

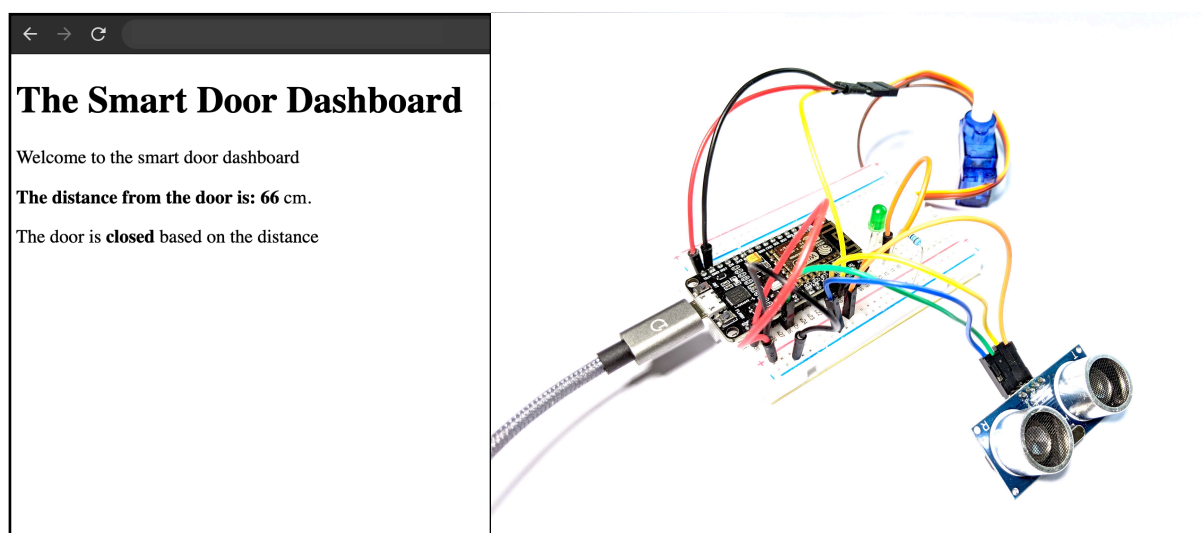
# Smart Door

## Part six: Automation

### Project description:

---

You have now written quite a lot of code for the smart door project. In the process of assembling the circuit, you explored new digital and analog components such as the distance sensor and the servo motor. In this exercise, you will work on your own on the existing smart door circuit and modify the code to automate the door opening and closing mechanism according to the distance detected by the ultrasonic sensor.



### Project objectives:

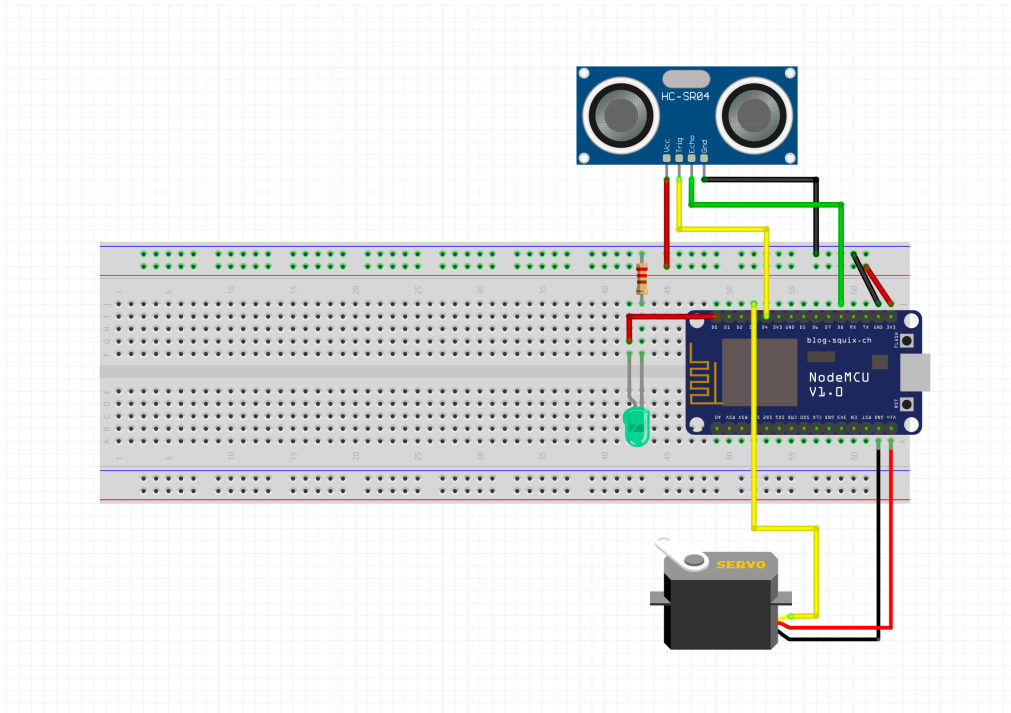
---

- Modify the smart door circuit code to add one automation
- Display the automation outcome on the web server dashboard

# The Smart Door Automation Exercise Specifications

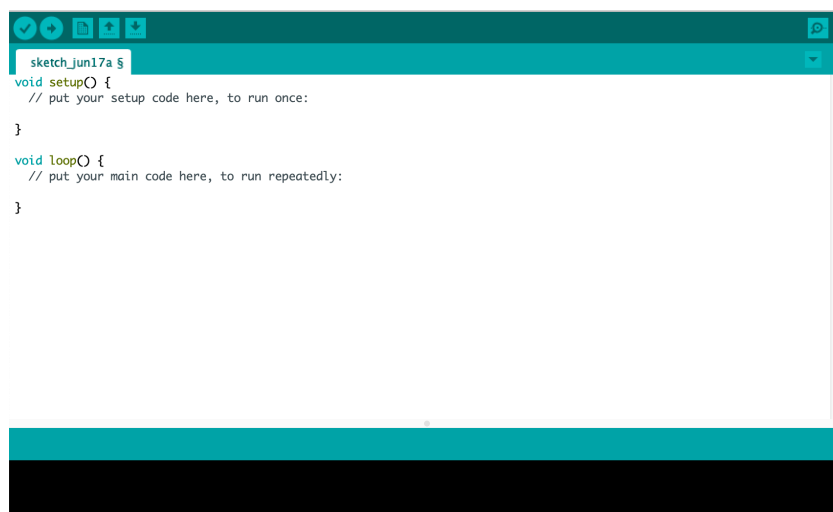
---

You will work on the latest version of the smart door circuit:



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\_part6** for instance.

Now copy and paste the code that you wrote for the fifth part of the smart door exercise. You should have the code with the **ledControl()**, the **distanceCentimeter()**, the **jsonDistanceSensor()**, the **servoMovement()**, and the **get\_json()** utility functions. You should also have an additional function **get\_index()** and the web server functionality for the smart door dashboard.

The **servoMovement()** function that we wrote on the previous exercise was just to get you familiar with the code to control the servo motor component. In this exercise, your task is to transform that function into a useful automation process for your smart door. Change the function to automatically open the door (servo motor) when you are 30cm or less away from the door (distance sensor). Furthermore, use the web server dashboard to write “The door is **closed** based on the distance” when your distance from the door is above 30cm and “The door is **open** based on the distance” when your distance from the door is 30cm or less.

You are expected to attempt the following:

- Add a String global “doorStatus” variable to keep track of the door status: either “open” or “closed”.
- Change the name of the **servoMovement()** function to **openDoor()**.
- Call **openDoor()** instead of **servoMovement()** in the **loop()** function.
- The **openDoor()** function should first check the distance from the door (distance sensor). If the distance is less than 30cm, the function should open the door (set the servo value to 90 degrees) and change the “doorStatus” variable to “open”. If instead, the distance is more than 30 cm, the function should close the door (set the servo value to 0 degrees) and change the “doorStatus” variable to “close”.
- Modify the **get\_index()** function to add an appropriate message for the door status to your dashboard as shown below:

## The Smart Door Dashboard

Welcome to the smart door dashboard

The distance from the door is: 170 cm.

The door is **closed** based on the distance

## The Smart Door Dashboard

Welcome to the smart door dashboard

The distance from the door is: 3 cm.

The door is **open** based on the distance

# The Smart Door Automation

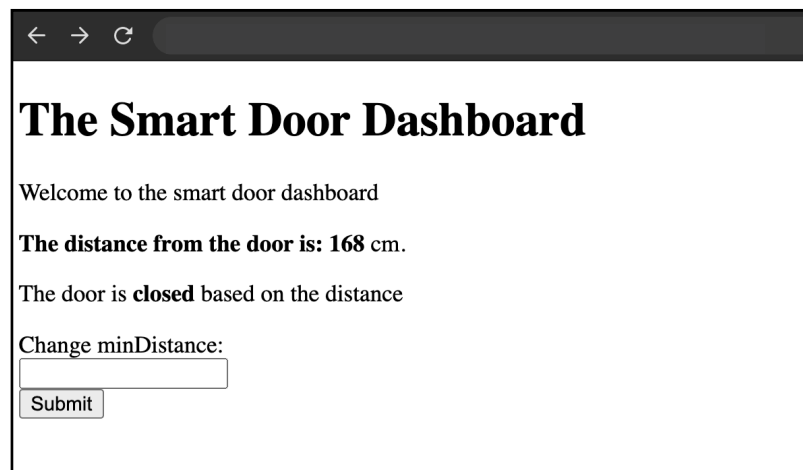
## Additional Tasks

---

The smart door does now open and close according to the distance recorded by the ultrasonic sensor. If you followed the above instructions, you should have the servo motor reacting to a distance lower than 30cm. You should also have an updated dashboard with the appropriate messages: “The door is closed based on the distance” or “The door is open based on the distance”.

The additional task for this exercise is for you to dynamically change the door opening distance threshold from the web dashboard. Instead of having a fixed minimum distance for the door to open (30cm), you should write the code to change its threshold value from the web server.

Go ahead and replicate the following dashboard:



The screenshot shows a web browser window with a dark header bar containing navigation icons (back, forward, refresh). The main content area has a title 'The Smart Door Dashboard' in bold. Below the title, it says 'Welcome to the smart door dashboard'. Then, it displays 'The distance from the door is: 168 cm.' followed by 'The door is **closed** based on the distance'. At the bottom, there is a section 'Change minDistance:' with a text input field and a 'Submit' button.

Write the code so that you can input and submit the new minimum distance for the door to open from the web server dashboard. The servo should then react to the new submitted value.

You are expected to attempt the following:

- Add a global int variable called **minDistance** to store the minimum distance value for the door to open. Use it inside the **openDoor()** function instead of a fixed value.

- Change the **get\_index()** function so that it has a form with an input field and a button. The form should send a get request to “/changeMinDistance” with the name and the value of the input field. The new route should open on a different page.
- Change the **get\_index()** function so that the dashboard refreshes every 10 seconds. This will allow you time to complete and submit the form.
- Create a function called **setMinDistance()** which parses the query string and assigns the value of the input field to the **minDistance** variable.
- Add the following request and callback to the **setup()** function:  
server.on("/changeMinDistance", setMinDistance);

After completing the additional task, you should be able to set the minimum distance threshold value for the door to open from the web server dashboard.

If you need any help, you can refer back to the smart fridge exercise where you wrote the code to control the buzzer from the web dashboard.