Collections.Generic;
 - 3 **using** UnityEngine;
 - 4 **using** UnityEngine.Networking;
- Next inside the ClickUrl class, we create a public string URL to hold the URL for each button.
- Also, create a Function named Open () to Create the HTTP Get Request. We will Call this open() function on click action of the button.

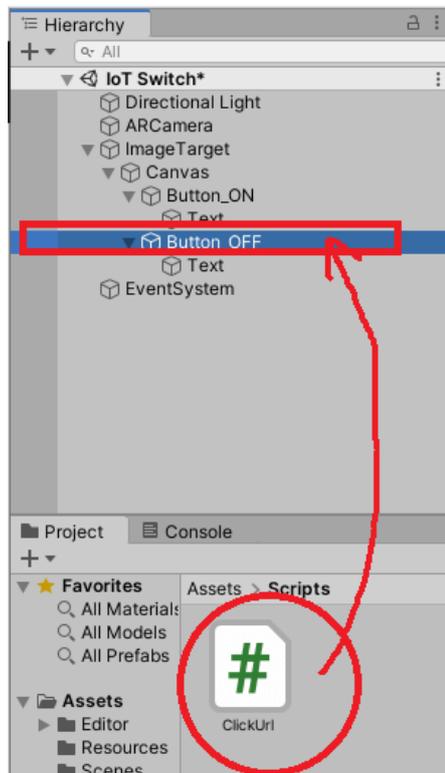
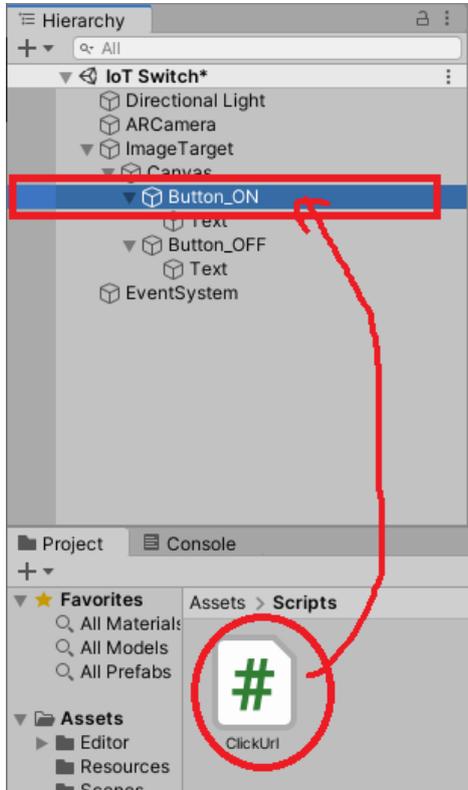
- We create a Coroutine and create Get Request and wait until it returns.

```

5      public class ClickUrl : MonoBehaviour
6      {
7          public string url;
8          public void open()
9          {
10             StartCoroutine(GetRequest(url));
11         }
12
13         IEnumerator GetRequest(string uri)
14         {
15             using (UnityWebRequest webRequest = UnityWebRequest.Get(uri))
16             {
17                 // Request and wait for the desired page.
18                 yield return webRequest.SendWebRequest();
19             }
20         }
21     }
22

```

- Save the code and Come back to unity screen.
- Now attach this script to both the buttons(On,Off).(Simply drag and drop.)



Inside the Inspector Panel you can see that the Script component is added to each button element. .

- Also a URL field is present(which we made public in our script). **Type in the Required URL in here.**

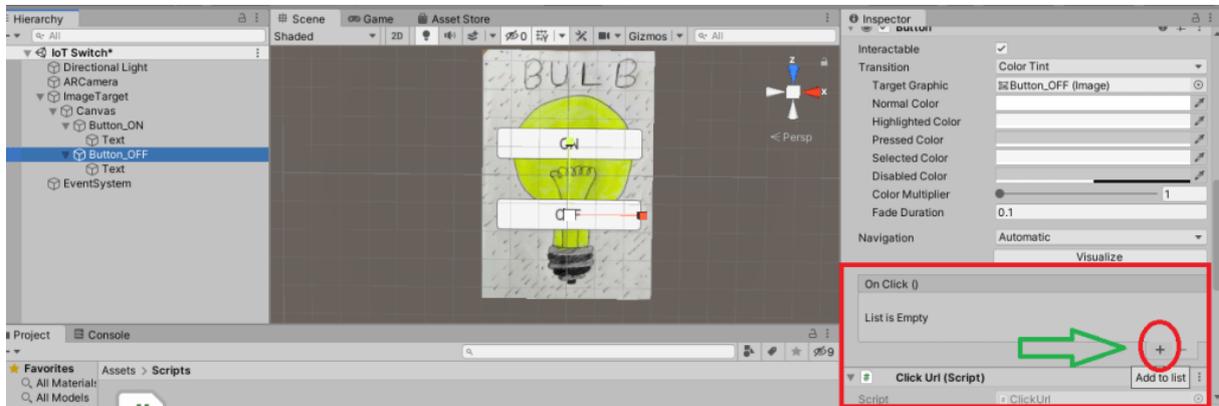
1.To Turn Device ON:

<https://blr1.blynk.cloud/external/api/update?token=E73pXWQ2I23F7A13HnZemyX6W97sSbTD&V0=1>

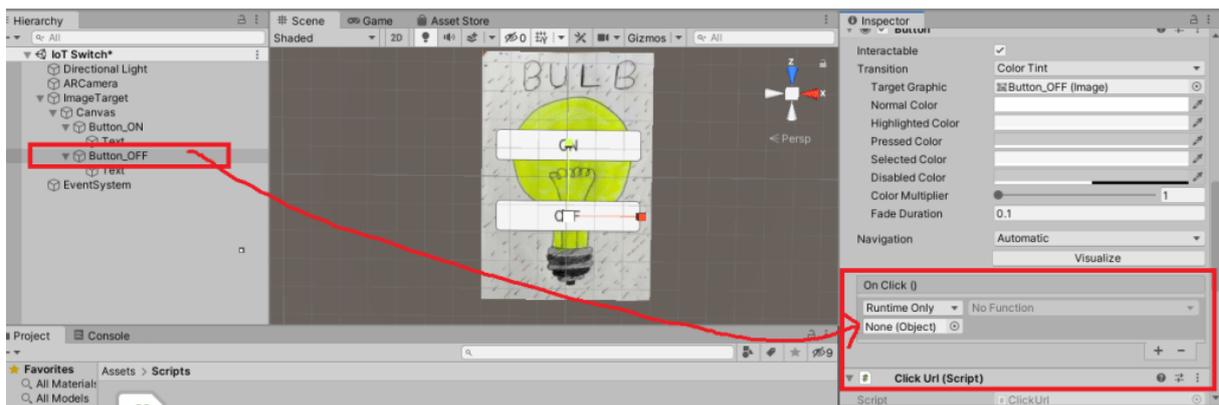
2.To Turn Device OFF:

<https://blr1.blynk.cloud/external/api/update?token=E73pXWQ2I23F7A13HnZemyX6W97sSbTD&V0=0>

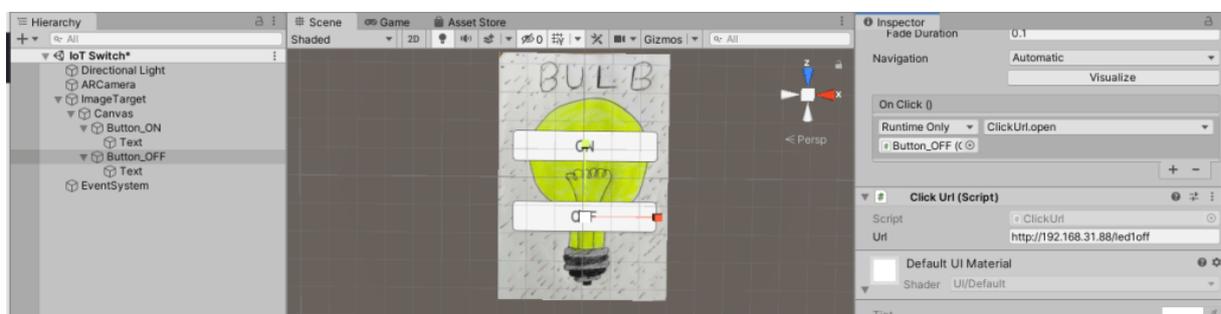
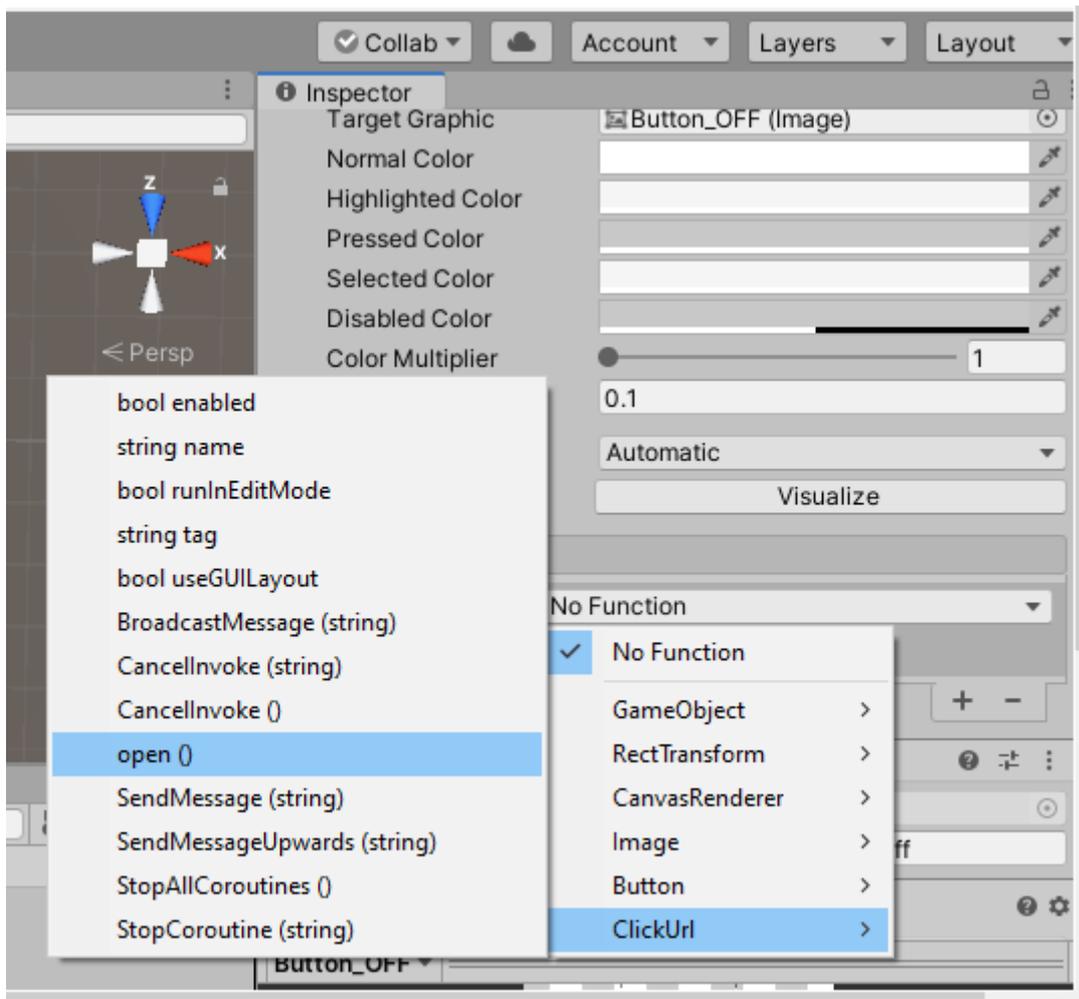
- Next we need to define the click action.
- Select any button and click on the + button under On Click():



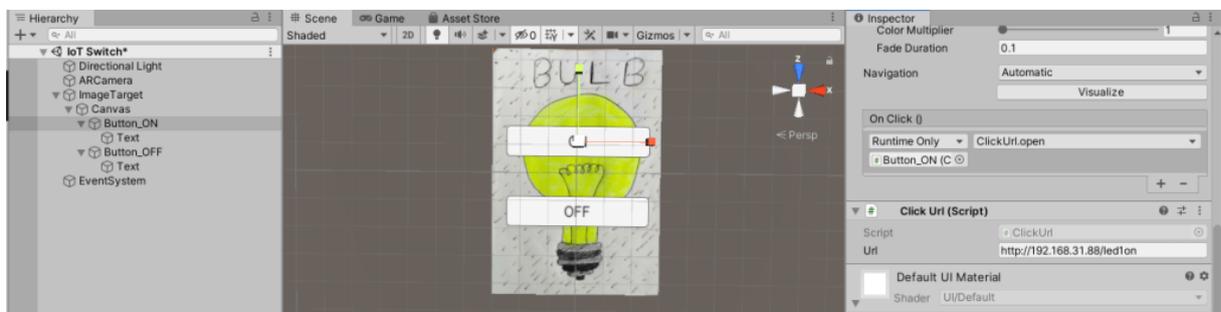
- Next drag and drop the Button_off Game Object from Hierarchy to the On Click ().



- Finally Select the **Open()** function we defined in our **ClickUrl** script as shown



Note: Repeat the above steps for the other button (**Button_ON**) as well to call **Open()** function.



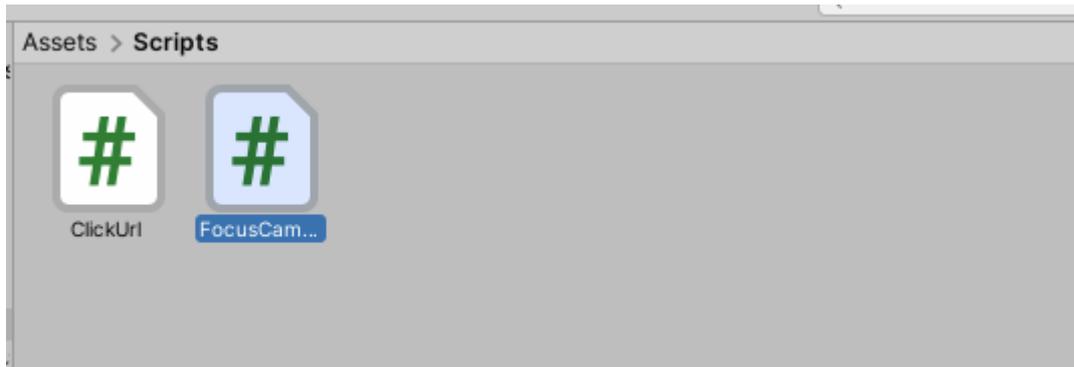
With this our Buttons are ready and functional to trigger the HTTP request.

STEP7: Add script to Autofocus AR Camera.

The progress up to STEP 6 works properly with the computer camera inside the Unity environment. But when we build apk file and **run on android the camera is unable to focus** and therefore fails to recognize the features on our target image.

To solve this problem we add a **new script to the AR camera** element so that it is able to **autofocus**.

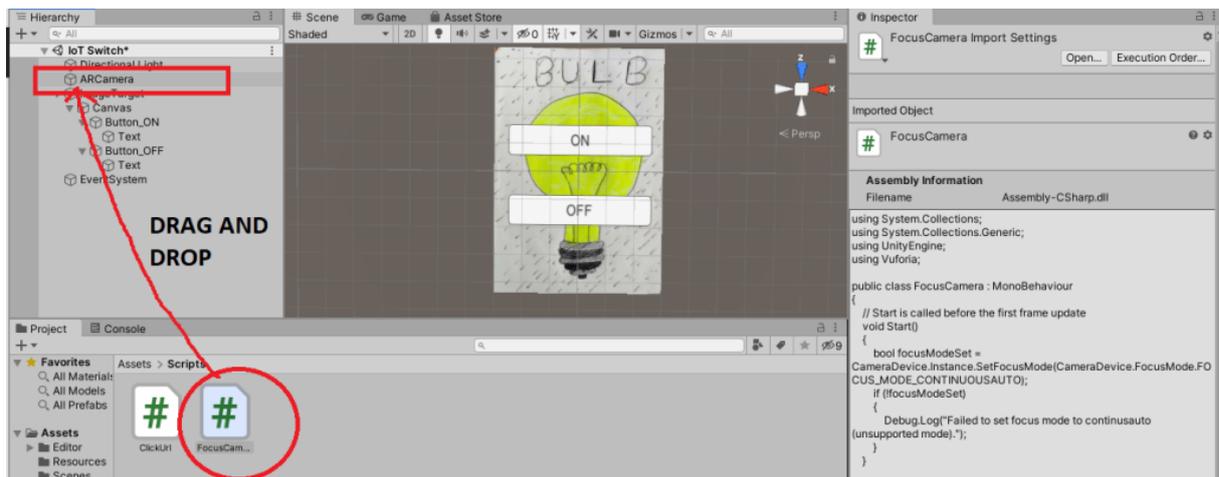
Create a new script inside **Assets>Scripts** named **“FocusCamera”**.



Add the following code to the above script.

```
1      using System.Collections;
2      using System.Collections.Generic;
3      using UnityEngine;
4      using Vuforia;
5
6      public class FocusCamera : MonoBehaviour
7      {
8          // Start is called before the first frame update
9          void Start()
10         {
11             bool focusModeSet = CameraDevice.Instance.SetFocusMode(CameraDevice.FocusMode.FOCUS_MODE_CONTINUOUS_VIDEO);
12             if (!focusModeSet)
13             {
14                 Debug.Log("Failed to set the focus mode to continuous (unsupported mode).");
15             }
16         }
17     }
18 }
```

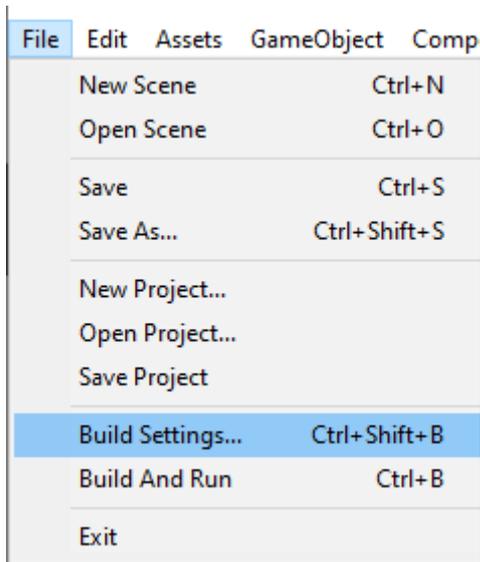
- Attach this FocusCamera Script to the AR Camera Element (Simply Drag and drop)



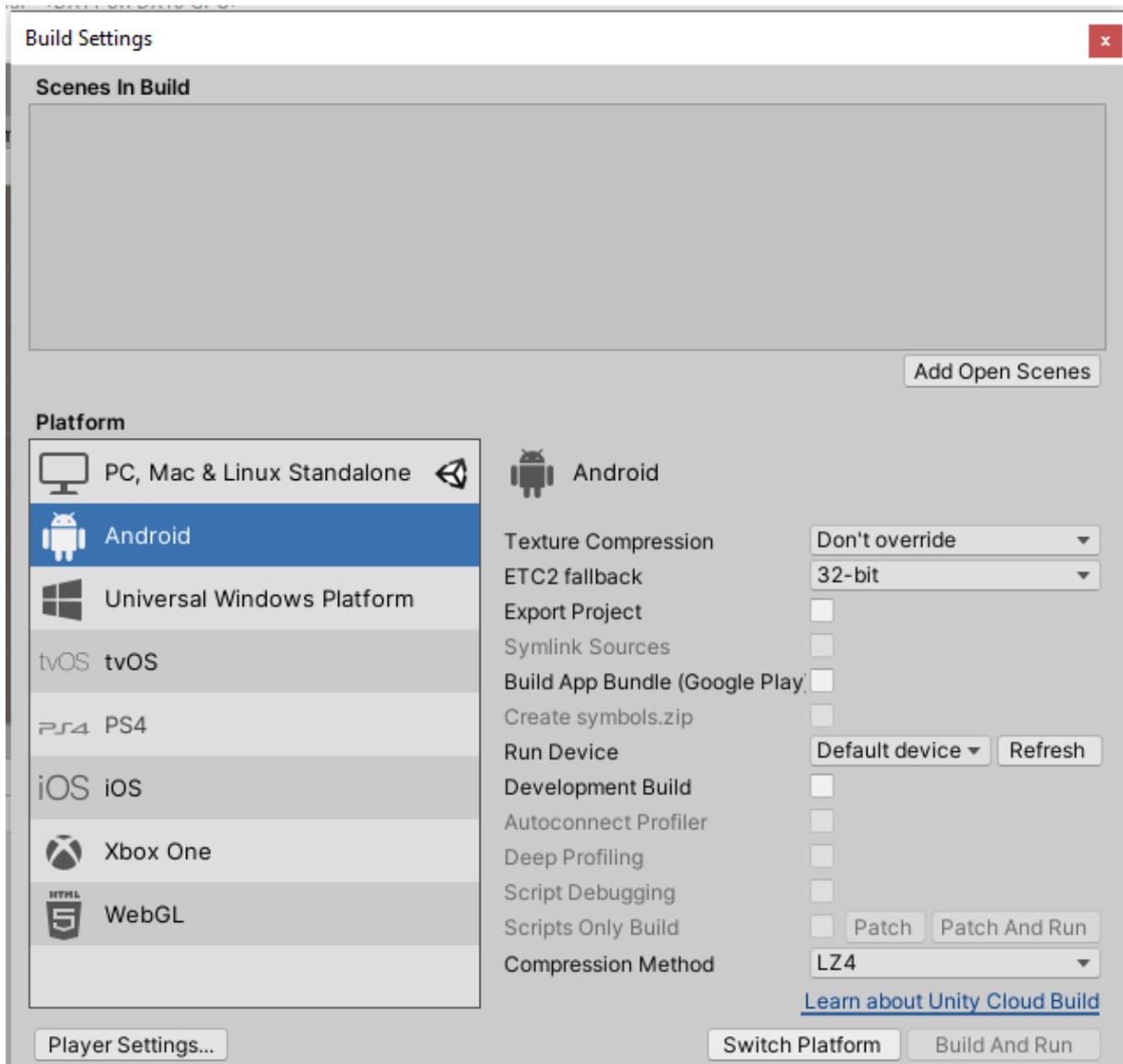
STEP8: Build a .apk file and test it on any Android Device.

Finally It is time to **build** our **android app**

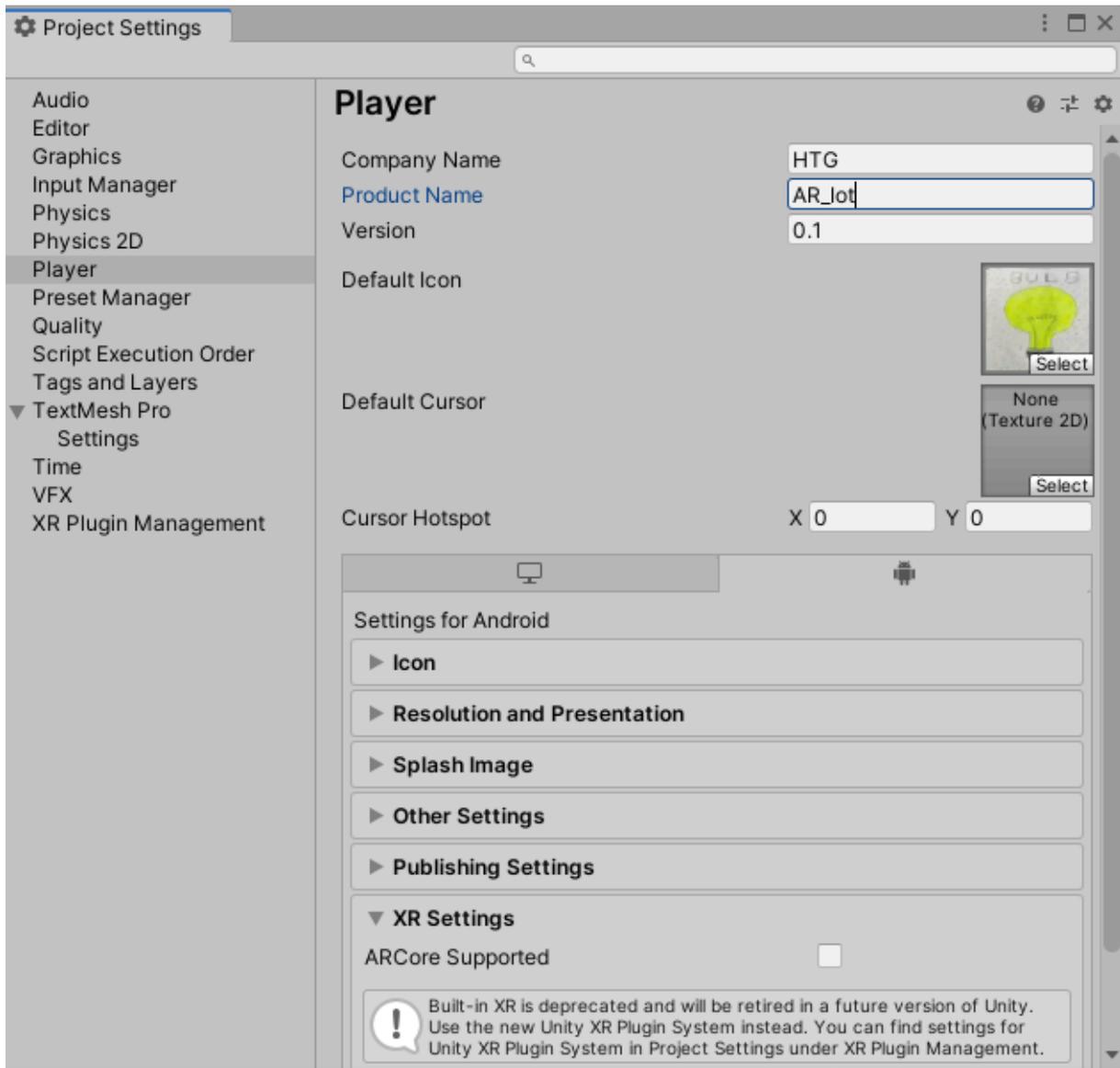
- Goto File > Build Settings



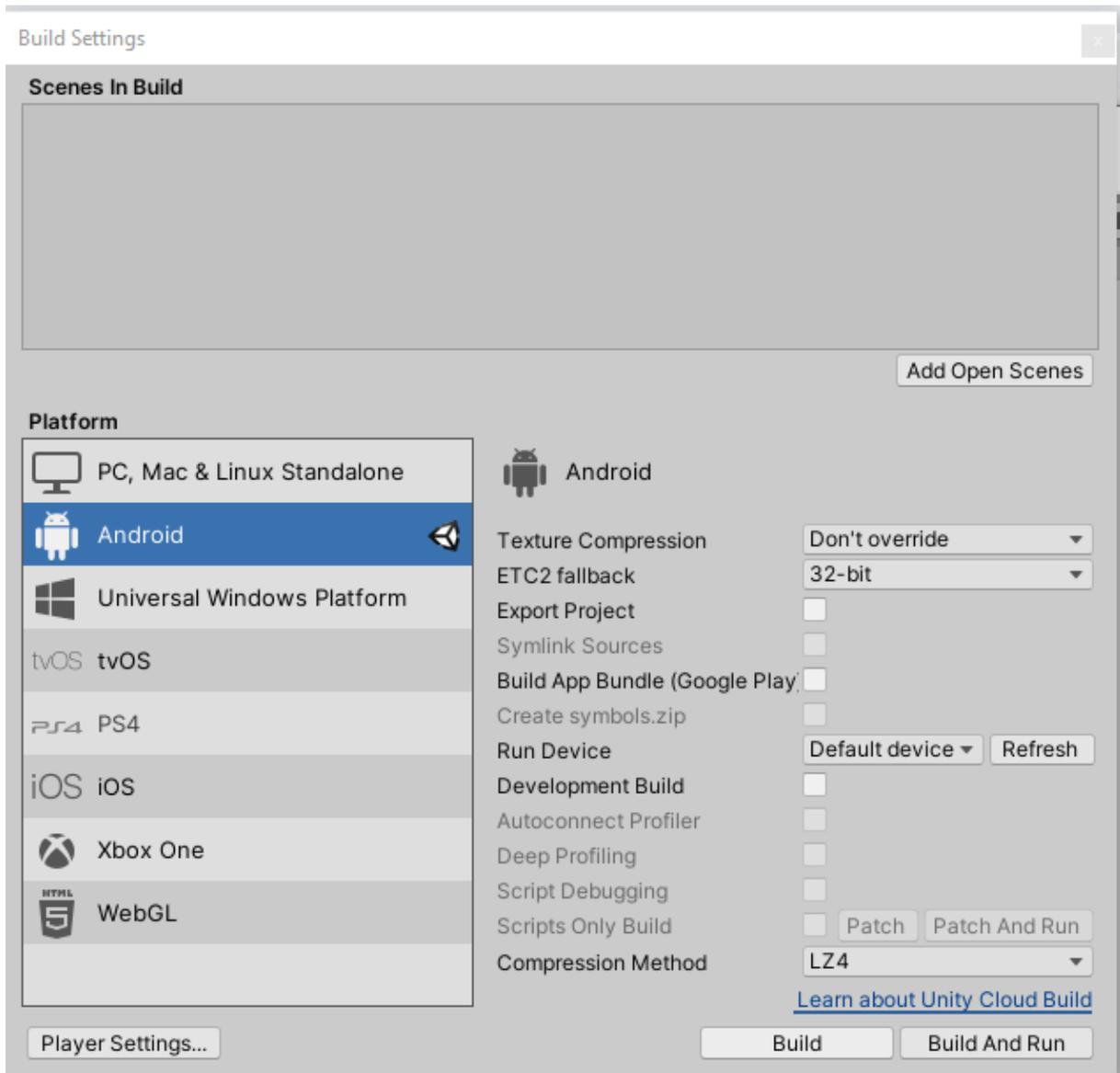
- Switch to Android. [Click on Switch Platform]



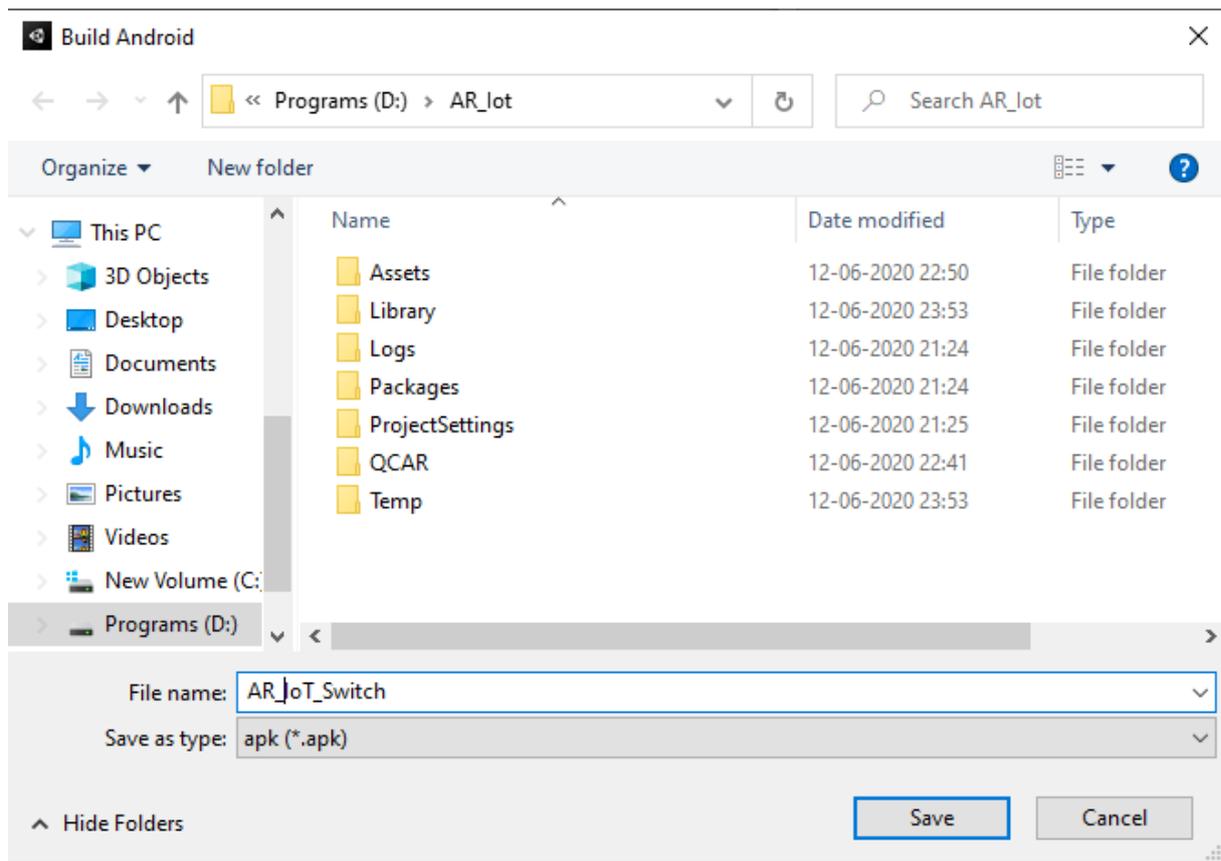
- Click on **Player settings** and add the details you want. **App icon** can also be changed from here.



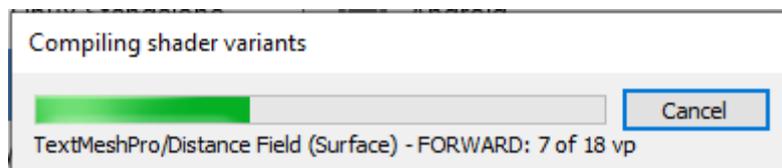
- Finally click on build.



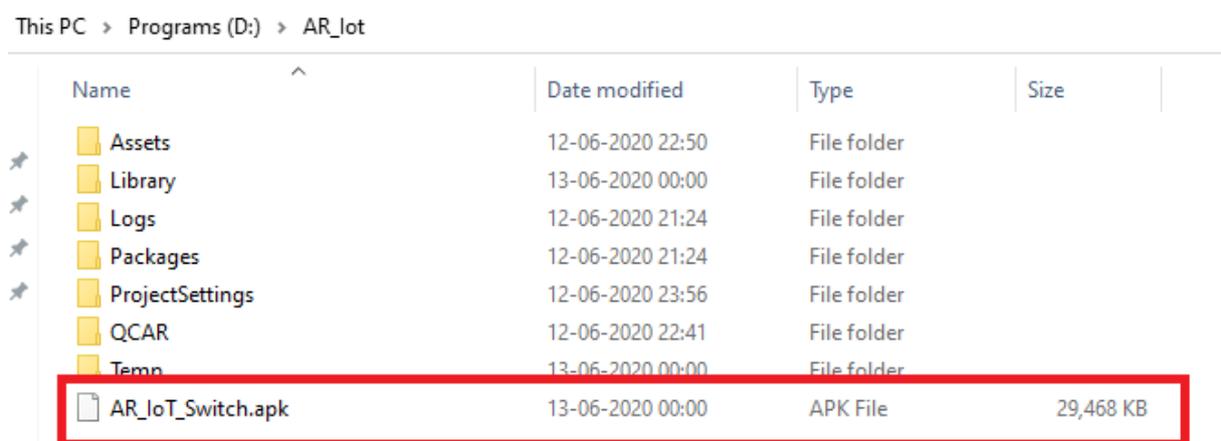
- Give a name to your Apk and save.



- It will **start compiling** and finish after some time.



- After some time you will have your **apk file**.



Transfer and install this apk file on any android device to see it in action.

That's All We need to build an AR based IoT switch.

Appendix C

(Software coding)

```

#define BLYNK_TEMPLATE_ID      "TMPLqUGcc6I7"
#define BLYNK_DEVICE_NAME     "AG IoT Switch"
#define BLYNK_AUTH_TOKEN      "E73pXWQ2I23F7A13HnZemyX6W97sSbTD"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

char auth[] = BLYNK_AUTH_TOKEN;
#define APP_DEBUG

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)
#define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Wokwi-GUEST";
char pass[] = "";

// This function will be called every time Slider Widget
// in Blynk app writes values to the Virtual Pin V1
BLYNK_WRITE(V0)
{
  if(param.asInt()==1){
    digitalWrite(25, HIGH);
    display.clearDisplay();
    display.setCursor(0,0);
    display.drawRoundRect(0, 0, 128, 64, 8, WHITE);
    display.drawRoundRect(5, 5, 118, 54, 8, WHITE);
    // Sets the color to black with a white background
    display.setTextColor(WHITE);
    display.setTextSize(1);
    display.setCursor(25,9);
    display.println("AG IoT Switch");
    display.drawLine(6,19,121,19, WHITE);
    display.setCursor(17,22);
    display.setTextSize(2);
    display.println("Relay ON");
    display.drawLine(6,40,121,40, WHITE);
    display.setCursor(33,46);
    display.setTextSize(1);
    display.println("R C C I I T");
    display.display();
  }
  else{
    digitalWrite(25, LOW); // assigning incoming value from pin V1 to a variable
    display.setTextColor(BLACK);
    display.setCursor(17,22);
    display.setTextSize(2);
    display.println("Relay ON");
    display.setTextColor(WHITE);
    display.setCursor(13,22);
    display.setTextSize(2);
    display.println("Relay OFF");
    display.display();
  }
}

```

```

    // process received value
}

void setup()
{
    // Debug console
    Serial.begin(115200);
    pinMode(25,OUTPUT);
    Blynk.begin(auth, ssid, pass);
    display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // initialize with the I2C addr 0x3D (for the 128x64)

    display.clearDisplay();
    display.setCursor(0,0);
    display.drawRoundRect(0, 0, 128, 64, 8, WHITE);
    display.drawRoundRect(5, 5, 118, 54, 8, WHITE);
    // Sets the color to black with a white background
    display.setTextColor(WHITE);
    display.setTextSize(1);
    display.setCursor(25,9);
    display.println("AG IoT Switch");
    display.drawLine(6,19,121,19, WHITE);
    //display.setCursor(23,21);
    //display.println("Please connect");
    //display.setCursor(32,31);
    //display.println("the wifi...");
    display.drawLine(6,40,121,40, WHITE);
    display.setCursor(33,46);
    display.println("R C C I I T");
    display.display();

    delay (500);
    // You can also specify server:
    //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
    //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
}

void loop()
{
    Blynk.run();
}

```

Appendix D

(Data sheets)

ESP32 Datasheet



Espressif Systems

October 8, 2016

About This Guide

This document provides introduction to the specifications of ESP32 hardware.

The document structure is as follows:

Chapter	Title	Subject
Chapter 1	Overview	An overview of ESP32, including featured solutions, basic and advanced features, applications and development support
Chapter 2	Pin Definitions	Introduction to the pin layout and descriptions
Chapter 3	Functional Description	Description of the major functional modules
Chapter 4	Peripheral Interface	Description of the peripheral interfaces integrated on ESP32
Chapter 5	Electrical Characteristics	The electrical characteristics and data of ESP32
Chapter 6	Package Information	The package details of ESP32
Chapter 7	Supported Resources	The related documents and community resources for ESP32
Appendix	Touch Sensor	The touch sensor design and layout guidelines

Release Notes

Date	Version	Release notes
2016.08	V1.0	First release

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice. THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein. The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

Copyright © 2016 Espressif Inc. All rights reserved.

Name	No.	Type	Function
SD_DATA_0	32	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SD_DATA_1	33	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
VDD3P3_CPU			
GPIO5	34	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
GPIO18	35	I/O	GPIO18, VSPICLK, HS1_DATA7
GPIO23	36	I/O	GPIO23, VSPID, HS1_STROBE
VDD3P3_CPU	37	P	CPU IO power supply input (1.8V - 3.3V)
GPIO19	38	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
GPIO22	39	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
U0RXD	40	I/O	GPIO3, U0RXD, CLK_OUT2
U0TXD	41	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
GPIO21	42	I/O	GPIO21, VSPIHD, EMAC_TX_EN
Analog			
VDDA	43	I/O	Analog power supply (2.3V - 3.6V)
XTAL_N	44	O	External crystal output
XTAL_P	45	I	External crystal input
VDDA	46	P	Digital power supply for PLL (2.3V - 3.6V)
CAP2	47	I	Connects with a 3 nF capacitor and 20 k Ω resistor in parallel to CAP1
CAP1	48	I	Connects with a 10 nF series capacitor to ground

2.3 Power Scheme

ESP32 digital pins are divided into three different power domains:

- VDD3P3_RTC
- VDD3P3_CPU
- VDD_SDIO

VDD3P3_RTC is also the input power supply for RTC and CPU. **VDD3P3_CPU** is also the input power supply for CPU.

VDD_SDIO connects to the output of an internal LDO, whose input is **VDD3P3_RTC**. When **VDD_SDIO** is connected to the same PCB net together with **VDD3P3_RTC**; the internal LDO is disabled automatically.

The internal LDO can be configured as 1.8V, or the same voltage as **VDD3P3_RTC**. It can be powered off via software to minimize the current of Flash/SRAM during the Deep-sleep mode.

Note:

It is required that the power supply of **VDD3P3_RTC**, **VDD3P3_CPU** and analog must be stable before the pin **CHIP_PU** is set at high level.

2.4 Strapping Pins

ESP32 has 6 strapping pins:

- MTDI/GPIO12: internal pull-down
- GPIO0: internal pull-up
- GPIO2: internal pull-down
- GPIO4: internal pull-down
- MTDO/GPIO15: internal pull-up
- GPIO5: internal pull-up

Software can read the value of these 6 bits from the register "GPIO_STRAPPING".

During the chip power-on reset, the latches of the strapping pins sample the voltage level as strapping bits of "0" or "1", and hold these bits until the chip is powered down or shut down. The strapping bits configure the device boot mode, the operating voltage of VDD_SDIO and other system initial settings.

Each strapping pin is connected with its internal pull-up/pull-down during the chip reset. Consequently, if a strapping pin is unconnected or the connected external circuit is high-impedance, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values, users can apply the external pull-down/pull-up resistances, or apply the host MCU's GPIOs to control the voltage level of these pins when powering on ESP32.

After reset, the strapping pins work as the normal functions pins.

Refer to Table 2 for detailed boot modes configuration by strapping pins.

Table 2: Strapping Pins

Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3V		1.8V	
MTDI	Pull-down	0		1	
Bootling Mode					
Pin	Default	SPI Boot		Download Boot	
GPIO0	Pull-up	1		0	
GPIO2	Pull-down	Don't-care		0	
Debugging Log on U0TXD During Bootling					
Pin	Default	U0TXD Toggling		U0TXD Silent	
MTDO	Pull-up	1		0	
Timing of SDIO Slave					
Pin	Default	Falling-edge Input Falling-edge Output	Falling-edge Input Rising-edge Output	Rising-edge Input Falling-edge Output	Rising-edge Input Rising-edge Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

Note:

Firmware can configure register bits to change the setting of "Voltage of Internal LDO (VDD_SDIO)" and "Timing of SDIO Slave" after bootling.

3. Functional Description

This chapter describes the functions implemented in ESP32.

3.1 CPU and Memory

3.1.1 CPU

ESP32 contains two low-power Xtensa® 32-bit LX6 microprocessors with the following features.

- 7-stage pipeline to support the clock frequency of up to 240 MHz
- 16/24-bit Instruction Set provides high code-density
- Support Floating Point Unit
- Support DSP instructions, such as 32-bit Multiplier, 32-bit Divider, and 40-bit MAC
- Support 32 interrupt vectors from about 70 interrupt sources

The dual CPUs interface through:

- Xtensa RAM/ROM Interface for instruction and data
- Xtensa Local Memory Interface for fast peripheral register access
- Interrupt with external and internal sources
- JTAG interface for debugging

3.1.2 Internal Memory

ESP32's internal memory includes:

- 448 KBytes ROM for booting and core functions
- 520 KBytes on-chip SRAM for data and instruction
- 8 KBytes SRAM in RTC, which is called RTC SLOW Memory and can be used for co-processor accessing during the Deep-sleep mode
- 8 KBytes SRAM in RTC, which is called RTC FAST Memory and can be used for data storage and main CPU during RTC Boot from the Deep-sleep mode
- 1 Kbit of EFUSE, of which 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including Flash-Encryption and Chip-ID

3.1.3 External Flash and SRAM

ESP32 supports 4 x 16 MBytes of external QSPI Flash and SRAM with hardware encryption based on AES to protect developer's programs and data.

ESP32 accesses external QSPI Flash and SRAM by the high-speed caches

- Up to 16 MBytes of external Flash are memory mapped into the CPU code space, supporting 8-bit, 16-bit and 32-bit access. Code execution is supported.

- Up to 8 MBytes of external Flash/SRAM are memory mapped into the CPU data space, supporting 8-bit, 16-bit and 32-bit access. Data read is supported on the Flash and SRAM. Data write is supported on the SRAM.

3.1.4 Memory Map

The structure of address mapping is shown in Figure 3. The memory and peripherals mapping of ESP32 is shown in Table 3.

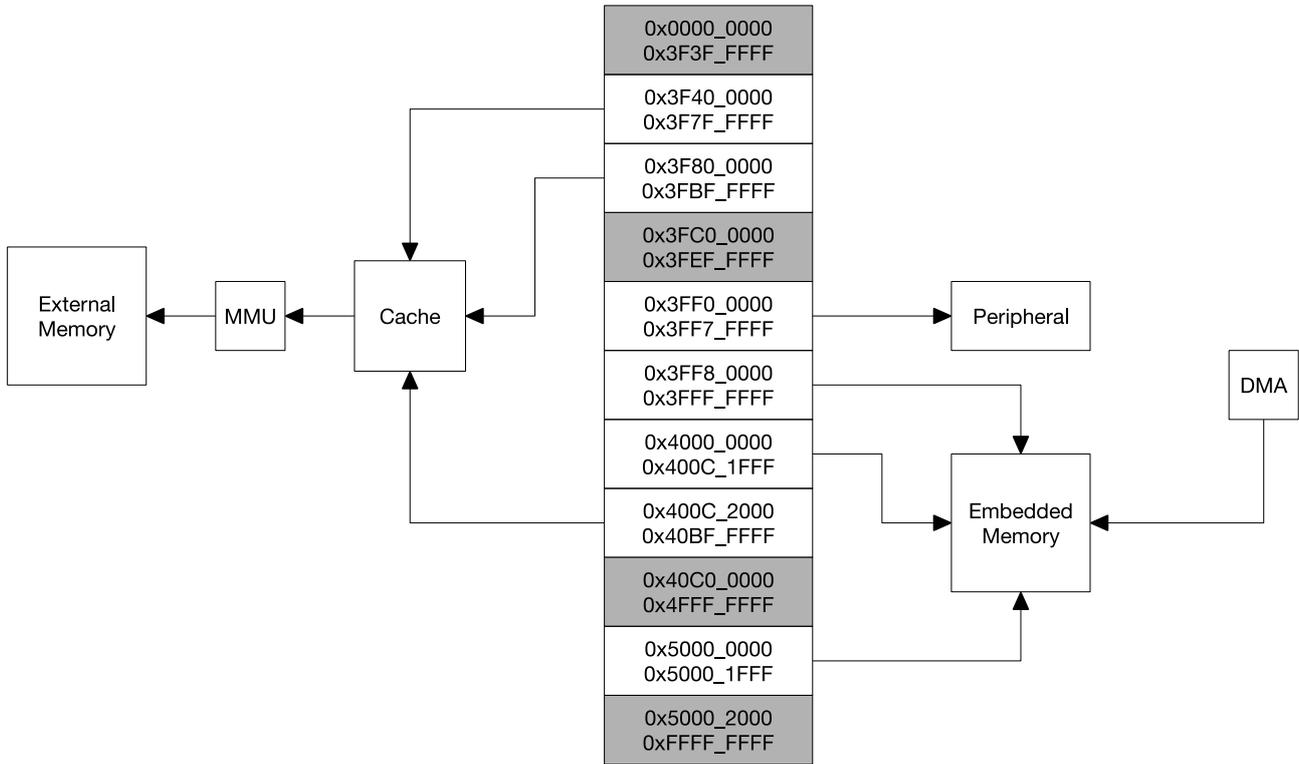


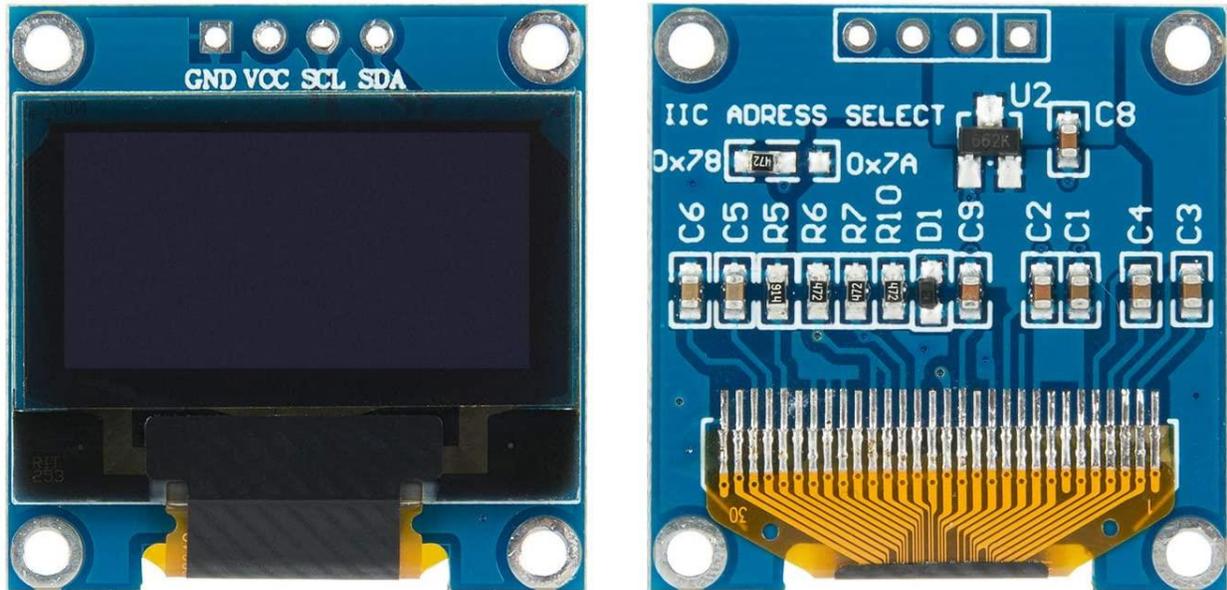
Figure 3: Address Mapping Structure

Table 3: Memory and Peripheral Mapping

Category	Target	Start Address	End Address	Size
Embedded Memory	Internal ROM 0	0x4000_0000	0x4005_FFFF	384 KB
	Internal ROM 1	0x3FF9_0000	0x3FF9_FFFF	64 KB
	Internal SRAM 0	0x4007_0000	0x4009_FFFF	192 KB
	Internal SRAM 1	0x3FFE_0000	0x3FFF_FFFF	128 KB
		0x400A_0000	0x400B_FFFF	
	Internal SRAM 2	0x3FFA_E000	0x3FFD_FFFF	200 KB
	RTC FAST Memory	0x3FF8_0000	0x3FF8_1FFF	8 KB
0x400C_0000		0x400C_1FFF		
RTC SLOW Memory	0x5000_0000	0x5000_1FFF	8 KB	
External Memory	External Flash	0x3F40_0000	0x3F7F_FFFF	4 MB
		0x400C_2000	0x40BF_FFFF	11 MB 248 KB
	External SRAM	0x3F80_0000	0x3FBF_FFFF	4 MB

Category	Target	Start Address	End Address	Size
Peripheral	DPort Register	0x3FF0_0000	0x3FF0_0FFF	4 KB
	AES Accelerator	0x3FF0_1000	0x3FF0_1FFF	4 KB
	RSA Accelerator	0x3FF0_2000	0x3FF0_2FFF	4 KB
	SHA Accelerator	0x3FF0_3000	0x3FF0_3FFF	4 KB
	Secure Boot	0x3FF0_4000	0x3FF0_4FFF	4 KB
	Cache MMU Table	0x3FF1_0000	0x3FF1_3FFF	16 KB
	PID Controller	0x3FF1_F000	0x3FF1_FFFF	4 KB
	UART0	0x3FF4_0000	0x3FF4_0FFF	4 KB
	SPI1	0x3FF4_2000	0x3FF4_2FFF	4 KB
	SPIO	0x3FF4_3000	0x3FF4_3FFF	4 KB
	GPIO	0x3FF4_4000	0x3FF4_4FFF	4 KB
	RTC	0x3FF4_8000	0x3FF4_8FFF	4 KB
	IO MUX	0x3FF4_9000	0x3FF4_9FFF	4 KB
	SDIO Slave	0x3FF4_B000	0x3FF4_BFFF	4 KB
	UDMA1	0x3FF4_C000	0x3FF4_CFFF	4 KB
	I2S0	0x3FF4_F000	0x3FF4_FFFF	4 KB
	UART1	0x3FF5_0000	0x3FF5_0FFF	4 KB
	I2C0	0x3FF5_3000	0x3FF5_3FFF	4 KB
	UDMA0	0x3FF5_4000	0x3FF5_4FFF	4 KB
	SDIO Slave	0x3FF5_5000	0x3FF5_5FFF	4 KB
	RMT	0x3FF5_6000	0x3FF5_6FFF	4 KB
	PCNT	0x3FF5_7000	0x3FF5_7FFF	4 KB
	SDIO Slave	0x3FF5_8000	0x3FF5_8FFF	4 KB
	LED PWM	0x3FF5_9000	0x3FF5_9FFF	4 KB
	Efuse Controller	0x3FF5_A000	0x3FF5_AFFF	4 KB
	Flash Encryption	0x3FF5_B000	0x3FF5_BFFF	4 KB
	PWM0	0x3FF5_E000	0x3FF5_EFFF	4 KB
	TIMG0	0x3FF5_F000	0x3FF5_FFFF	4 KB
	TIMG1	0x3FF6_0000	0x3FF6_0FFF	4 KB
	SPI2	0x3FF6_4000	0x3FF6_4FFF	4 KB
	SPI3	0x3FF6_5000	0x3FF6_5FFF	4 KB
	SYSCON	0x3FF6_6000	0x3FF6_6FFF	4 KB
	I2C1	0x3FF6_7000	0x3FF6_7FFF	4 KB
	SDMMC	0x3FF6_8000	0x3FF6_8FFF	4 KB
	EMAC	0x3FF6_9000	0x3FF6_AFFF	8 KB
	PWM1	0x3FF6_C000	0x3FF6_CFFF	4 KB
	I2S1	0x3FF6_D000	0x3FF6_DFFF	4 KB
	UART2	0x3FF6_E000	0x3FF6_EFFF	4 KB
	PWM2	0x3FF6_F000	0x3FF6_FFFF	4 KB
	PWM3	0x3FF7_0000	0x3FF7_0FFF	4 KB
	RNG	0x3FF7_5000	0x3FF7_5FFF	4 KB

OLED 4 Pin 128*64 Display Module 0.96" Blue Color



In contrast to LCD technology, Organic Light-Emitting Diode (OLED) displays do not require a backlight and are regarded as the ultimate technology for the next generation of flat-panel displays.

OLED displays are composed of a thin, multi-layered organic film placed between an anode and cathode, which are made up of electric conductive transparent Indium Tin Oxide.

The multi-layered organic film includes a Hole Transporting Layer, Emission Layer and Electron Transporting Layer.

By applying an appropriate electrical voltage, the holes and electrons are injected into the Emission Layer from the anode and cathode respectively and combine to form excitons, after which electroluminescence occurs.

This 0.96" 128*64 Blue OLED Module offers 128*64-pixel resolution. They are featuring much less thickness than LCD Displays with good brightness and produce better and true colors.

This OLED Display Module is very compact and will add a great ever user interface experience to your Arduino project. The connection of this display with Arduino is made through the I2C (also called as IIC) serial interface.

The 0.96" 4 pin 128*64 Blue OLED Display Module produces blue text on black background with very good contrast when supplied with 3.3V-5V Supply. The OLED Display Modules also offers a very wide viewing angle.

FEATURES:

- Supply voltage: 3.3V-5V
- Pixel: 128*64
- Display size- 0.96 inch

- Operating temperature range: -40°C - +80°C
- Use I2C Interface
- Chip No: SSD1306
- Color: Blue
- Drive Duty: 1/64 Duty
- Only need 2 I/O port to control
- Supported platforms: For Arduino,51 series, MSP430 series, STIM32/2, SCR chips
- Super high contrast and brightness(adjustable)
- Low power consumption
- High contrast, thus supporting clear display with no need of backlight
- For OLED SSD1306, a more elaborate and beautiful screen than LCD with more functions

PIN DESCRIPTION:

Pin No.	Pin Name	Description
1.	Supply Voltage (Vcc, 5V)	Can be powered by either 3.3V or 5V
2.	Ground (GND)	Pin Ground
3.	Serial Clock(SCL)	Pin SCL of I2C interface
4.	Serial Data(SDA)	Pin SDA of I2C interface

MECHANICAL SPECIFICATIONS:

ITEM	NORMAL DIMENSION
Module Dimension	27.30*27.30*2.37
Active Area	21.74*10.86
Pixel Size	0.148*0.148
Pixel Pitch	0.17*0.17

ABSOLUTE MAXIMUM RATING:

PARAMETER	SYMBOL	MIN	MAX	UNIT
Supply voltage for logic	VCC	1.65	5.5	V
Operating temperature	TOP	-40	+80	°C
Storage temperature	TSTG	-40	+80	°C

ELECTRONICAL CHARACTERISTICS:

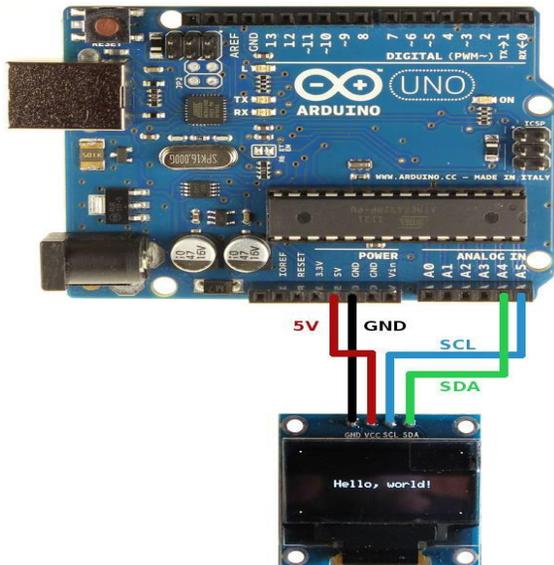
ITEM	SYMBOL	CONDITION	MIN	TYP	MAX	UNIT
Supply voltage for logic	VCC	-----	2.8	3.3	5.2	V
Input high voltage	VIH	-----	0.8*VCC	-----	VCC	V
Input low voltage	VIL	-----	0	-----	0.2*VCC	V
Output high voltage	VOH	-----	0.9*VCC	-----	VCC	V
Output low voltage	VOL	-----	0	-----	0.1*VCC	V
50%check board operating current	ICC	VCC=3.3	-----	12.0	20.0	mA

CONNECTION DIAGRAM OF ARDUINO UNO TO 4 PIN 0.96 INCH I2C OLED DISPLAY TO ARDUINO UNO:

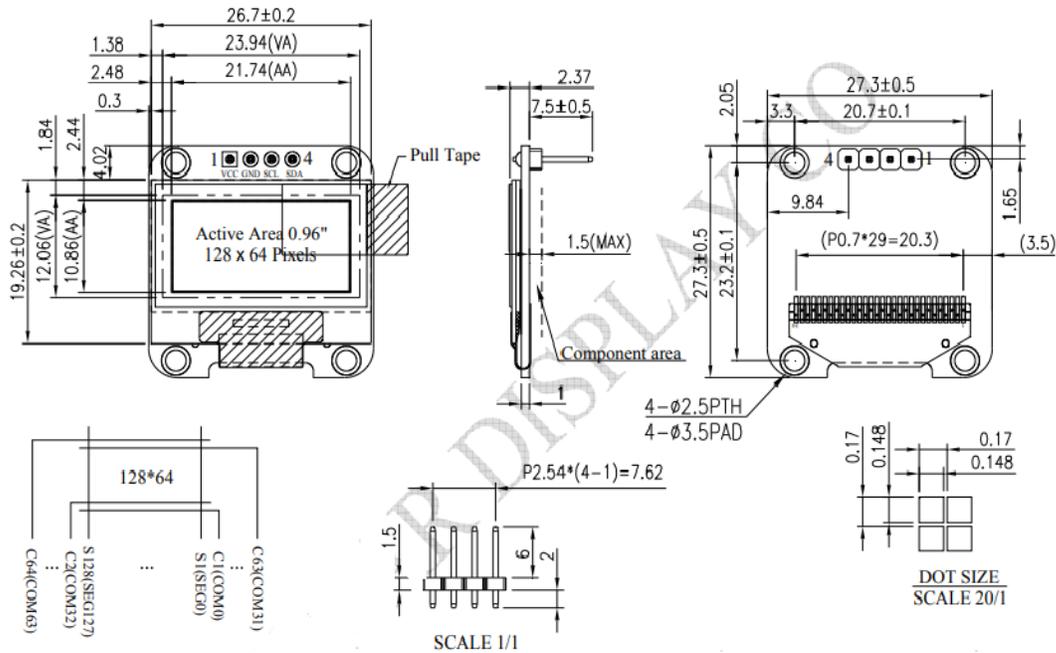
Arduino Uno OLED Wiring

The image below shows how to connect the 0.96inch OLED I2C display to Arduino. Pin connections are as follows for wiring the OLED display to an Arduino Uno.

- OLED GND – Arduino GND
- OLED VCC – Arduino 5V
- OLED SCL – Arduino Uno A5
- OLED SDA – Arduino Uno A4



OUTER DIMENSION:



APPLICATIONS:

Due to its capability in displaying, it is often used in various application for instances, smart watch, MP3, function cellphone, portable health device and many others.