Controlling IOT Devices with Augmented Reality

Summer Training - 6th Semester Project

Vivek Saini
2K18CSUN01047
B.Tech CSE 7B

Pulkit Chopra
2K18CSUN01124
B.Tech CSE 7B

Introduction

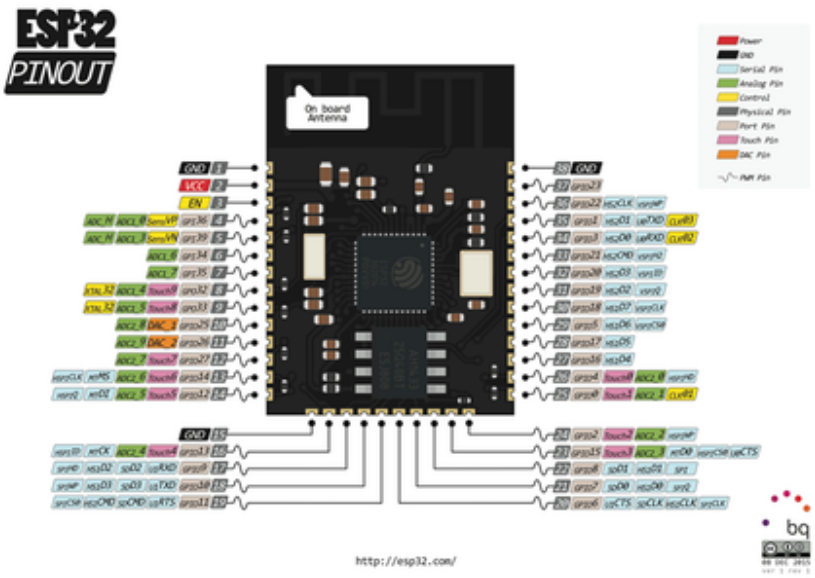
PROBLEM : Imagine you work in a thermal power plant and there is an emergency where a valve of heating system has broken. you are the in-charge of repairing that valve but due to high heat you are facing problems. If your company was using the idea that i am going to discuss next, this task would be very simple to perform.

SOLUTION : We've created an IoT and augmented reality system that can be used to operate devices without requiring physical interaction.To create a prototype of our concept, we used basic and open source platforms. The prototype includes two gadgets that are meant to showcase the notion of utilising an AR interface to operate and obtain data from an IoT device.

Requirements

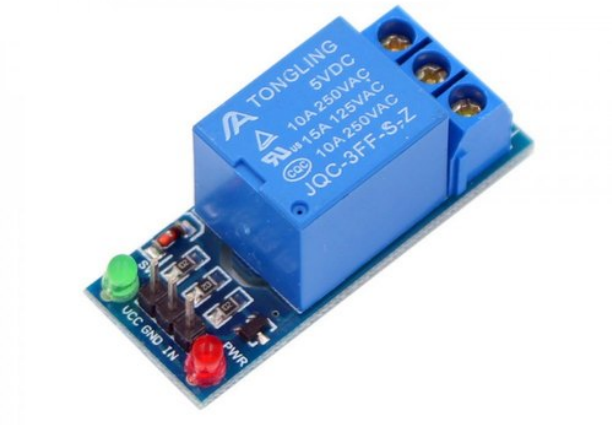
1. Hardware Components

1.1 ESP-32



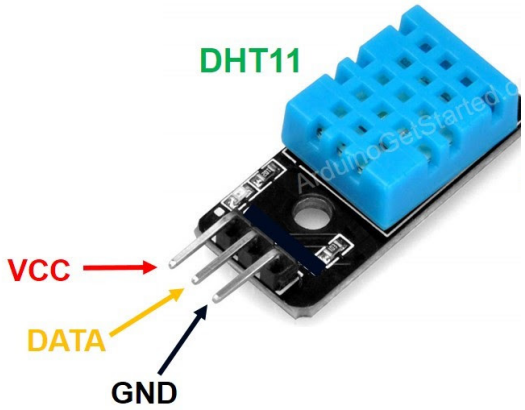
ESP32 is a low-power system on a chip (SoC) series with Wi-Fi & dual-mode bluetooth capabilities. It is highly integrated with built-in antenna switches, RF, power amplifier, low-noise receiver, amplifier, filters, and power management module.

1.2 Relay Module



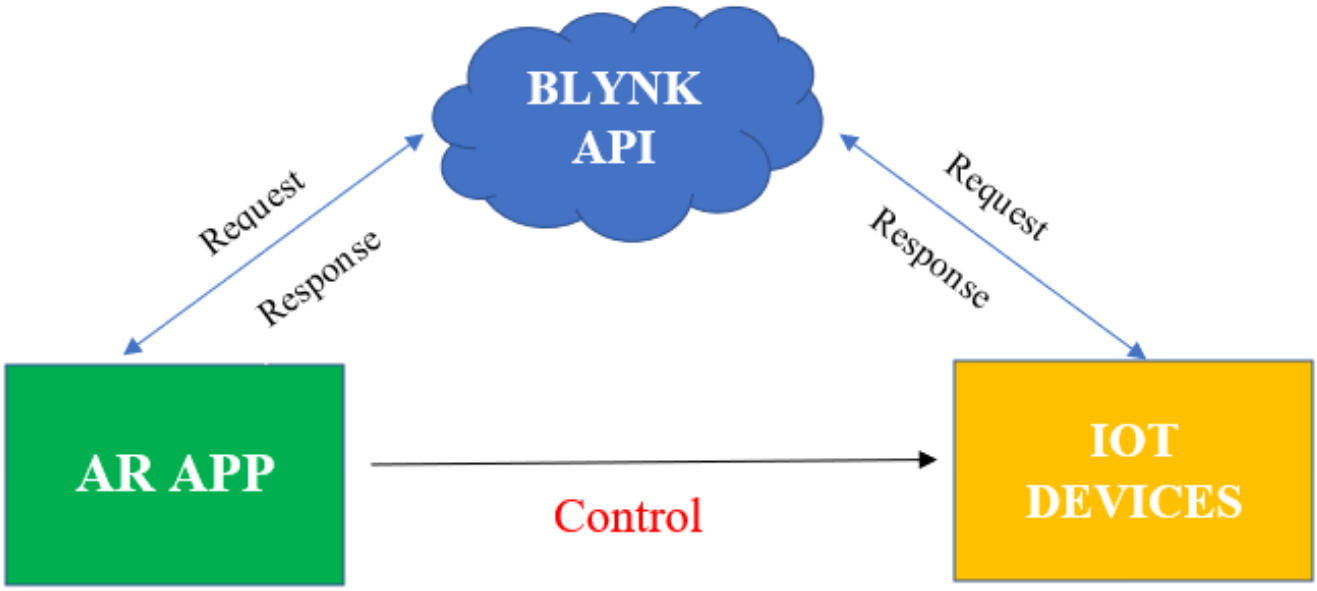
A power relay module is an electrical switch that is operated by an electromagnet. The electromagnet is activated by a separate low-power signal from a microcontroller. When activated, the electromagnet pulls to either open or close an electric switch.

1.3 DHT-11



The DHT-11 is a basic digital temperature and humidity sensor. It uses a capacitative humidity and a thermistor to measure the surrounding air and an 8-bit microcontroller to output the value of temperature and humidity as sensed data.

Flow Chart & Working



The AR application uses the Blynk platform's API to send and receive data from IoT devices by generating a unique token. This token is used to connect ESP module with the app.

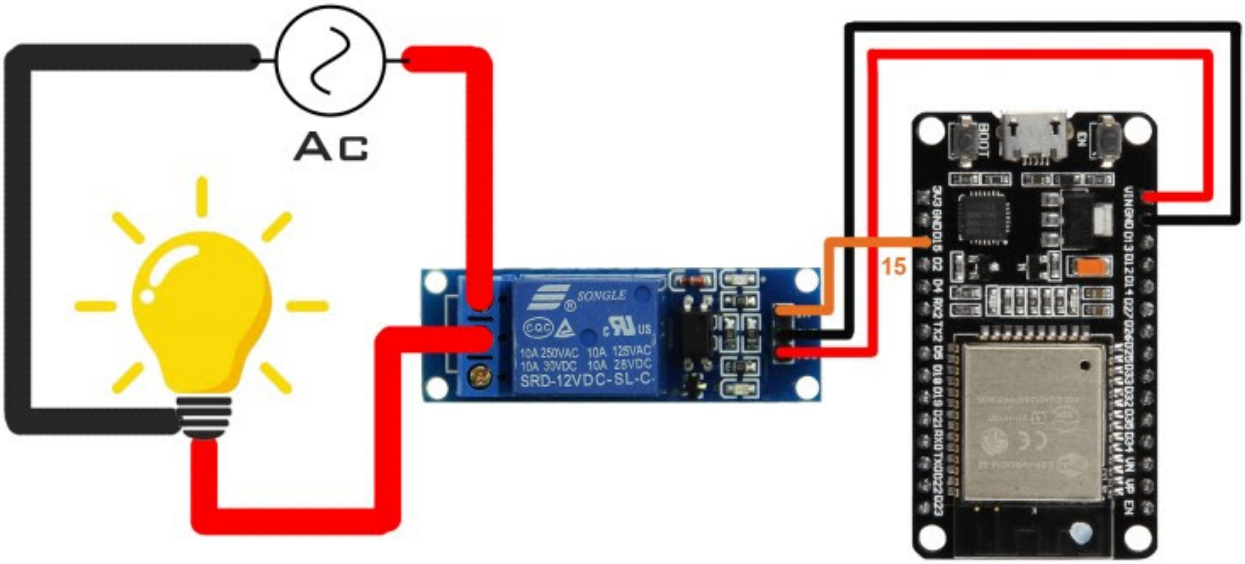
This application is developed on the Unity Engine which helps combine the data from the IOT devices and further connects the data with the AR interface.

This interface is created by Vuforia - a software that works as an image database for storing the target image and recognises the coordinates of that particular image and it further creates an interface on which when the user moves the camera, the interface emerges on the image thus responding to the user's needs.

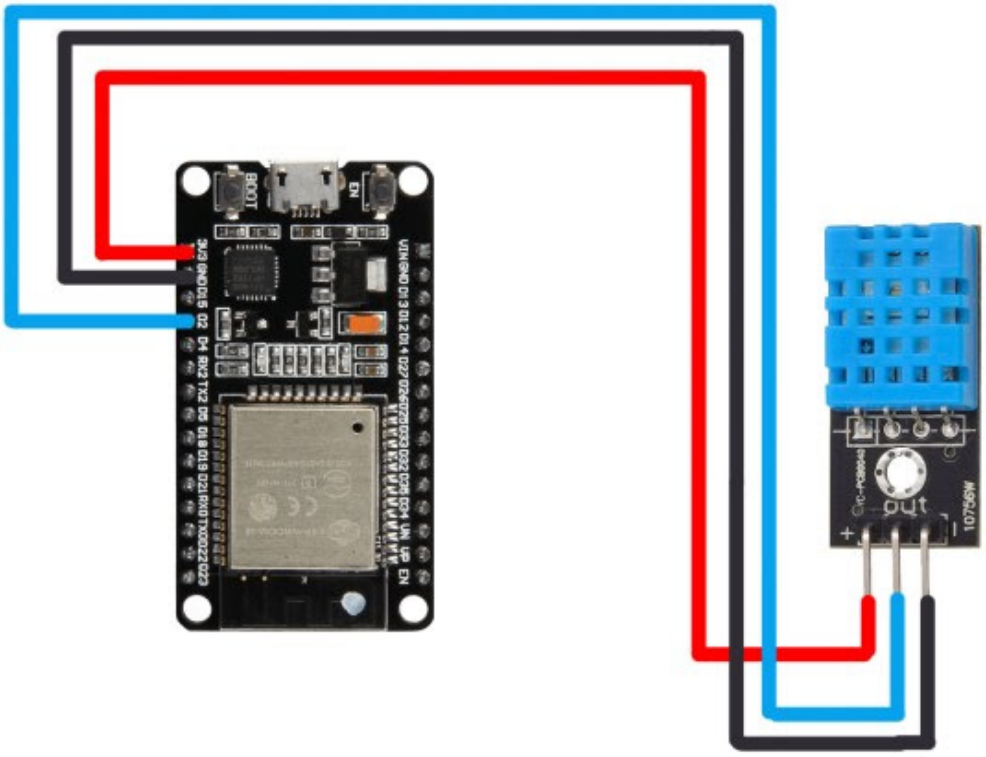
The software allows the user to control or query the device's insights without having to physically contact it. Because this system is developed on the internet, it requires internet access.

Circuit & Code

GADGET 1



GADGET 2



Software

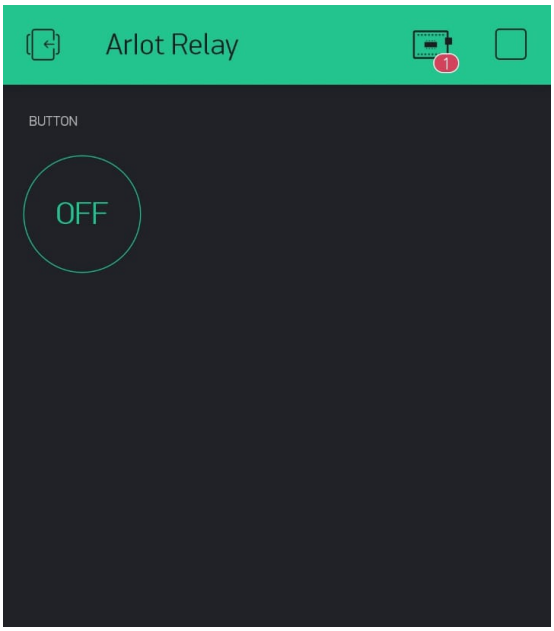
1. Blynk

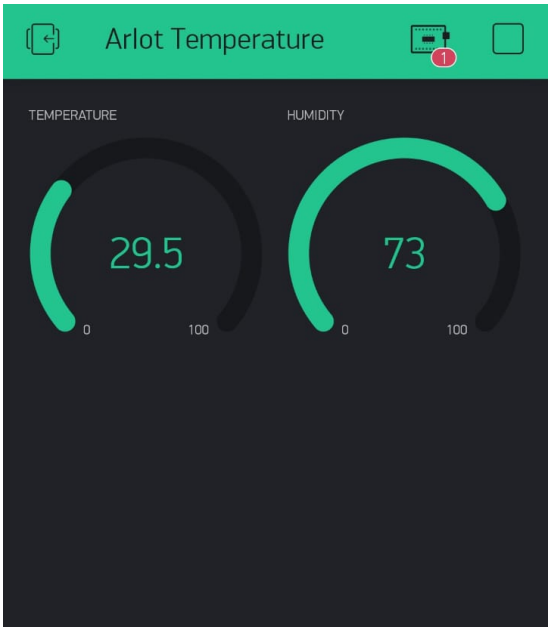
Blynk app is designed to work with the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualise it and do many other things.
It consists of three major components:

- **Blynk App - allows us to create amazing interfaces for your projects using various widgets we provide.**

- **Blynk Server is responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private blynk server locally. It's open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.**

- **Blynk Libraries - for all the popular hardware platforms - enable communication with the server and process all the incoming and out-coming commands.**

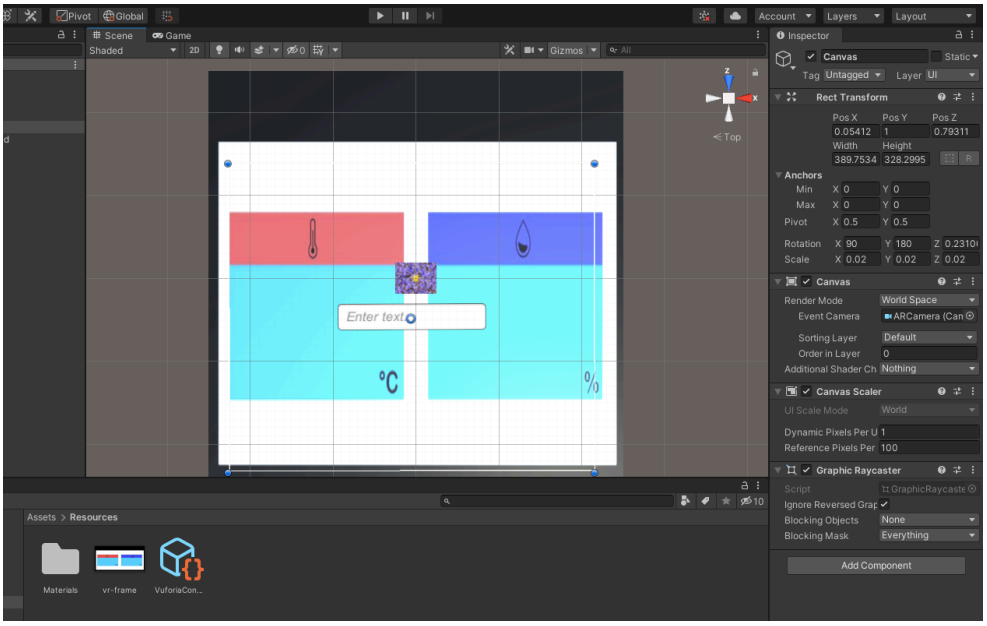


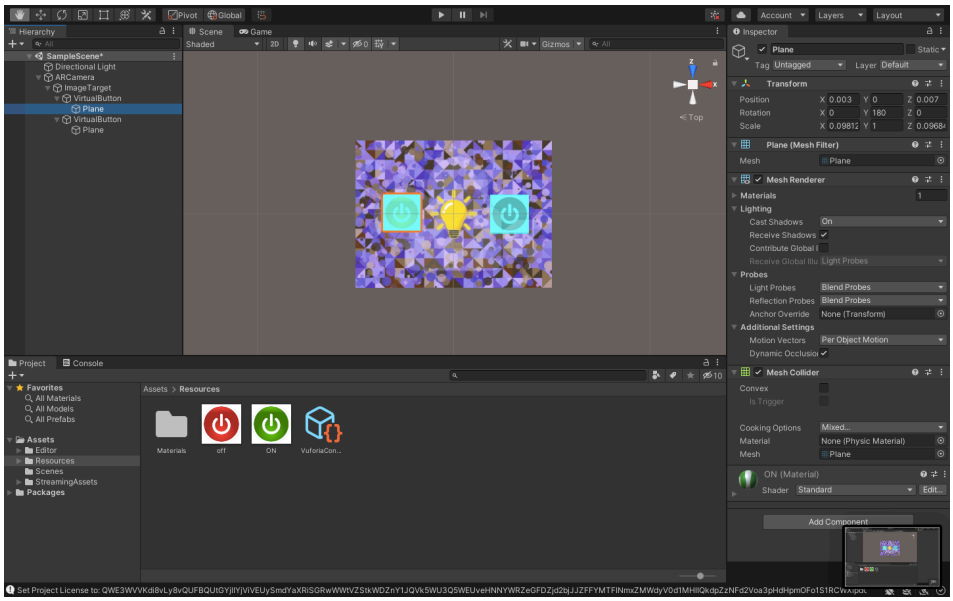


2. Unity

Unity is a 3D/2D game engine and powerful cross-platform IDE for developers. Unity is able to provide many of the most important built-in features that make a game work.

Unity is an IDE. IDE stands for "integrated development environment," which describes an interface that gives you access to all the tools you need for development in one place. The Unity software has a visual editor that allows creators to simply drag and drop elements into scenes and then manipulate their properties. The Unity IDE uses C# to handle the code and logic.





3. Vuforia Developer Engine

Vuforia View is an application that delivers Augmented Reality (AR) enabled experiences to a mobile or eyewear device. It is used to access and utilise an AR experience.
Vuforia View works by tracking in your real-world space where to launch and anchor the AR content. There are different tracking methods depending on the AR-enabled device categorised under three main categories.

- **When using Vuforia View on a mobile device**
- **When using Vuforia View on a HoloLens device**
- **When using Vuforia View on a RearWear device**

C# Code For Unity

GADGET 1

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

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

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

        }
    }

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

        Vb_off.RegisterOnButtonPressed(OnButtonPressed_off);

    }


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

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

}
```

GADGET 2

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

public class click : MonoBehaviour
{
    InputField field;
    InputField Hum;
    public VirtualButtonBehaviour Vb_on;

    void Start()
    {
        field = GameObject.Find("InputFieldTemp").GetComponent<InputField>();

        Hum = GameObject.Find("InputFieldHum").GetComponent<InputField>();

        Vb_on.RegisterOnButtonPressed(OnButtonPressed_on);
        // GameObject.Find("GetButton").GetComponent<Button>().onClick.AddListener(GetData);
    }

    public void OnButtonPressed_on(VirtualButtonBehaviour Vb_on)
    {
        GetData_tem();
        GetData_hum();
       //Debug.Log("Click");
    }

    void GetData_tem() => StartCoroutine(GetData_Coroutine1());
    void GetData_hum() => StartCoroutine(GetData_Coroutine());

    IEnumerator GetData_Coroutine1()
    {
        Debug.Log("Getting Data");
        field.text = "Loading...";
        string uri = "http://blynk-cloud.com/LnyllOlWd77kKLmK1m6Dg3FJpZY37mOy/get/v0";
        using(UnityWebRequest request = UnityWebRequest.Get(uri))
        {
            yield return request.SendWebRequest();
            if (request.isNetworkError || request.isHttpError)
                field.text = request.error;
            else
            {

                field.text = request.downloadHandler.text;
                field.text = field.text.Substring(2,2);
            }
        }
          }
     }
     IEnumerator GetData_Coroutine()
     {
         Debug.Log("Getting Data");
         Hum.text = "Loading...";
         string uri = "http://blynk-cloud.com/LnyllOlWd77kKLmK1m6Dg3FJpZY37mOy/get/v1";
         using(UnityWebRequest request = UnityWebRequest.Get(uri))
         {
             yield return request.SendWebRequest();
             if (request.isNetworkError || request.isHttpError)
                 Hum.text = request.error;
             else
             {

                 Hum.text = request.downloadHandler.text;
                 Hum.text = Hum.text.Substring(2,2);
             }
         }
     }
}
```

Future Developments

We intend to develop a mixed reality integrated solution for the automation and manufacturing industries.

This concept will aid workers in emergency circumstances and help them be more efficient in their daily activities.

Applications

- **Used in Automation and Manufacturing Industries.**
- **Detecting and Preventing Pump Failures**
- **Visualising Product Design in Business.**
- **Used to Make Medical Education Interactive.**
- **Monitoring Crops and Farm Vehicles.**
- **Gaming and Entert**

References

https://www.mdpi.com/1424-8220/19/19/4330/htm

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0268-2

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8298525