

IOT

SmartLIGHT

AHMED SHAN

Villa Student ID: S2200355

UWE Student ID: 22023492

CONTENTS

Introduction	3
Items Used	4
Setup	5
How it works?	7
Conclusion	9
Resources	10

Introduction

This project is a prototype of how a smart home would function. In the project, we are going to turn an ordinary bulb to a smarter version of it by using few other electronic equipments. We will go to a brief explanation on how to make this work and how to implement it.

For this project, we are using PHP, Javascript, MySQL, Python and C++.

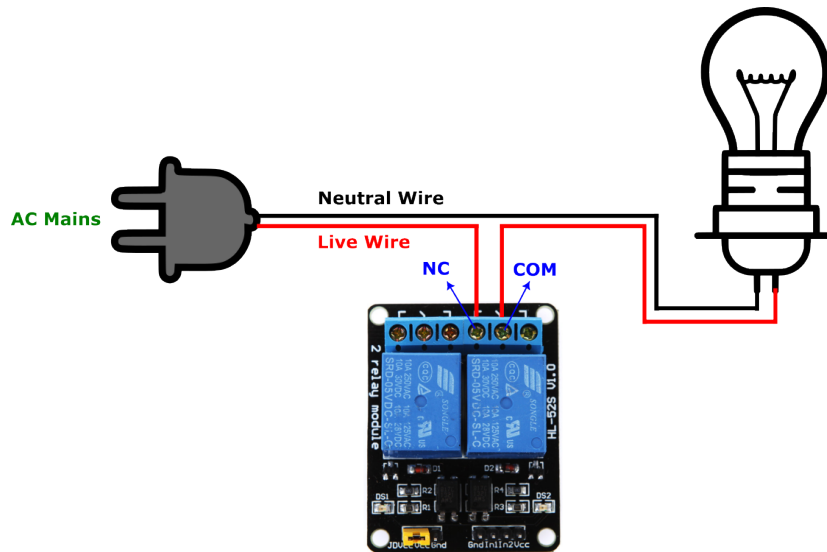
Have a look at the resources section for source code and video demonstration link.

Items Used

	ESP 8266
	BBC Micro:bit
	Arduino Relay Module (2 Channel)
	Jumper Cables
	Micro USB Cable
	Light Bulb
	Light Bulb Holder
	Plug

Setup

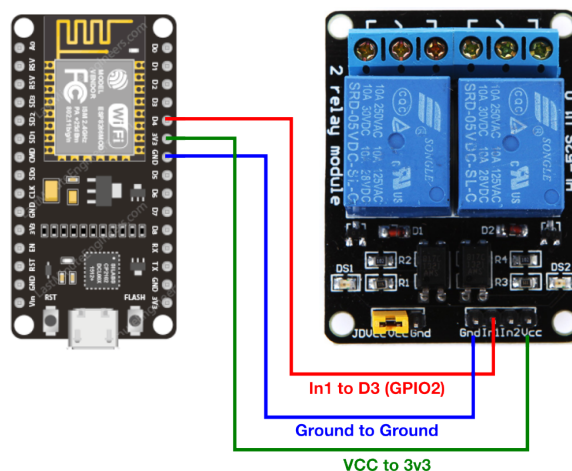
1. Connecting the relay and bulb



For this application, we are using a simple bulb circuit. Neural wire (black colour wire) is connected to the build directory from the plug. Live wire (red colour wire) is also connected to the bulb, however we place the Arduino relay module in between the bulb and plug which will work as a switch.

The Live wire that comes from the plug is connected to the NC connector of the relay and the other end of the Live wire is connected to the bulb from the COM connector of the replay module.

2. Connect the Arduino Relay Module to ESP8266, using jumper wires



- A. Connect **Input 1 (In1)** Pin of the relay to the **D4 (GPIO2)** Pin of the ESP8266.
- B. Connect **Ground (Gnd)** Pin of the relay to the **Ground (Gnd)** of the ESP8266.
- C. Connect **VCC** Pin of the relay to the **3v3** of the ESP8266.

How it works?

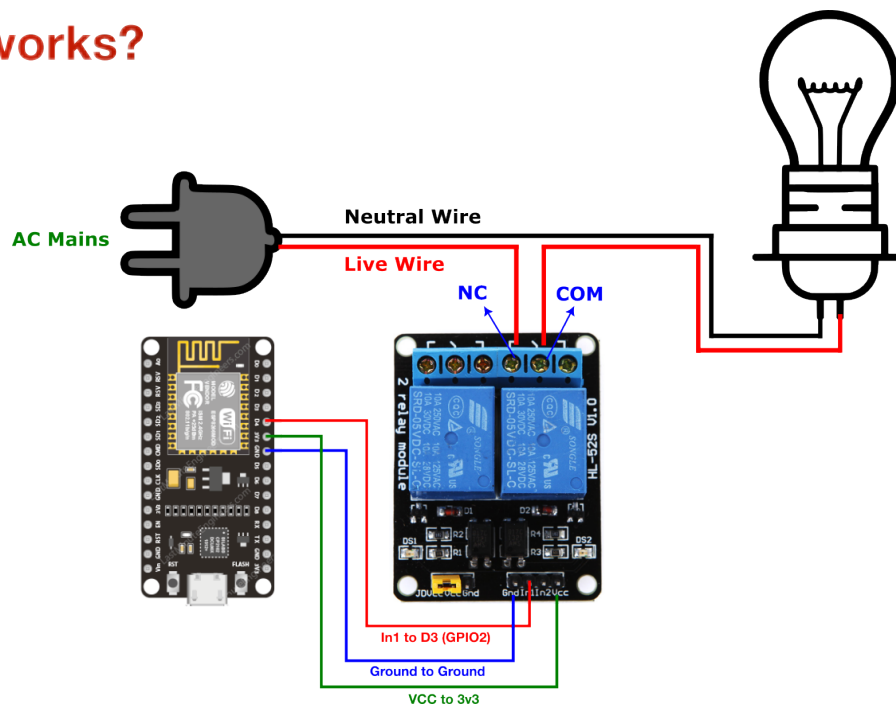
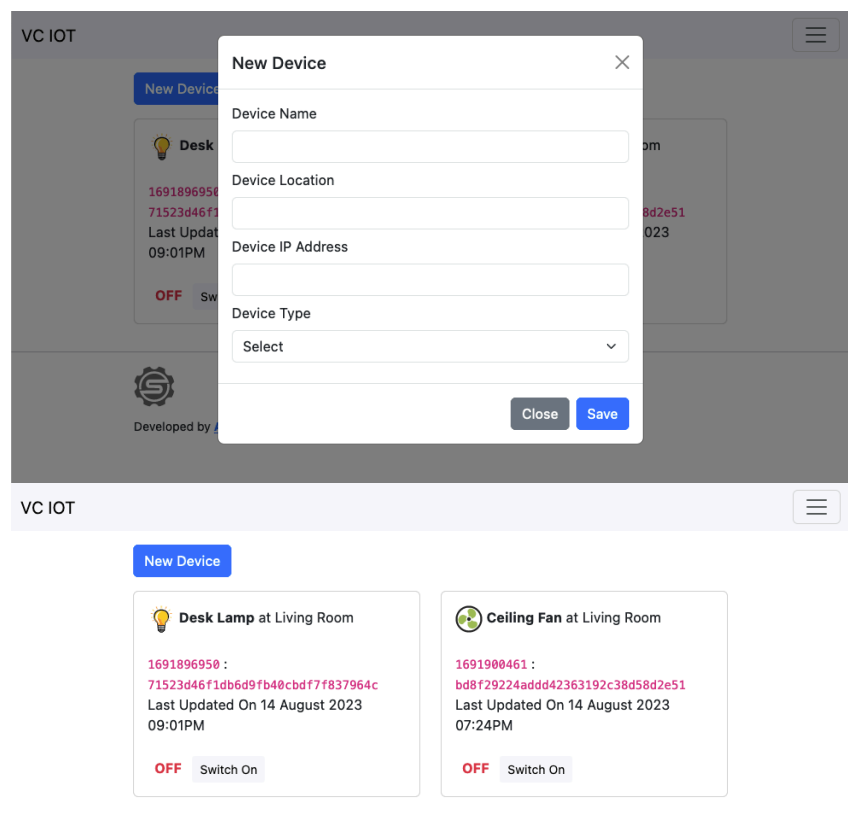


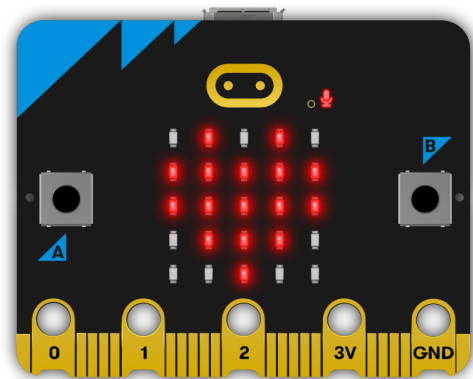
Diagram of the full setup

There are three parts for this project. Micro:bit controlling the bulb, ESP controlling the relay module to switch the bulb turning on and off and a web application that maintains the status of the bulb and also which also can be used for controlling the bulb.

First lets talk about the web application. Initially a device needs to be registered in the system. Upon registration, each device gets a unique ID (UID) and a hash key which must be used to communicate with the web service.

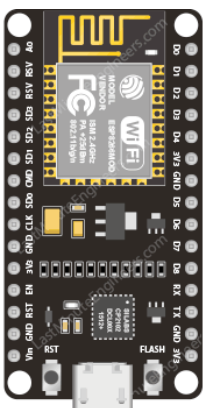


Micro:bit has 2 button controls and 1 touch control which can be used for 3 different devices. For this project we have used Button A as a toggle button to control the bulb. The model, we used does not have Wi-Fi capability, therefore we will send the signal to a computer via USB serial communication.



By pressing Button A on the micro:bit, it sends a text "L1" (which refers to light 1) to the computer. The computer receives the signal and identifies the requested device, then sends a post request to the web server along with the UID and hash key.

When the web server receives the request, it looks up for the device and updates the status of the device to the opposite status. If current status of the device is ON then the server sets the status to OFF and updates the database.



On the other end, the EPS8266 keeps on checking for a device status update. Upon receiving the update and depending on the status the ESP turns of Arduino relay module ON or OFF.

Since the web server is cloud based, all the devices can be controlled remotely from everywhere around the world.

Conclusion

We have used one button of the micro:bit to control a single light, likewise we can add two more devices to the same micro:bit to control them.

This project was a fun project to work on. Identifies challenges and how to overcome them using different technologies and hacks.

Resources

1. Web Application

URL: <https://vciot.ahmdshan.com>

Username: admin

Password: admin@vciot

2. Video Demonstration

1. Youtube Video Link:

3. GIT Repositories

1. UWE GIT LAB: (unable to access uwe git lab due to IP blocking)

2. GIT HUB: <https://github.com/thaanu/smart-light>