International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-11 Issue-1, November, 2021

Augmented Reality based Mobile Application for Energy Monitoring and IoT Device Control

Abishek R, D. Vaishali, Adhitya Narayan R, Vignesh Sundar M

Abstract: ***IoT has become an integrated part of our lives changing ways in which we 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. We 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 we at all remember which App controls which appliance. We intend to solve these two major inconveniences by creating a single mobile application that can control all these appliances using Augmented Reality technology. All we have to do is point our camera at the appliance that we need to operate and the App will display control options in real-time AR. This paper produces five important contributions:***

1) An AR-based mobile application to control IoT devices and monitor the environment.

2) Implementing the mobile application using Unity 3D engine and Vuforia SDK.

3) Integrating a commercially available IoT device with the mobile application.

4) Integrating custom-made hardware IoT device with mobile application.

5) Integrating this combination to make our industries and homes smarter

Keywords: ***Augmented reality, Vuforia SDK, unity engine, AR with IOT, Home automation, Internet of things***

I. INTRODUCTION

The integration of augmented reality into our work and leisure life is increasing exponentially each passing day with companies pouring millions into their R&D to increase their foothold in the technology. The purpose of any technology is to reduce the hardware devices needed to perform a task and there is no better way to do it that augmented reality.

The transition from smartphones to instrumented glasses carries the same inevitable nature that the transition of smartphones
from computers had.

To lay down the purpose of augmented reality in much simpler words, it is a filter through which we will be seeing the natural world with inputs and data from the digital world laid on top of it.

It is the fastest growing mass medium after smartphones and all major industries are placing their big bets on it. It is more than evident that augmented reality is going to transform the way life and work as we know it in a good way.

In this project we have built a 3D application that can be used across various platforms using Unity AR engine. Therefore it helps us control and understand our devices with more efficiency by using augmented reality.

II. PROPOSED SYSTEM

The system consists of three major parts: hardware, software and Augmented reality. The hardware part of the system consists of a commercial IOT device and a custom IOT device.

The software part incorporates the mobile application, programming of microcontroller, APIs and IOT server. Finally the augmented reality part consists of the whole system design which was made using Vuforia SDK and unity editor.

A. Hardware

1. Commercial IOT device

A Commercial IOT device from any brand or any type will be suitable for this project given that the device supports IFTTT integration.

For this project we have used a generic brand IOT smart switch which is capable of controlling the on/off controls of the appliances and also monitors the current, wattage and watt-hour readings. The device requires a power supply of 90V – 250V AC and a maximum load of 10A can be connected to it. It works on 2.4GHz IEEE 802.11b/g/n Wi-Fi standard.

The device comes with its own dedicated APP to control the appliances and also to monitor the electricity usage. The device can be integrated with google assistant, amazon Alexa and also IFTTT. IFTTT will be used to integrate the IOT device's APP with the AR mobile application.

Manuscript received on November 13, 2021.
Revised Manuscript received on November 20, 2021.
Manuscript published on November 30, 2021.

*****Correspondence Author

Abishek R*, B.Tech. Student, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai (Tamil Nadu), India.
E-mail: ar6857@srmist.edu.in

Dr. D. Vaishali, Assistant Professor, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai (Tamil Nadu), India.
E-mail: vaishalb@srmist.edu.in

Adhitya Narayan R, B.Tech. Student, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai (Tamil Nadu), India.
E-mail: rn2067@srmist.edu.in

Vignesh Sundar M, B.Tech. Student, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai (Tamil Nadu), India.
E-mail: vs6312@srmist.edu.in

Fig. 1. Commercial IOT device

The left side terminals of the IOT smart switch are the inputs which are connected to the power supply and the right side terminals are the output which are connected to the appliance we want to control.

Fig. 2. Circuit diagram of commercial IOT device

2. Custom IOT device

The custom IOT hardware is made using ESP8266 SOC microcontroller along with a relay module and an ACS712 current sensor.

We have used the NODEMCU AMICA V1.0 development board which is based on the ESP8266-12E SOC microcontroller. The SOC contains a Tensilica 32-bit RISC Xtensa LX106 CPU which is clocked at 80MHz. It has 128KB RAM and 4MB of flash memory. The SOC also contains an inbuilt 2.4GHz IEEE 802.11b/g/n Wi-Fi chip with an onboard PCB antenna. It also features a Silicon Labs CP2102 UART chip for USB-TTL communication. It comes with different open-source firmwares so that it can be programmed in C using Arduino IDE or in Python using MicroPython. It has one analog pin and 11 digital pins which act as GPIOs. In addition to this it also supports SPI, UART and I²C serial communication protocols to interface various sensors and microcontrollers. In this project it is used to control the relay module by getting the instructions from the internet through Wi-Fi and also to send the ACS712 sensor readings to the IOT cloud server.

Fig. 3. NODEMCU AMICA V1.0 Development Board

We have used a two-channel relay module with an optocoupler to control the appliances. The relay module requires a 5V DC power supply in order to operate. Apart from the relay, the module also contains an optocoupler, flyback diode, some resistors and a NPN transistor. Here the transistor acts as a switch since the relay requires 5V DC supply to trigger and not all microcontrollers work on 5V logic. ESP8266 SOC works on 3.3V logic. So, the input of the relay module will be given to the base of the transistor, the power supply is connected to the collector pin and relay input is connected to the emitter pin. When the microcontroller signals the transistor, the base gets saturated and the current flows from collector to emitter triggering the relay. The relay contains a coil which is inductive in nature. When the relay is switched off a large flyback voltage is developed which will damage the microcontroller. In order to prevent it a flyback diode is connected parallel to the relay to restrict the flow of flyback voltage. Both AC and DC appliances can be controlled with the relay module. It supports a maximum of 250V AC and 30V DC with a load of 10A max.

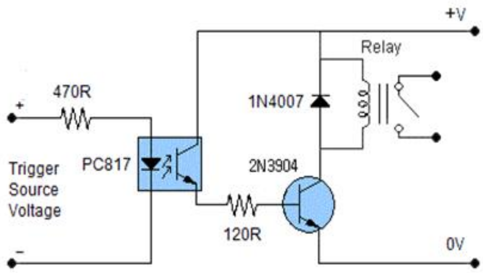

Fig. 4.Circuit diagram of relay module

Fig. 5. Relay module

The ACS712 sensor is available in three versions based on current ratings - 5A, 20A and 30A. The one used in the project is the 20A variant. The sensor works on the principle of Hall Effect. It is capable of measuring both AC and DC currents accurately. It gives analog voltage in the range of 0V - 5V as the output based on the amount of current passing through it.

The sensor comes with factory calibration and there is no hardware calibration present. Any error in readings can be rectified by using offsets in the programming. The output sensitivity is 100 mV/A for the 20A version. It requires a 5V DC power supply to operate. The sensor is connected in series with the appliance and relay in order to measure the current and the output of the sensor is connected to the A0 pin of ESP8266 development board.

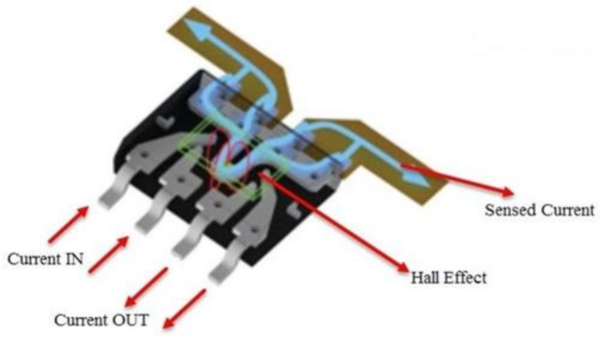

Fig. 6.Hall Effect in ACS712 Sensor

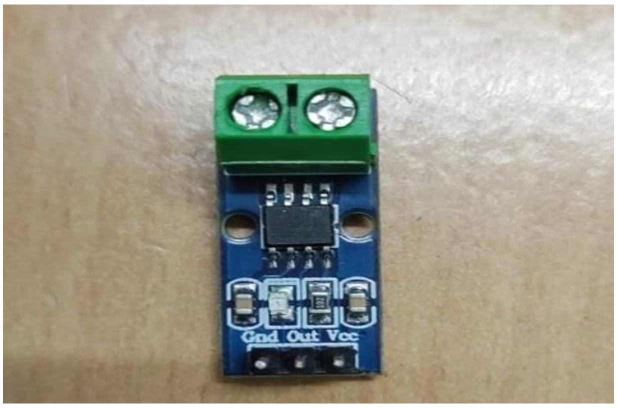

Fig. 7.ACS712 Sensor module

Both ESP8266 development board and ACS712 sensor require a 5V DC power supply for operation, therefore it is connected to an external battery power source of 5V. The sensor output pin connected to the A0 pin of the microcontroller. The ACS712 sensor module is connected in series with the appliance and relay in order to measure the current. The live wire of the 240V AC power supply is connected to the common terminal of the relay and the neutral wire is connected to the appliance.

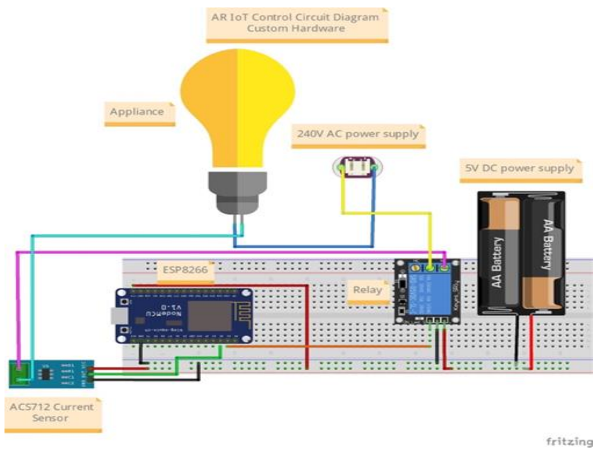

Fig. 8. Circuit diagram of custom IOT device

B. Software

1. Custom IOT device

The custom IOT hardware runs on the Blynk IOT cloud platform. The reason to go for Blynk is that Blynk provides a database to store the sensor readings so that no separate SQL database is required. Blynk provides a library for the Arduino IDE which makes it easier to control the relays and also to push the sensor readings to the cloud. It provides many HTTP APIs which can be used to get the sensor readings and also to change the state of the relay by simply calling them. The HTTP APIs are used in the augmented reality part which makes the integration of AR and IOT much easier. For this project we have used one Blynk button for relay control and two Blynk gauges to get the wattage and current readings. The microcontroller and the Blynk server are connected through an unique authentication token which also serves as a means to identify the device. The whole setup is done in the Blynk mobile application and the APIs are created in the Blynk web interface.

Fig. 9.Blynk setup

2. Commercial IOT device

Commercial IOT device uses a combination of IFTTT service along with Blynk and the device's own application and software. The problem with these commercial products is that we cannot control the device with any third party applications other than the device's own mobile application. Same is applicable for the current and wattage reading also. This is where IFTTT Services comes into play. It's an integration platform where two or more services can be integrated together. To turn on and off the appliance we have integrated the device's mobile application with webhooks. Webhooks is a IFTTT service similar to APIs. Whenever the webhooks URL is requested it will execute some tasks that have been integrated with it. For the wattage and current readings we have integrated three services. The mobile application of the device is integrated with the Blynk server, So that the data can be forwarded and stored in it. And the Blynk server is integrated with the webhooks service, So that the sensor readings can be requested by using the webhooks URL. The IFTTT platform has a simple drag and drop user interface, therefore eliminating the need for coding.

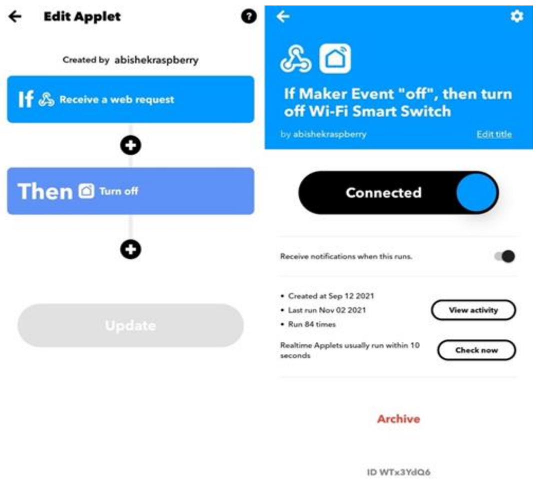

Fig. 10. Integration of webhooks and device's mobile application in IFTTT

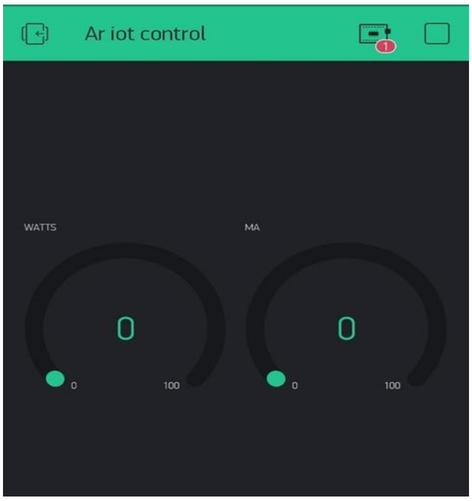

Fig. 11. Integration of webhooks and Blynk

3. AR Mobile application

The mobile application was developed after the AR application was designed using unity editor. We have used Vuforia engine SDK along with Android SDK to develop the android mobile application. One of the useful features of Vuforia engine is that it automatically generates the .APK file using the android SDK after the AR application is designed in the unity editor. This eliminates the coding required for the mobile application. There is no UI in the mobile application, it opens the camera directly which contains a Vuforia watermark.

Fig. 12.AR mobile application loading and main screen

C. AR application

1. Target image selection

It is very important to obtain proper image targets so that they can be easily recognized by the camera and the virtual information can be properly overlapped around the target image. Other important parameters are that the target image should not contain any repetitive patterns and should have been taken in good light conditions. It is also important that the image target is placed in a well lit room condition so that the camera can easily recognize it. The Vuforia web application has a target image rating facility which rates the quality of the target image in a 5 star point system. Target images getting 5 stars are easily recognized by the camera and for the target images getting less stars it will be much more difficult for the camera to recognize. For this project we have used 4 target images two for the commercial IOT device and the other two for the custom IOT device. In this one target image will be used to switch on and off the appliance and the second target image will be used to display the current and wattage readings. All the four target images are awarded 5 star rating in Vuforia so that it will be much easier for the camera to recognize it.

Fig. 13. The target images used in the project

2. Training

We have to upload target images in the Vuforia target manager for the mobile applications to recognize them. Vuforia target manager is a web application which assists in creating target databases which contain feature points of target images that can be used to identify them by detection algorithms. Vuforia provides an option to use either a local database or a cloud database. In the local database, the database is stored within the mobile application, which identifies the target images much faster. A local database is only suitable to handle fewer target images. In the cloud database, the target images will be stored in the Vuforia Cloud Recognition service. This method will be suitable if the target images are more than 100. In this project, we have used a local database as we are identifying only 4 target images.

Retrieval Number: 100.1/ijitee.A95981111121
DOI: 10.35940/ijitee.A9598.1111121
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-11 Issue-1, November, 2021

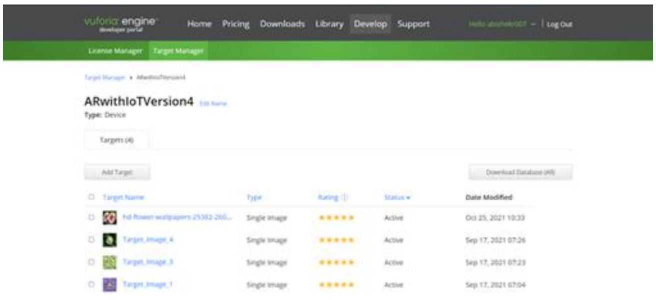

Fig. 14.Vuforia target manager

3. Application workflow

We have used three game objects in unity editor. Touch button game object is used to control the appliance using the touch screen of the smartphone. Virtual button game object is used to control the appliance in real-time AR. Finally the input field game object is used to display the current and wattage readings in real-time AR. All the game objects has been integrated with the APIs with the help of some C# scripts. Whenever the game objects are pressed the corresponding APIs will be called and the tasks it has been assigned to will be carried out. We made the application to track two target images simultaneously so that the user can control the appliances as well as see the current and wattage readings in real-time AR.

Fig. 15.AR application design in unity editor

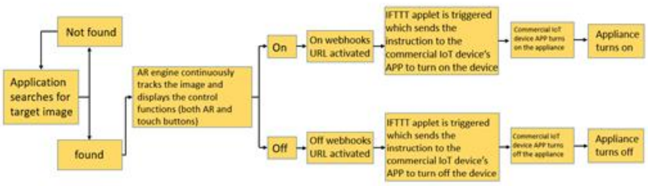

Fig. 16.Appliance control workflow for commercial IOT device

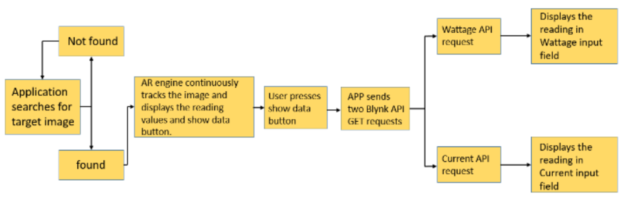

Fig. 17.Sensor reading workflow for custom IOT device

III. RESULT AND DISCUSSION

The project is successfully implemented with the features to control both commercial and custom IOT devices simultaneously along with displaying current and wattage readings of the appliances in real-time AR. We experienced a delay of around 2 to 3 seconds after the API execution. This is due to the mobile application dependence on many third party services that have been integrated with the project. The android application consumed a lot of smartphone's resources while using it. This is because we are using the local database for the target images and also the application is tracking two target images simultaneously. To get a clear idea we tested the application in two different android smartphones sporting a medium performance SOC and a high performance SOC to understand how the application fared in real testing conditions. We have one mid performance and one high performance smartphone for testing. For the mid performance smartphone we have used Motorola one fusion plus sporting a snapdragon 730G SOC and running android 10. For the high performance smartphone we have used the OnePlus 6T sporting a snapdragon 845 SOC and running android 10. The mobile application was tested in a standard environment where some of the resources were taken by OS and background running Apps. We were able to monitor individual CPU cores, SOC temperature, and RAM usage. We weren't able to monitor GPU usage as it was blocked by the OEM.

Fig. 18.Virtual information displayed after successful recognition of target images

Table- I: Performance of 845 SOC

Parameters	Percentage of Resources used
CPU core 1	47.5% of 2.8 GHz
CPU core 2	47.5% of 2.8 GHz
CPU core 3	47.5% of 2.8 GHz
CPU core 4	47.5% of 2.8 GHz
CPU core 5	45.8% of 1.8 GHz
CPU core 6	45.8% of 1.8 GHz
CPU core 7	45.8% of 1.8 GHz
CPU core 8	45.8% of 1.8 GHz
GPU	OEM Blocked
RAM	79.1% of 6Gb
SOC Temperature	56°C

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Table- II: Performance of 730G SOC

Parameters	Percentage of Resources used
CPU core 1	100% of 1.8 GHz
CPU core 2	100% of 1.8 GHz
CPU core 3	100% of 1.8 GHz
CPU core 4	100% of 1.8 GHz
CPU core 5	100% of 1.8 GHz
CPU core 6	100% of 1.8 GHz
CPU core 7	100% of 2.2 GHz
CPU core 8	100% of 2.2 GHz
GPU	OEM Blocked
RAM	66.83% of 6Gb
SOC Temperature	65°C

Snapdragon 730G SOC struggled quite a bit as the application was using 100 percent of the CPU. We were able to observe reduced frame rates and also the virtual information did not properly overlap with the target images. Snapdragon 845 SOC performed much better in our testing. We were able to achieve 60 frames per second and the virtual information also overlapped properly with the target images. Image tracking performance was good in both the smartphones.

IV. CONCLUSION

In this paper we seen two technologies that are spearheading the next industrial revolution work hand in hand. We have made possible to integrate existing IoT devices that are commercially available in the market and custom made IoT devices with augmented reality technology. In addition to this we have also seen how to monitor the environment as well as necessary data using suitable sensors. These advances when implemented brings our digital and physical worlds further together thereby fulfilling the purpose of any digital advancement.

REFERENCES

1. Kim, S. L., Suk, H. J., Kang, J. H., Jung, J. M., Laine, T. H., & Westlin, J. (2014, March). Using Unity 3D to facilitate mobile augmented reality game development. In 2014 IEEE World Forum on Internet of Things (WF-IoT) (pp. 21-26). IEEE.
2. Pokric, B., Krco, S., Drajic, D., Pokric, M., Rajs, V., Mihajlovic, Z., & Jovanovic, D. (2015). Augmented Reality Enabled IoT Services for Environmental Monitoring Utilising Serious Gaming Concept. JoWUA, 6(1), 37-55.
3. Grahn, I. (2017). The Vuforia SDK and Unity3D Game Engine: Evaluating Performance on Android Devices.
4. Shinde, Gitanjali Rahul, et al. "Augmented Reality and IoT." Internet of Things Integrated Augmented Reality. Springer, Singapore, 2021. 55-71.
5. Jo, Dongsik, and Gerard Jounghyun Kim. "ARIoT: scalable augmented reality framework for interacting with Internet of Things appliances everywhere." IEEE Transactions on Consumer Electronics 62.3 (2016): 334-340.
6. Lobo, Nestor. "Intelli-mirror: An augmented reality based IoT system for clothing and accessory display." 2016 International Conference on Internet of Things and Applications (IOTA). IEEE, 2016.
7. Sun, Yongbin, et al. "Magichand: Interact with iot devices in augmented reality environment." 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE, 2019.
8. Fraga-Lamas, Paula, et al. "A review on industrial augmented reality systems for the industry 4.0 shipyard." Ieee Access 6 (2018): 13358-13375.
9. De Pace, Francesco, Federico Manuri, and Andrea Sanna. "Augmented reality in industry 4.0." American Journal of Computer Science and Information Technology 6.1 (2018): 17.
10. Vermesan, Ovidiu, and Peter Friess, eds. Internet of things: converging technologies for smart environments and integrated ecosystems. River publishers, 2013.
11. Electronics hub, “Interfacing ACS712 Current Sensor with Arduino – Measure Current with Arduino” [online] Available: https://www.electronicshub.org/interfacing-acs712-current-sensor-with-arduino/
12. Circuits Diy, “Optocoupler Relay Driver with PC817 & 2N3904” [online] Available: https://circuits-diy.com/optocoupler-relay-driver-with-pc817-2n3904/

AUTHORS PROFILE

Abishek R, B.Tech. Student, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai, Tamil Nadu, India.

Dr. D. Vaishali, Assistant Professor, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai, Tamil Nadu, India.

Adhitya Narayan R, B.Tech. Student, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai, Tamil Nadu, India.

Vignesh Sundar M, B.Tech. Student, Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai, Tamil Nadu, India.