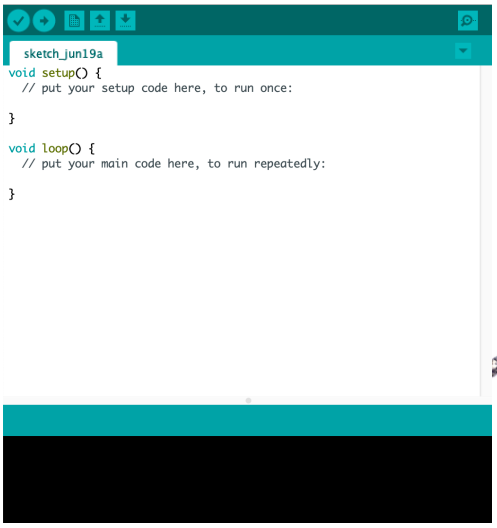
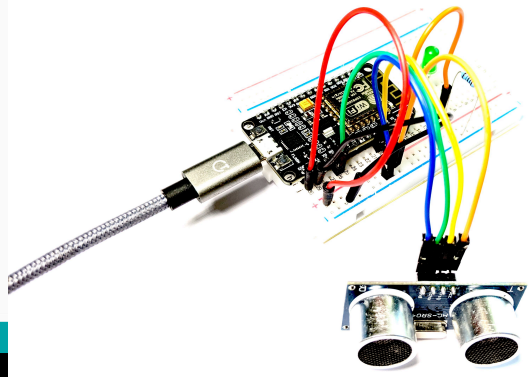
Smart Fridge

Part three: Automation

Project description:

In this project, you will work on your own to add a small automation to your smart door circuit. You will build on the previous smart door exercise and add a green LED to the smart door circuit. Your task is to use PWM to change the brightness of the LED according to the distance from the door. The closer you are to the door, the brighter the LED should become.



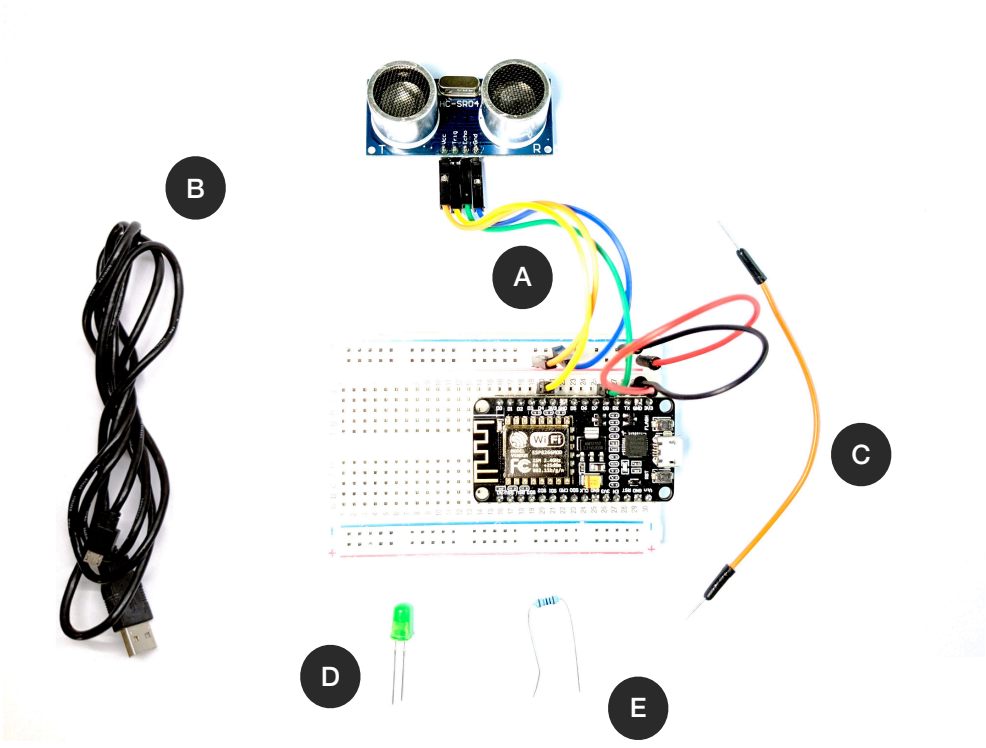


Project objectives:

- Add a green LED to the smart door circuit
- Create a simple automation function to control the LED behaviour

Project components:

Component Reference	Component Quantity	Component Name	Component Description
A	1	Smart Door Part 2	The smart door circuit which features the ultrasound component
B	1	Micro USB Cable	A USB cable to power and upload instructions to a microcontroller
C	1	Jumper Wire	Conductive cables frequently used with a breadboard to connect two points in a circuit
D	1	2V Green LED	A semiconductor light source that emits light when current flows through it
E	1	220 Ohm Resistor	A passive two-terminal electrical component that implements electrical resistance as a circuit element



Step One

Adding the green LED to the Smart Fridge Circuit

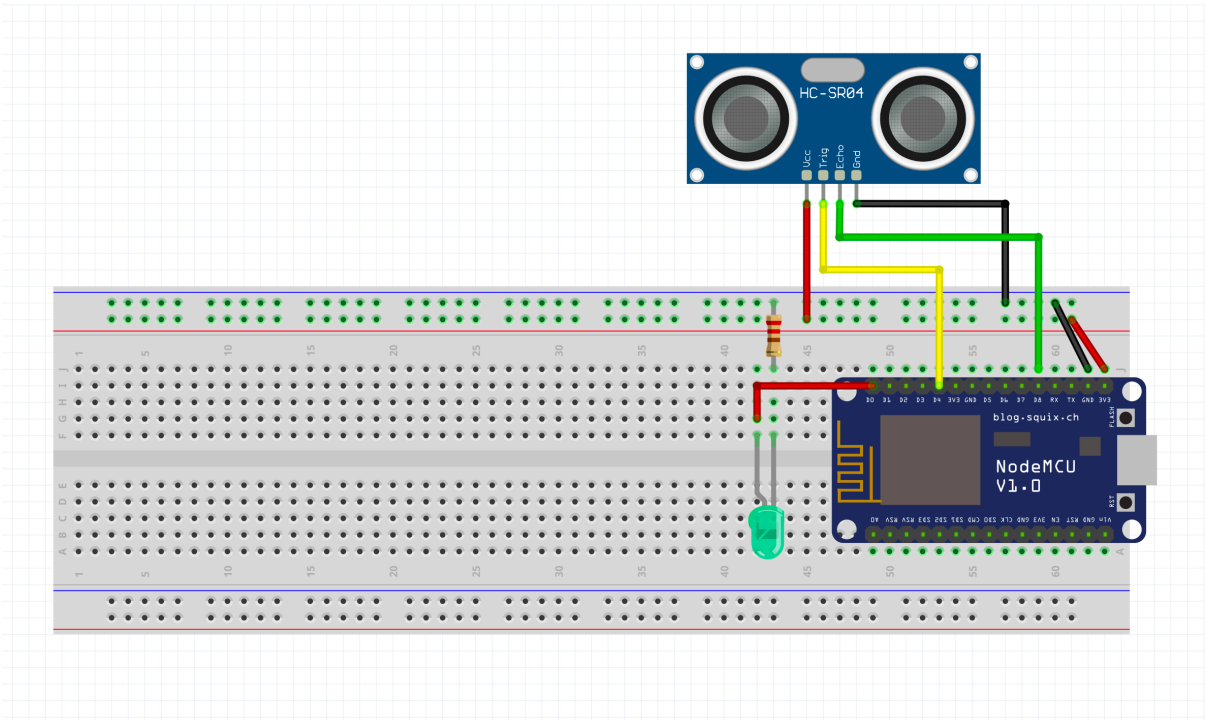
The circuit assembly for this exercise requires you to wire up one green LED and one 100 Ohm resistor.

You will add the LED and the resistor to the existing smart door circuit that you have built so far. Refer back to the second part of the smart door exercise to view the latest version of the circuit.

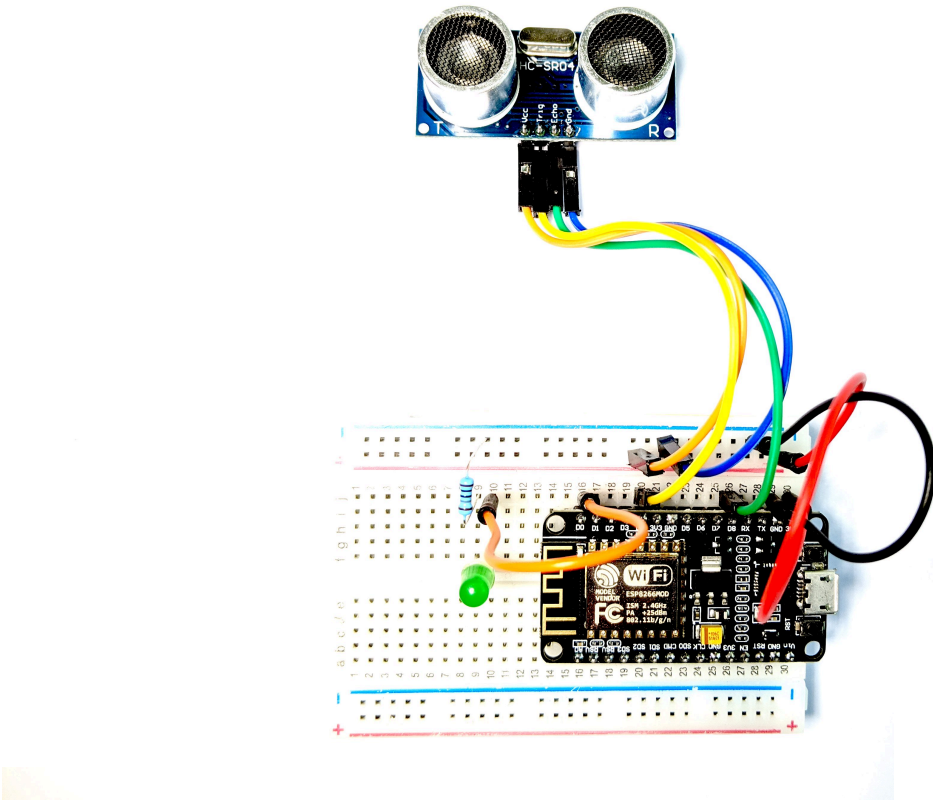
You should be now familiar on how to wire up and LED and the respective resistor to the ESP board. The LED should be connected to pin DO.

It is now your turn to add the additional component to the smart door circuit. Use the same half of the longer breadboard to build this circuit.

The diagram below shows you how the circuit should be assembled. It combines the circuit components for the second part of the smart door circuit with the additional LED and resistor components. Also, use different colour wires for the connections if you do not have wires with the same colours left in your kit.



Furthermore, see below a picture of the circuit assembled:



Step Two
Task Specifications

Now that you have assembled the circuit, it is time to add the code for the LED component automation. Go ahead and create a new empty sketch from the Arduino IDE.

You should see an empty sketch like the following:

Feel free to save the sketch and rename it to something sensible: **smart_door_part3** for instance.

Now copy and paste the code that you wrote for the second part of the smart door exercise. You should have the code with the **distanceCentimeter()** utility function. You should also have the **get_index()** utility function if you completed the previous exercise additional tasks (create a web server dashboard to display the distance).

Quickly upload the code to your ESP board and make sure that it compiles successfully. The Serial Monitor should print the distance from the ultrasound sensor in centimetres.

Your task here is to add one utility function to create a small automation process. The green LED brightness should be relative to the distance from the ultrasound sensor. The brightness of the LED should be maximum when the distance from the ultrasound sensor is equal to 0cm and minimum when the distance from the ultrasound is grater than the 40cm (the maximum distance that the ultrasound can detect).

Step One -> At the top of the sketch, before the setup() function:

• Initialise a const int variable called ‘ledPin’ with a reference to pin D0.

Step Two -> At the end of the setup() function, but inside its body:

- Set the LED pinMode to OUTPUT
- Set the initial value of the LED pin so that it has no brightness (analogWrite)

Step Three -> At the end of the sketch, below the loop() function:

- Create a void function called 'ledControl'
- Create an int variable inside the function called 'mappedValue'
- Assign to 'mappedValue' the value obtained from the distance mapping. The map function should map the distance value so that it returns 255 when the distance is 0 and 0 when the distance is 40
- Use analogWrite to set the LED to the mapped distance value

Step Four -> At the end of the loop() function, but inside its body:

- Call the ledControl function

Step Six -> Upload the code to your smart door circuit:

- You should see the green LED brightness change according to the distance from the ultrasound sensor. Putting your hand in front of the sensor should set the LED brightness to the maximum value (255). On the other hand, the brightness should lower as you move your hand away from the sensor. The LED should be OFF when the recorded distance is above 40cm.

Step Three
Additional Tasks

In upcoming exercises, you will keep adding new components to the smart door circuit. You will also integrate more automations.

Meantime, you can try the following:

- Modify the map function to only change the LED brightness when your distance from the door is in the range of 0cm - 20cm.

- Print a message ("Door should now open") on the Serial Monitor when your distance from the door is below a certain threshold (e.g. 20cm).