Lab 5 Report
IoT System of Garage Door and Parking Distance Indicator through Home Assistant

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:

- Install an open-source home automation platform
- Interface this platform via MQTT with your various devices
- Use Home Assistant to monitor the state of the garage door, and whether the car is parked in the garage.

Materials
I used the following materials to accomplish this lab:

- 3 x Arduino Wemos D1 Mini with ESP8266
- 1 x Raspberry Pi 3b+
- 4 x MicroUSB Cable with power source
- Wifi device, this lab uses a smartphone with a wifi hotspot
- 1 x SR04 Ultrasonic sensor
- 3 x breadboards
- 1 x Reed switch
- 1 x Green LED
- 1 x Yellow LED
- 1 x Red LED
- 3 x 100 Ω Resistor (BrBlBrGl)
- 8 x Jumper Wires

References
I used the following resources in this lab:

- This lab assumes the hardware setup of all devices from the previous lab
- https://www.home-assistant.io/installation/raspberrypi/
- https://www.home-assistant.io/getting-started/onboarding/
- https://esphome.io/
-
- https://1drv.ms/w/s!Ais81h0TsK5NhqEUuevEs8zMavzLag?e=hdaL2m - Todd Berrett's example write-up. Used for formatting.

Procedures:

1. Follow the Home Assistant guide to install Home Assistant on a RaspberryPi found here: https://www.home-assistant.io/installation/raspberrypi/
2. Follow the configuration steps here: https://www.home-assistant.io/getting-started/onboarding/
3. Once in Home Assistant install the ESPHome Add-on
4. Within ESPHome Configure all of the sensor devices by creating a new device and copying the .yaml files into them.
5. Once all of the devices are connected and talking to Home Assistant you should be able to see these in the dashboard and manually control lights on it.
6. To connect all of the devices logically, automations need to be created. yaml files for these automations can be found in the repository.

System view

This project simulates an environment of a garage with a door, and parking spot. Two sensors and an actuator are used which communicate to a central event hub controlled by an ESPHome addon for Home Assitant. All devices are communicating over a shared wifi network.

Stoplight module

This part includes the Ardunio and the three LEDs to model a stoplight. The user is able to receive visual feedback from these lights determined by the ultrasonic distance module while the garage door is open. Once the door is closed all lights are turned off until a new light automation is triggered. All of the LEDs are configured as binary lights in ESPHome allowing Home ASsistant to see them as simple on-and-off lights for use in automation.

Ultrasonic distance module

This part includes the second Ardunio and an SR04 ultrasonic distance sensor. The module as a whole receives distance readings from the sensor and sends this data back to Home Assistant. The distance data stream is sampled every 0.1s and then is filtered before being sent using a median window size of 6 samples with a distance being sent every 3 samples. This provides smooth distance data, however, it does reduce the accuracy of the instantaneous data. The module is configured as a simple sensor using the ultrasonic template from ESPHome.

Reed switch module

This simple module contains a reed switch that simulates a garage door being opened or closed. The sensor is hypothetically located at the bottom of the garage door. This allows the sensor to show when the door is definitely secured in the closed position however any location away from the sensor will read as open simulating that the door is open to some capacity. The switch is configured as a simple binary sensor in ESPHome and returns on when the door is open and off when it is closed.

Raspberry Pi module

This device is running the Home Assistant Operating System and also controls all communication between ESP devices.

Altogether

This system allows a user to park a vehicle in a garage at the perfect distance to the wall of the garage using the stoplight indicator to find and track the distance. This system will not provide feedback when the garage is closed and not in use.

Component View

Model of the functionality, logical flow, and components of the system as controlled by Home Assistant automation. All logic for these devices is controlled by Home Assistant automation (listed in Appendix 2) and updates to the Home Assistant dashboard are seen in Appendix 1.

This group of automations controls the basic function taking the distance data and translating it into logical operations to activate specific lights depending on the distance data. The trigger is started by the ultrasonic distance.

Distance > Stoplight Green

- If the UltraSonic sensor is returning a distance above 30cm but below 100cm and the door sensor is on the green light will be turned on and all others off.

Distance > Stoplight Yellow

- If the UltraSonic sensor is returning a distance above 20cm but below 30cm and the door sensor is on the yellow light will be turned on and all others off.

Distance > Stoplight Red

- If the UltraSonic sensor is returning a distance above 10cm but below 20cm and the door sensor is on the red light will be turned on and all others off.

Distance > Stoplight Red Flash

- If the UltraSonic sensor is returning a distance above 1cm but below 10cm and the door sensor is on yellow and green will be turned off. A while loop is activated that repeats while the distance sensors still return a value between 1cm and 10cm and the door is open. The red light is turned on and off with a 200-millisecond delay in between each command.

This group of automations allows for the added feature of re-activating the light at whatever distance state the current distance is at. This turns the light back on when the door is reopened after being closed. The trigger is started by the door sensor.

Door Opened Green On

- When the door sensor is opened and the Ultra Sonic sensor is returning a distance above 30cm but below 100cm the green light will be turned on and all others off.

Door Opened Yellow On

- When the door sensor is opened and the Ultra Sonic sensor is returning a distance above 20cm but below 30cm the yellow light will be turned on and all others off.

Door Opened Red On

- When the door sensor is opened and the UltraSonic sensor is returning a distance above 10cm but below 20cm the red light will be turned on and all others off.

Door Opened Red Flash On

- When the door sensor is opened and the UltraSonic sensor is returning a distance above 1cm but below 10cm the green light will be turned on and all others off. A while loop is activated that repeats while the distance sensors still return a value between 1cm and 10cm and the door is open. The red light is turned on and off with a 200-millisecond delay in between each command.

This group of automations changes the status of the “Vehicle in Garage Status” indicator located on the dashboard. The trigger is started by the ultrasonic distance.

Car in Garage No

- If the UltraSonic sensor is returning a distance above 30cm but below 100cm then it will turn the distance sensor's status indicator light off, notifying of the absence of an object. This effectively represents that a vehicle is not within the garage.

Car in Garage Yes

- If the UltraSonic sensor is returning a distance above 1cm but below 30cm then it will turn the distance sensor's status indicator light on, notifying of the presence of an object. This effectively represents that a vehicle is within the garage.

This automation shuts off all lights on the stoplight when the door is closed. The trigger is started by the door sensor.

Door Sensor > Stoplight All Off

- If the Door sensor returns closed then all lights on the stoplight module are shut off. This shut-off command is repeated twice to ensure that all the lights get completely turned off and none are left stuck on.

Observations

The lab was a ton of fun. Everything went relatively smoothly and I enjoyed learning all of the things Home Assistant can do. When I first started it up it started detecting all of my other smart devices which was really cool. I will definitely consider establishing this in my home going forward as a central command center. Yaml configuration files are provided in Appendix 2 as well as the GitHib repository.

Thought Questions

1. Which version of Home Assistant did you choose to install? (Home Assistant Operating System, Home Assistant Container or one of the more experienced versions) Why did you choose this version?

I used the Home Assistant Operating System as I wanted to reduce all risk of incompatibilities so I opted for the clean OS install and it was very straightforward. I also have other devices that would be able to run this operating system and so I wanted to test out the experience on the PI first.

2. How should you decide which logic to perform in Home Assistant versus coding the logic directly into the devices? What guiding principles would you establish for future devices?

With previous labs, I liked the idea of doing parts of the logic across all devices in the hope to reduce the computation load between all devices. Once I got Home Assistant running however I saw how easy it was to create automation in the GUI. Because of this I opted to go for a central logic approach where I design each of the devices to primarily send their raw data and then I would use that data to run automation off of Home Assistant. This would allow me to establish a principle of creating bare minimum devices and centralizing the logic. This also makes it easier to adjust the logic as you test without having to re-upload any code to devices.

3. What features do you like the most about Home Assistant?

I really like Home Assistant's ability to pull all sorts of data out from devices. This primarily goes for generic devices but you can also program things into your own devices very easily. I really like having a GUI for creating automation as I am able to be far more creative with the automation instead of trying to wrap my head around how the technical things need to work. I also like how many add-ons are available for it including ESPHome which was really fun to work with.

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

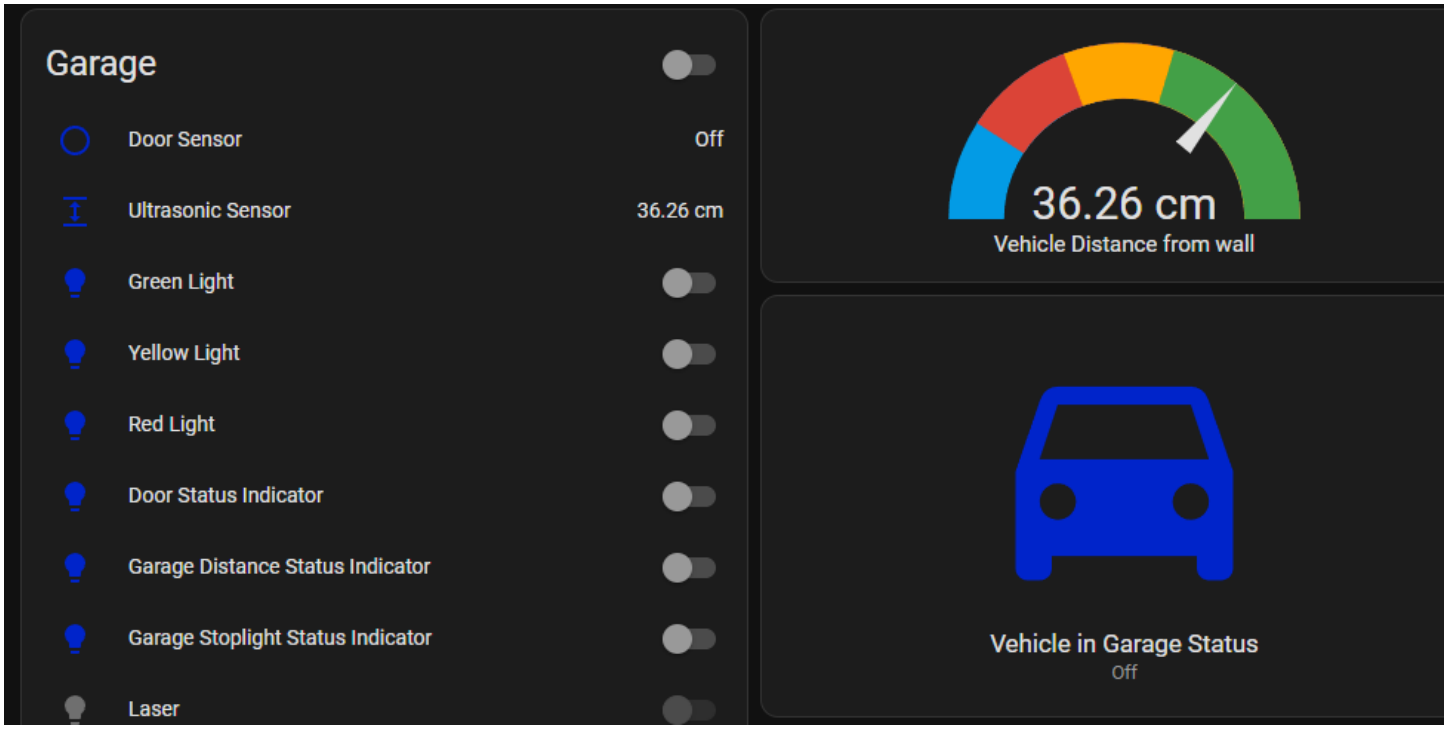
I spent about 12 hours on the project with 3 hours on this lab report.

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

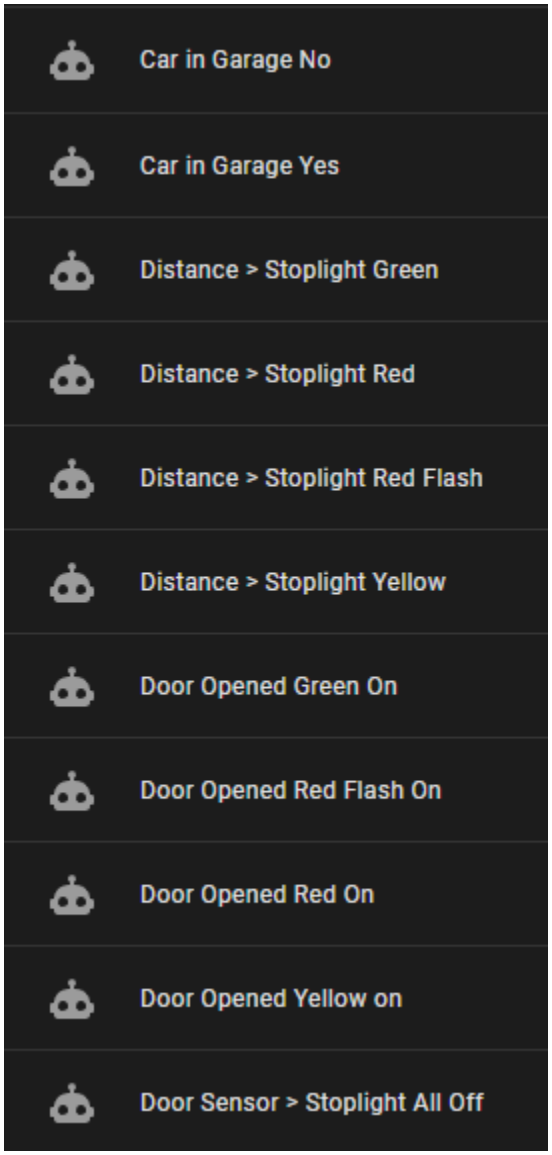
Appendix 1: Home Assistant Dashboard



Appendix 2: Home Assistant Automations

Some examples of each different type will be shown but all are accessible here:
https://github.com/DemonPiranha/ITC-441-labs/tree/main/Lab%205/Automations

List of all automations:



Distance > Stoplight Green:

```
alias: Distance > Stoplight Green
description: ""
trigger:
  - type: value
    platform: device
    device_id: 45b49d3b8c324ff1b8402005040628b1
    entity_id: sensor.ultrasonic_sensor
    domain: sensor
    above: 30
    below: 100
condition:
  - condition: and
    conditions:
      - type: is_on
        condition: device
        device_id: 6628b5f7c3e24b93dc9c6f117d1280a7
        entity_id: binary_sensor.garage_door_sensor
        domain: binary_sensor
action:
  - type: turn_on
    device_id: eba45fac4c4764fd905016f1d2670fbe
    entity_id: light.garage_green_light
    domain: light
  - type: turn_off
    device_id: eba45fac4c4764fd905016f1d2670fbe
    entity_id: light.garage_yellow_light
    domain: light
  - type: turn_off
    device_id: eba45fac4c4764fd905016f1d2670fbe
```

```
    entity_id: light.garage_red_light
    domain: light
mode: single
```

Door Opened Green On:

```
alias: Door Opened Green On
description: ""
trigger:
  - type: turned_on
    platform: device
    device_id: 6628b5f7c3e24b93dc9c6f117d1280a7
    entity_id: binary_sensor.garage_door_sensor
    domain: binary_sensor
condition:
  - condition: and
    conditions:
      - type: is_value
        condition: device
        device_id: 45b49d3b8c324ff1b8402005040628b1
        entity_id: sensor.ultrasonic_sensor
        domain: sensor
        above: 30
        below: 100
action:
  - type: turn_on
    device_id: eba45fac4c4764fd905016f1d2670fbe
    entity_id: light.garage_green_light
    domain: light
mode: single
```

Car in Garage Yes:

```
alias: Car in Garage Yes
description: ""
trigger:
  - type: value
    platform: device
    device_id: 45b49d3b8c324ff1b8402005040628b1
    entity_id: sensor.ultrasonic_sensor
    domain: sensor
    below: 30
    above: 1
    for:
      hours: 0
      minutes: 0
      seconds: 0
condition: []
action:
  - type: turn_on
    device_id: 45b49d3b8c324ff1b8402005040628b1
    entity_id: light.garage_distance_status_indicator
    domain: light
mode: single
```

Distance > Stoplight Red Flash:

```
alias: Distance > Stoplight Red Flash
description: ""
trigger:
  - type: value
    platform: device
    device_id: 45b49d3b8c324ff1b8402005040628b1
    entity_id: sensor.ultrasonic_sensor
    domain: sensor
    above: 1
    below: 10
condition:
  - condition: and
    conditions:
      - type: is_on
        condition: device
        device_id: 6628b5f7c3e24b93dc9c6f117d1280a7
        entity_id: binary_sensor.garage_door_sensor
        domain: binary_sensor
action:
  - type: turn_off
    device_id: eba45fac4c4764fd905016f1d2670fbe
    entity_id: light.garage_green_light
    domain: light
  - type: turn_off
    device_id: eba45fac4c4764fd905016f1d2670fbe
    entity_id: light.garage_yellow_light
    domain: light
  - repeat:
      while:
```

```
    - type: is_value
      condition: device
      device_id: 45b49d3b8c324ff1b8402005040628b1
      entity_id: sensor.ultrasonic_sensor
      domain: sensor
      above: 1
      below: 10
    - condition: and
      conditions:
        - type: is_on
          condition: device
          device_id: 6628b5f7c3e24b93dc9c6f117d1280a7
          entity_id: binary_sensor.garage_door_sensor
          domain: binary_sensor
  sequence:
    - delay:
        hours: 0
        minutes: 0
        seconds: 0
        milliseconds: 200
    - type: turn_on
      device_id: eba45fac4c4764fd905016f1d2670fbe
      entity_id: light.garage_red_light
      domain: light
    - delay:
        hours: 0
        minutes: 0
        seconds: 0
        milliseconds: 200
    - type: turn_off
```

```
      device_id: eba45fac4c4764fd905016f1d2670fbe
      entity_id: light.garage_red_light
      domain: light
  - type: turn_on
    device_id: eba45fac4c4764fd905016f1d2670fbe
    entity_id: light.garage_red_light
    domain: light
mode: single
```

Appendix 3: ESPHome .yaml files

Distance sensor:

```
esphome:
  name: distance-sensor

esp8266:
  board: d1_mini

# Enable logging
logger:

# Enable Home Assistant API
api:
  encryption:
    key: "WysYoTm7RJa+yv9uNfgXQtZBq1XlLSxFfbJPQdJlisE="

ota:
  password: "6e991a99b919d063ba40f958796f21e1"

wifi:
  ssid: !secret wifi_ssid
  password: !secret wifi_password

  # Enable fallback hotspot (captive portal) in case wifi connection fails
  ap:
    ssid: "Distance-Sensor Fallback Hotspot"
    password: "CQ8uezqyIxR8"
```

```
captive_portal:


sensor:
  - platform: ultrasonic
    trigger_pin: D1
    echo_pin: D2
    name: "Ultrasonic Sensor"
    update_interval: 0.1s
    unit_of_measurement: "cm"
    filters:
      - median:
          window_size: 6
          send_every: 3
          send_first_at: 3
      - multiply: 100


light:
  - platform: status_led
    name: "Garage Distance Status Indicator"
    pin:
      number: D4
      inverted: true
```

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

light:
  - platform: status_led
    name: "Garage Door Status Indicator"
    pin:
      number: D4
      inverted: true
```

Stoplight

```
esphome:
  name: stoplight

esp8266:
  board: d1_mini

# Enable logging
logger:

# Enable Home Assistant API
api:
  encryption:
```

```
    key: "qt7iJf6TXaDgYA8clDgc3Ri8I6J8hrmXn6zJ3tQmT4w="


ota:
  password: "950e611b356fdc2b01c5b485799ed4cf"


wifi:
  ssid: !secret wifi_ssid
  password: !secret wifi_password

  # Enable fallback hotspot (captive portal) in case wifi connection fails
  ap:
    ssid: "Stoplight Fallback Hotspot"
    password: "FjwQsTs7NJez"


captive_portal:


light:
  - platform: binary
    name: "Garage Green Light"
    output: light_output_green
  - platform: binary
    name: "Garage Yellow Light"
    output: light_output_yellow
  - platform: binary
    name: "Garage Red Light"
    output: light_output_red
  - platform: status_led
```

```
    name: "Garage Stoplight Status Indicator"
    pin:
      number: D4
      inverted: true


output:
  - id: light_output_green
    platform: gpio
    pin: D1
  - id: light_output_yellow
    platform: gpio
    pin: D2
  - id: light_output_red
    platform: gpio
    pin: D3
```