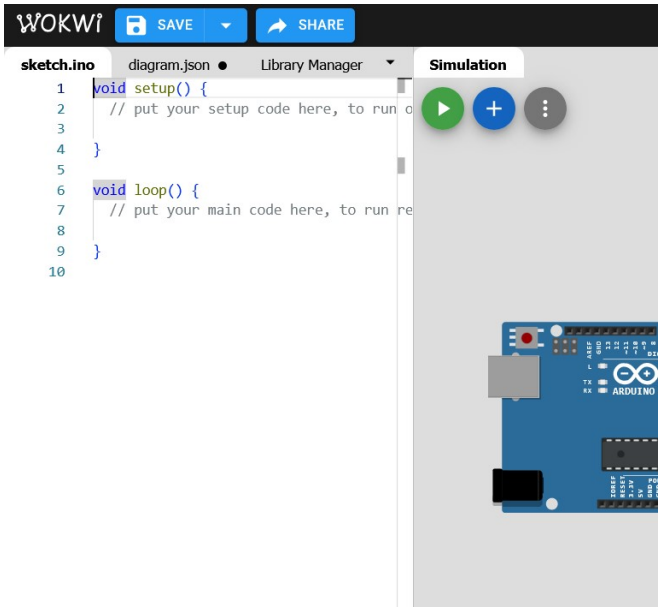
Power an LED by pushbutton with an Arduino uno r3 in Wokwi

First off go to https://wokwi.com/ and setup an account, after that is done you can begin a project.

- Click on “New Project at the top of the screen after you have logged in.
- Select the board/micro-processor you will be simulating in the left column
- I am choosing Arduino because it matches the physical board that I have
- Now select the type of Arduino board (I am choosing the Uno Rev3)

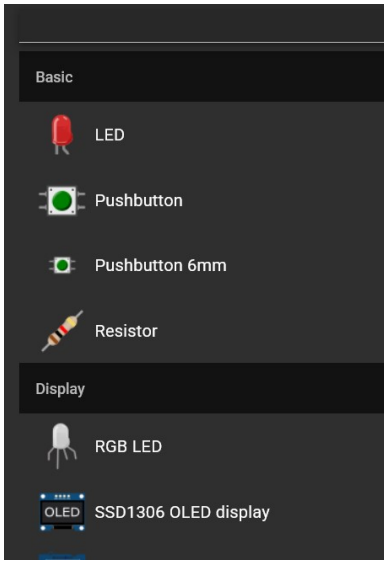
Once selected you are taken to a splitscreen. In the left panel is the code that you will write to your micro-controller board. The panel on the right is the simulation program where you can add components and connect them to your board and test out ideas and projects. I will be making a device that will power on an LED by the push & hold of a pushbutton.

In the below example is a default code/simulator screen with no code or components added.

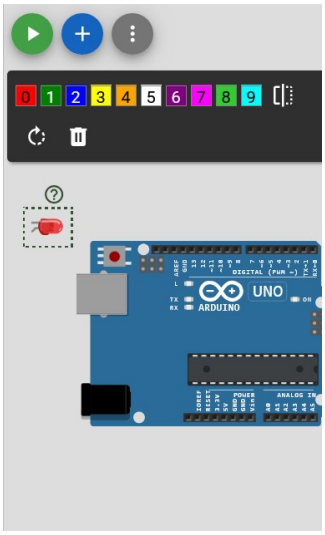


I will add a few components to the simulation screen and demonstrate how To power on an LED by the push of a button. The LED will turn off when the ‘Pushbutton’ is no longer pressed. The LED light only stays on when consistent contact is made to the push button. It must be held down in order for the LED light to stay powered on.

In the simulation portion of the window, click on the " + " button, to bring up the Component selection menu. From here I will select the components that I need to complete this project.



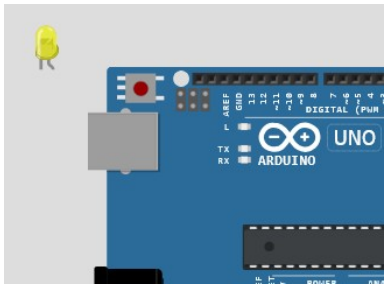
The first component to select is the LED, because it is the component that we want to activate. The other components that we will need are resistors and a Pushbutton. We will get the larger pushbutton from the menu screen. Once an item is selected it will instantly appear in the simulation window. If you left click on it, then an few extra options become available in terms of what you can do with the item that you selected. You can change the color as well as the directional orientation of the component, in order to better facilitate your project.



In image above you can see an LED turned 90 degrees to the right as well as a number color

selection panel which reveals the available colors that you can use to modify the look of

the component that you are using.



In the above image we turned the natural color of the LED yellow and adjusted is

Positional orientation to the default position. In future examples I will return the

color to red as well as add other needed components. So I selected 2 resitors,

and adjust the positional orientation to one of them at 90 degrees so that the

brown line is facing north and the yellow line on the resistor is facing south.

Now I will connect the arduino board to the LED:

- Click on the #13 digital output slot on the board and clicking again on the end of resistorA. This bridges a connection to the board and one end of resistorA
- Connect the empty side of resistor to the right wire/side of the LED
- Click on the GND (ground) port on the board and connect it to the unused (left side) wire of the LED
- Click on the resistor that we used and look at the to following columns..Value & Units…
- In this menu click and change the value to 500 and the unit to (Ω) Ohms

Now we will write the code that will power on the LED

- We must tell the micro-controller that #13 is an output pin
- Under the setup section of the code: add the following

pinmode()

- Inside the parenthesis I will add the pin number and it's assignment

pinmode(13, OUTPUT)

- In the loop section of the code I will add

digitalWrite()

- I will add values inside the parenthesis

digitalWrite(13, High)

<u>This code will turn on the LED when the simulator is ran:</u>

```
1. void setup() {
2. pinmode(13, OUTPPUT);
3. }
4. void loop() {
5. digitalWrite(13, HIGH)
6. }
```

<u>I will modify this code to get the LED to blink by adding a LOW/off feature and a 1 second delay:</u>

```
1. void setup() {
2. pinmode(13, OUTPPUT);
3. }
4. void loop() {
5. digitalWrite(13, HIGH)
6. delay(1000)
7. digitalWrite(13, LOW)
8. delay(1000)
9. }
```

- I am telling the micro-controller that pin 13 is my output pin and from that pin

I would like to luminate an LED for a second then turn it off continuously until

I tell the micro-controller otherwise.

<u>I will now push the 'G' button on my keyboard to activate the grid/breadboard beneath the board</u>

- The next section I will describe in detail how to convert the current project to a

project where I will use a push button mechanism to power on an LED.

- Now to add a second resistor to this project and call it resistorB and I will also

add a pushbutton component from the component library by click the ' + ' button on

the simulation side of the screen. I will now move this component closer to the board

to make wiring a lot easier. I added a digitalRead line of code to tell the micro-controller

read the status of digital OUTPUT 8 and pass it to the digitalWrite statement .

- The previous code from the loop section was removed to allow for the new actions

to take place. I will make connections (add wires) from the board to the pushbutton

so that the LED will be power on when the pushbutton is held down. The LED will

power off when the button is released.

- Create a wire from the 5v port on the board to the lower right connection on the pushButton

Next **I will clear the code in the loop section, but keep the code the same in the setup section:**

```
1. void setup() {
2. pinmode(13, OUTPPUT);
3. }
4. void loop() {
5. digitalWrite(13, digitalRead(8));
6. }
```

- I will make connections (add wires) from the board to the pushbutton so that the LED will be power on when the pushbutton is held down. The LED will power off when the button is released.

- Create a wired connection from 5v port to the upper right port of the pushbutton
- Create a wired connection from the digital 8 OUTPUT port to the lower right port of the pushbutton

Now that everything is wired correctly in the simulation and the code is written and saved as shown in the image below,

When I run the simulation, I should be rewarded by seeing the LED power on when I hold the power button. This same LED should also power off, the moment I release the pushbutton in the simulation.