Lab 6 Report
IoT Garage Door Automation Through HomeAssistant and IFTTT

Online Link:
This lab is available as part of my online portfolio at: https://github.com/DemonPiranha/ITC-441-labs.git

Objective

The purposes of this lab are to:

- Build a controller to interface with the existing garage door opener. (You don't actually have to connect the relay to a garage door opener but should be able to show that it is performing the necessary action to toggle the garage door opener.)
- Integrate the garage system with at least one external service (IFTTT, Adafruit,io, etc) to introduce some additional functionality of your choice.
- Utilize a non-web interface for the garage door opener

Materials
I used the following materials to accomplish this lab:

- 1 x Arduino Wemos D1 Mini with ESP8266
- 1 x Raspberry Pi 3b+
- 1 x MicroUSB Cable with power source
- Wifi device, this lab uses a smartphone with a wifi hotspot
- 1 x breadboard
- DuckDNS account

References
I used the following resources in this lab:

- https://esphome.io/cookbook/relay.html - Used for ESPHome programming for the relay.
- https://siytek.com/home-assistant-remote/#How-To-Access-Home-Assistant-with-DuckDNS-and-Port-Forwarding - Setting up home assistant to be accessible from an external URL on a secure SSL connection.
- https://www.home-assistant.io/integrations/ifttt - Beginning IFTTT integration.
- https://siytek.com/home-assistant-and-ifft-webhooks-example/ - Guide to creating IFTTT webhooks.
- https://1drv.ms/w/s!Ais81h0TsK5NhqEUuevEs8zMavzLag?e=hdaL2m - Todd Berrett's example write-up. Used for formatting.

Procedures:

1. With HomeAssistant running and ESPHome functioning configuration of the device is needed.
2. Within ESPHome configure the Arduino device that includes the relay by creating a new device and copying the .yaml file into them. This will set up a switch in HomeAssistant that has a device id and action associated with it which will be used by IFTTT.
3. Since we will be accepting triggers from IFTTT we need to configure an automation to handle these triggers.
4. First, add the IFTTT integration to HomeAssistant so that it can properly understand IFTTT triggers.
5. Then navigate to the automation tab and create a new automation copying the .yaml code in Appendix 4.
6. Now configure HomeAssistant to be accessible outside the local network. Follow this guide to configure DuckDNS to work with HomeAssistant: https://siytek.com/home-assistant-remote/#How-To-Access-Home-Assistant-with-DuckDNS-and-Port-Forwarding
7. Now that HomeAssistant is accessible anywhere on the internet IFTTT needs to be set up to trigger the correct device off of a GPS location action. Create a new applet in IFTTT and enter the information found in Appendix 2 making sure to replace the URL received from HomeAssistant when installing the IFTTT integration into the applet. Then set the body to call the switch_on function for the entity_id that matches the name of the relay device you configured in ESPHome.

System view

This project simulates an environment of a garage door being controlled from an external device automatically with minimal user input. A relay control by an Arduino simulates the door. Communication is made through ESPHome with HomeAssistant over wifi. External network access is provided by DuckDNS and automation is triggered by IFTTT.

Relay module

This part includes the Ardunio and the relay shield on top. Picture included in Appendix 3. Since the relay connects to the top of the Ardunio power and single are connected correctly. For this relay, the D1 pin is used to control the relay and should be set in the ESPHome configuration. A connection from the relay to the garage controller would be needed to complete the goal of controlling an actual door however for simulation purposes turning the relay on and off is sufficient.

HomeAssistant module

HomeAssistant requires more configuration to allow for external connections to the dashboard and its features. To allow access from outside sources a publicly accessible address needs to be created. DuckDNS is a free Add-on to HomeAssistant that allows this. Duck DNS provides a URL that can be used by external services to access the HomeAssistant system. API calls are able to be made to control the automation set but HomeAssistant.

Router module

The network, the HomeAssistant device is connected to needs to provide access from outside into the network directly to just HomeAssistant on port 8123. This allows the external URL to communicate properly internally. For this to work, a port forwarding rule needs to be established in the network that forwards external port traffic to the specific IP address of HomeAssistant.

IFTTT module

This module runs off a phone or other mobile device and uses Applets to create an If This Then That process. Currently, the IFTTT Applet is triggered when the mobile device running the software enters a geolocation that is pre-set. This fires off a webhook API call that communicates directly to the HomeAssistant public URL and the HomeAssistant itself.

Altogether

This system allows a user to enter an area near their garage with the garage opening automatically without any direct interaction.

Component View

Model of the functionality, logical flow, and components of the system as controlled by IFTTT Applets.

The logical flow for the process completes as follows with explanations along the route provided.

- IFTTT Geo Location Applet is triggered when the device enters the geo-fenced location within a radius of the garage.
    - IFTTT has access to the mobile device's location services providing it with accurate information as to the location.
    - If the GPS location enters into the radius set the applet is triggered which sends an HTTP web POST request in JSON format.
    - This JSON contains the feature to be initiated and the action of that feature. In this case that is a feature of a switch with the action being to switch it on. Following this is the entity id which specifies which device within HomeAssistant should receive this action command.
    - This Webhook is sent to the Dynamic DuckDNS address for my instance of HomeAssistant.
- DuckDNS receives an HTTP request to the URL and sent traffic along.
    - An HTTP request is received at the https://portersville.duckdns.org:8123 address and sent to my HomeAssistant network which is connected to DuckDNS through a token.
- My Network receives this HTTP request and handles it.
    - The HTTP request is made to a specific port which is 8123. This is forwarded to the internal IP address of my HomeAssistant device along that same port.
- HomeAssistant receives the request and handles it.
    - HomeAssistant ingests the HTTP request using an automation that reads webhooks from IFTTT.
    - The JSON body of the request specify and action of switching on the relay device which the automation performs.
- ESPHome communicates this automation to the relay device turning it on.
    - Once the relay is on the garage door is simulated as opening.

Observations

Working on this was fun to see how all of what we have learned and built can come together in a seamless package that would make life easier. I can’t wait to put this into my own house in order to control my garage and anything else I find useful. It would be so cool to drive up and have the door open automatically and set various routines to run that turn on house settings.

Thought Questions

1. What services did you consider integrating into your project and why?

From the get-go I was already considering using IFTTT as I have used it in the past for projects and for automation in my own life. I also thought bout using Samsung Bixby routines to trigger things from my phone but other than things inside the phone features are limited.

2. What services would you like to integrate in the future?

Integrating the system with my other smart home devices would be cool as I could set up various automation to run when I come to the house or leave the house.

3. Consider internal and external security threats. Identify 3 likely attack vectors for someone to compromise this system? How did you mitigate these?

Someone could try and access my HomeAssistant dashboard as it is now accessible outside my network. To better secure this I made sure to set a better password than originally set.

Attacks could be made on the DNS connection to my HomeAssistant. To protect this I configured HomeAssistant and DuckDNS to use secure SSL protocol for HTTPS. This protects credentials from being passed in plain text.

Any threat that attacks the infrastructure of this system would effectively disable its abilities. Access to the mobile device running IFTTT could compromise the system. Damage to the wifi network running internal communication or to the HomeAssistant device would stop the ability for the automation.

4. What was the biggest challenge you overcame in this lab?

Getting the triggers on IFTTT to happen consistently to thoroughly test the system. Since I didn't want to move around enough to trigger the GPS I used a spoofing app to do testing. Even though I was able to change my location IFTTT would still not trigger the applet. This required some investigation to find how IFTTT interacts with my phones and the necessary permissions and configuration that needs to happen in order to get communication happening correctly.

5. Please estimate the total time you spent on this lab and report?

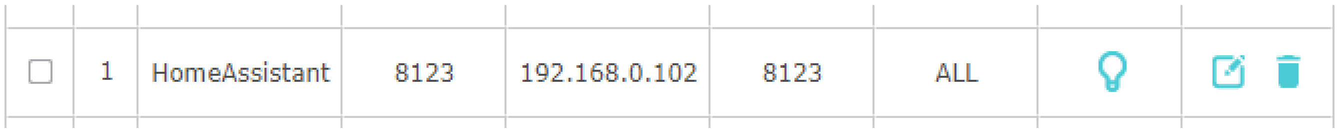
I spent about 10 hours on the project with 4 hours on this lab report.

Certification of Work

I certify that the solution presented in this lab represents my own work. In the case where I have borrowed code or ideas from another person, I have provided a link to the author's work in the references and included a citation in the comments of my code.

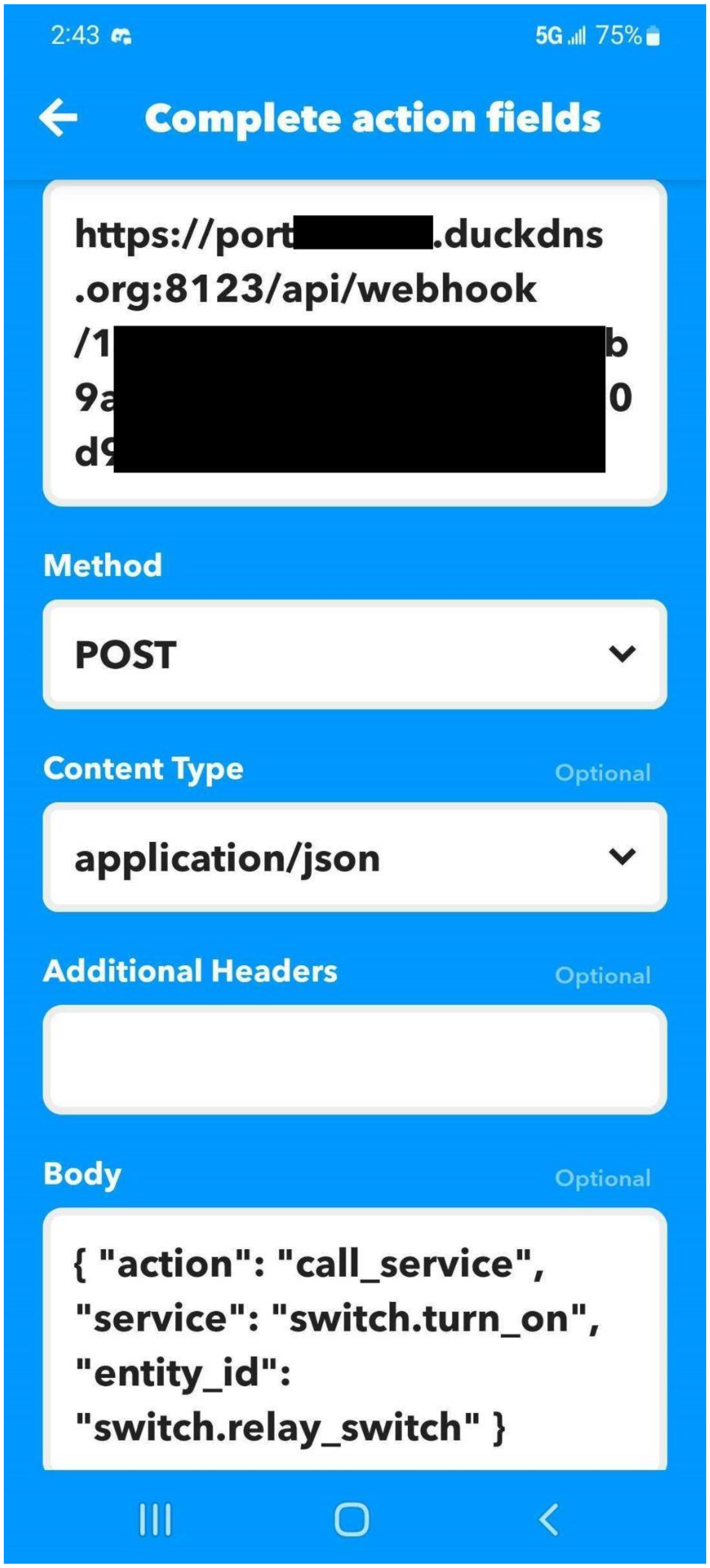
--Blake Porter

Appendix 1: NAT Forwarding Rule



☐	1	HomeAssistant	8123	192.168.0.102	8123	ALL		

Appendix 2: IFTTT Webhook trigger from GPS location



https://port[redacted].duckdns.org:8123/api/webhook/1[redacted]b 9a 0 d9

Method

POST

Content Type — Optional

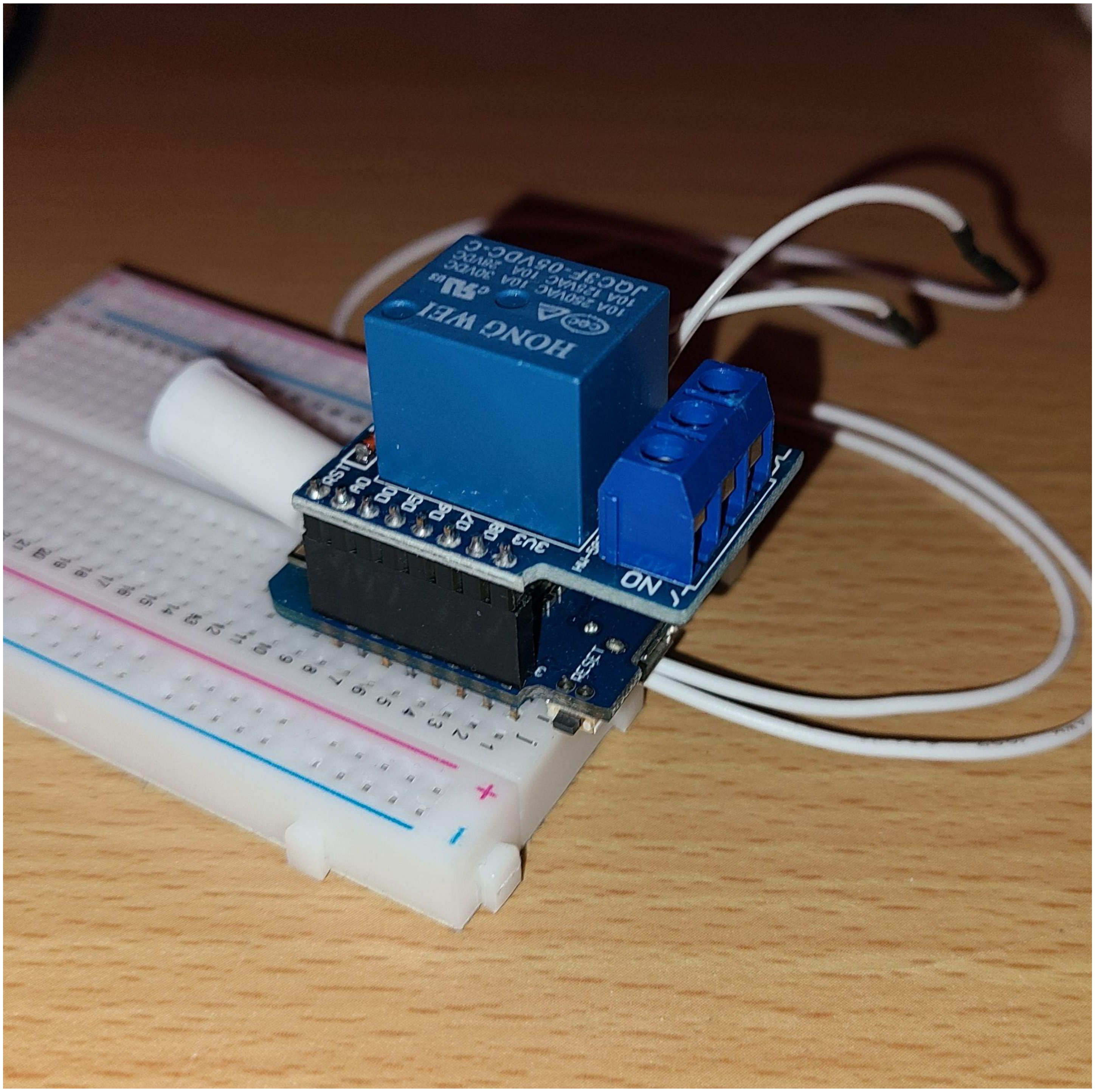
application/json

Additional Headers — Optional

Body — Optional

```
{ "action": "call_service",
"service": "switch.turn_on",
"entity_id":
"switch.relay_switch" }
```

Appendix 3: Hardware setup



Appendix 4: Home Assistant Automation to Receive IFTTT Webhooks

https://github.com/DemonPiranha/ITC-441-labs/blob/main/Lab%206/WebHook%20Inbound%20Automation.yaml

```yaml
alias: IFTTT WebHooks
description: ""
trigger:
  - platform: event
    event_type: ifttt_webhook_received
    event_data:
      action: call_service
action:
  - service_template: "{{ trigger.event.data.service }}"
    data_template:
      entity_id: "{{ trigger.event.data.entity_id }}"
```

Appendix 5: ESPHome Door Sensor Yaml with the addition of relay switch

https://github.com/DemonPiranha/ITC-441-labs/blob/main/Lab%206/Door%20Sensor%20Code.yaml

Door Sensor:

```
esphome:

  name: door-sensor


esp8266:

  board: d1_mini


# Enable logging

logger:


# Enable Home Assistant API

api:

  encryption:

    key: "BtcKVNjdSxtHiKddwcRkHgLfaMC27iIG/czfviO/9Fo="


ota:

  password: "473fbd7d2a3102dce8b89a788fa8f39f"


wifi:

  ssid: !secret wifi_ssid

  password: !secret wifi_password


  # Enable fallback hotspot (captive portal) in case wifi connection fails

  ap:

    ssid: "Door-Sensor Fallback Hotspot"
    password: "MYCPCgnvtIy1"


captive_portal:


binary_sensor:

  - platform: gpio

    name: "Garage Door Sensor"

    pin:

      number: D1

      mode:

        input: true

        pullup: true


switch:

  - platform: gpio

    name: "Relay Switch"

    pin: D1


light:

  - platform: status_led

    name: "Garage Door Status Indicator"

    pin:

      number: D4

      inverted: true
```