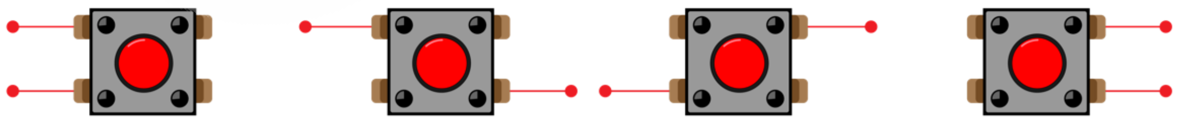
Topic	BOUNCING & DEBOUNCING SWITCH	
Class Description	Students will be introduced to the concept of bouncing and debouncing, using a concept they will operate relay switch using a push-button	
Class	**PRO C253**	
Class time	**50 mins**	
Goal	● Bouncing ● Debouncing	
Resources Required	● Teacher Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ○ Smartphone ● Student Resources: ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	**Warm-Up** **Student-Led Activity -1** **Student-Led Activity -1** **Wrap-Up**	**10 mins** **15 mins** **15 mins** **10 mins**
Credit & Permissions:	**Code samples used for Firebase-Google Authentication are licensed under the Apache 2.0 License.** **Expo documentation used from - https://expo.io** **Note: Keep this row section only if applicable**	

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

WARM-UP SESSION - 10 mins	
Teacher Action	**Student Action**
Hey <student's name>. How are you? It's great to see you! Are you excited to learn something new today? **Following are the WARM-UP session deliverables:** ● Greet the student. ● Revision of previous class activities. ● Quizzes.	**ESR**: Hi, thanks! Yes, I am excited about it! Click on the slide show tab and present the slides
WARM-UP QUIZ Click on In-Class Quiz	
Activity Details **Following are the session deliverables:** ● Appreciate the student. ● Narrate the story by using hand gestures and voice modulation methods to bring in more interest in students.	
STUDENT-LED ACTIVITY-1 - 15mins	
Student Initiates Screen Share	
ACTIVITY ● **Introduction to Bouncing**	
Teacher Action	**Student Action**
Note: This class will be a student-driven class, the teacher will guide the student to complete the activity. But before that teacher needs to do this activity on her end too. Have you ever seen flickering lights/LEDs?	ESR Varied!

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Did you notice when you operate your LED with push-button, How would be the behavior? What do you understand by Push Button? **Push Button:** The push-button is used to control devices like turning on and off circuits or electronics devices.	
A push-button usually has four pins that are connected internally in pairs. We only need to use two of the four pins, which are NOT in the same connected pair. Accordingly, there are four ways to do wiring with the button.	
	

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

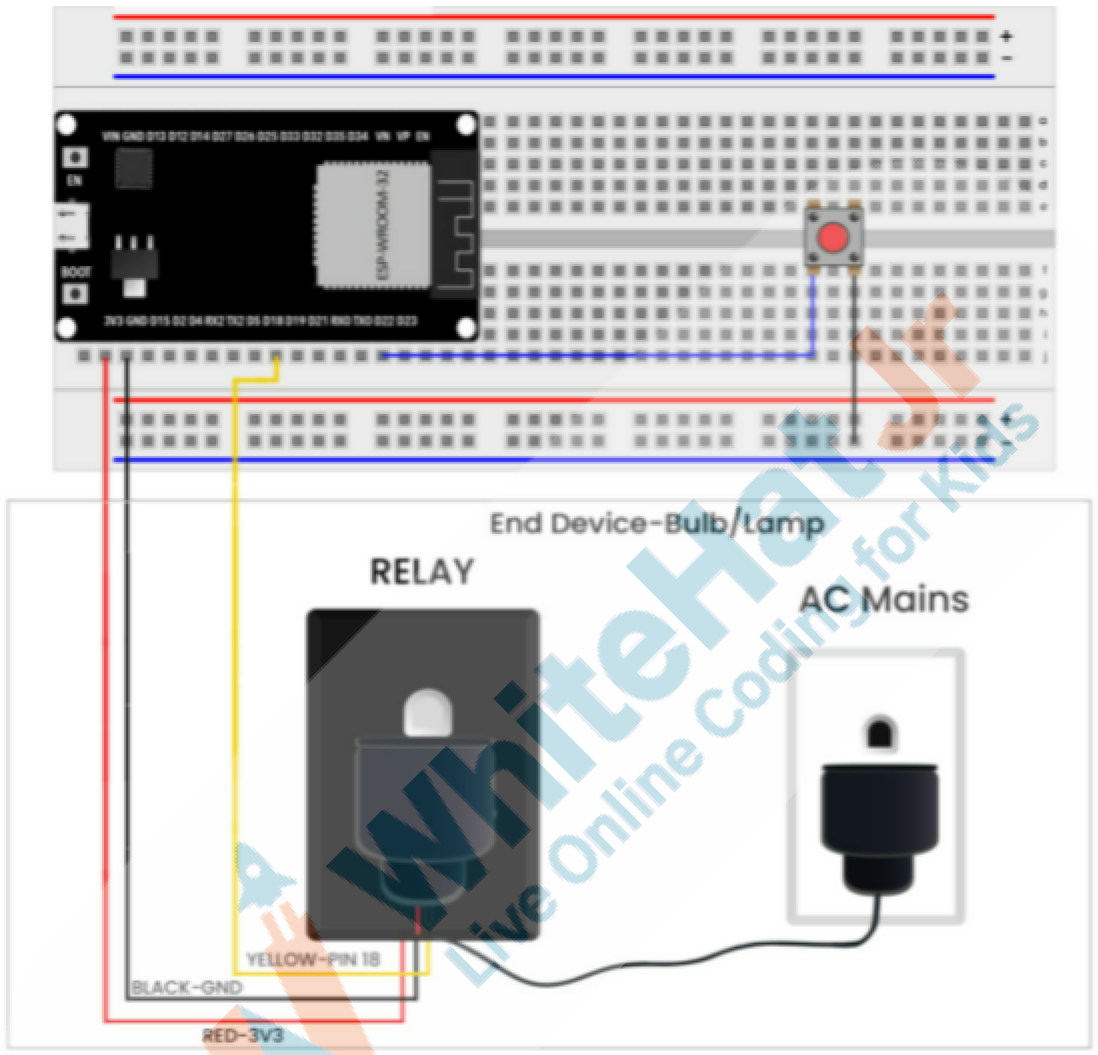
You must be wondering why we suddenly discussed the push button after a long time. Ok, when you have done the push button you might have seen chattering behavior or multiple triggering. Let's understand this: When we press a pushbutton, toggle switch, or micro switch, two metal parts come into contact, shorting the supply. There is no instant connection, but the metal parts are connected and disconnected several times before a stable connection is made. The same thing happens when the button is released. This results in false triggering or multiple triggering or chattering behavior like the button are pressed multiple times but in fact, you pressed just a single time. It's the same situation as falling a ball from a height, and the ball keeps bouncing until it comes to a rest.	ESR: Varied!
Today our task is to control a relay switch with a push-button and understand bouncing and debouncing circuits in electronics. We are using a relay because it will give a clear picture of on and off devices.	
Step -1:Gather the material from the IoT kit: ● 1 x ESP32 ● 1 x USB Cable ● 1 x Breadboard ● 4 x Jumper wires ● 1 x Push Button ● 1 x Relay ● 1 x Mosquito Repellant Machine/Lamp/Bulb with holder	
Step -2: Let's do connections:	

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

● Insert pushbutton on the breadboard ● Connect one end of the pushbutton with **ESP32 GPIO pin no 22** ● Connect another end of the pushbutton with GND of the ESP32 Take the Relay(Black Box), Insert the relay Plug into **AC mains**. ● Connect relay with ESP32 BOARD ● Connect **Black with GND, Red with 3.3V**, and **Yellow with ESP32 GPIO PIN 18** ● Take one device like **Mosquito Repellant Machine/Lamp/Bulb with holder and insert them into relay switch.** We are done with our connections.	

WhiteHat Jr Live Online Coding for Kids

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.



Let's write the program:

Define the setup function

1. Define GPIO pin for pushbutton i.e **PUSHBUTTON_PIN 22**
2. Define GPIO pin for relay i.e **RELAY_PIN** 18

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

`#define BUTTON_PIN 22`
`#define RELAY_PIN  18`	

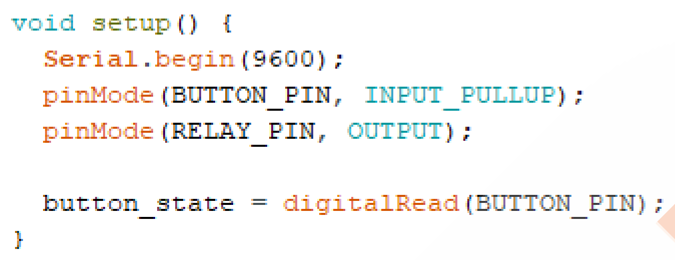
Define the datatypes for button_states, ● Int is used for integer values, ● declare **int** for **relay_state, button_state, last_button_state** ● **Last_button_state** will be store the last value of **push_buton**, **button_state** will store the current value of the button, **relay_state** will store the value of relay.	

`int relay_state = LOW;`
`int button_state;`
`int last_button_state;`	

Initialize using **void setup()** function ● **Serial. begin(9600)** is used for data exchange speed. speed parameters. This tells the Arduino to get ready to exchange messages with the Serial Monitor at a data rate of 9600 bits per second. That's 9600 binary ones or zeros per second and is commonly called a baud rate. ● **pinMode()** configures the specified pin to behave either as input or output. Since we want this pin for output, we are writing **OUTPUT** here. ● **Syntax**: *pinMode(pin, mode)* ● **pin**: The pin do we need to set ● **mode**: Set the mode INPUT, OUTPUT, INPUT_PULLUP, INPUT_PULLDOWN, ● In electronic circuits, a pull-up resistor or pull-down resistor is a resistor used to ensure a known state for a signal.**PULLUP** condition for push-button will ensure the state on the pin is **HIGH**	

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

- **PULLDOWN** condition for push button will ensure the state on the pin is **LOW**



```
void setup() {
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  pinMode(RELAY_PIN, OUTPUT);

  button_state = digitalRead(BUTTON_PIN);
}
```

Write the logic part under **void loop()**

Variable last_button_state will store the current value of push-button state

digitalRead() will check the state of button

Serial.println is used to print the statement

The pin needs to be programmed to be either ON or OFF, that is, we can command it to be ON (output 5 volts), or OFF (output 0 volts).

To switch it on and off, we need to use a function called **digitalWrite().**

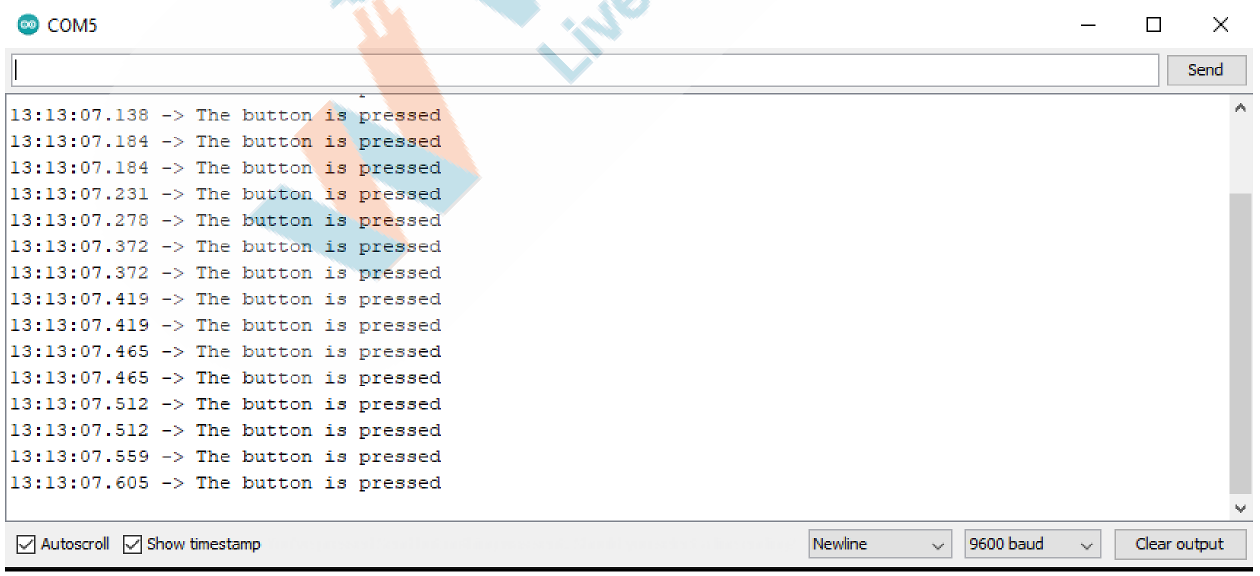
© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

```
void loop() {
  last_button_state = button_state;
  button_state = digitalRead(BUTTON_PIN);

  if (last_button_state == HIGH && button_state == LOW) {
    Serial.println("The button is pressed");

    // toggle state of relay
    relay_state = !relay_state;

    // control relay arccoding to the toggled state
    digitalWrite(RELAY_PIN, relay_state);
  }
}
```

Output:
Compile and upload the program to ESP32 board using Arduino IDE

- Verify the program by clicking the Tick option
- Upload the program by clicking the arrow option

Note: If the port is not selected, insert the USB cable in Computer's port and select the port.
Make sure hardware is connected properly.



```
13:13:07.138 -> The button is pressed
13:13:07.184 -> The button is pressed
13:13:07.184 -> The button is pressed
13:13:07.231 -> The button is pressed
13:13:07.278 -> The button is pressed
13:13:07.372 -> The button is pressed
13:13:07.372 -> The button is pressed
13:13:07.419 -> The button is pressed
13:13:07.419 -> The button is pressed
13:13:07.465 -> The button is pressed
13:13:07.465 -> The button is pressed
13:13:07.512 -> The button is pressed
13:13:07.512 -> The button is pressed
13:13:07.559 -> The button is pressed
13:13:07.605 -> The button is pressed
```

☑ Autoscroll ☑ Show timestamp   Newline   9600 baud   Clear output

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

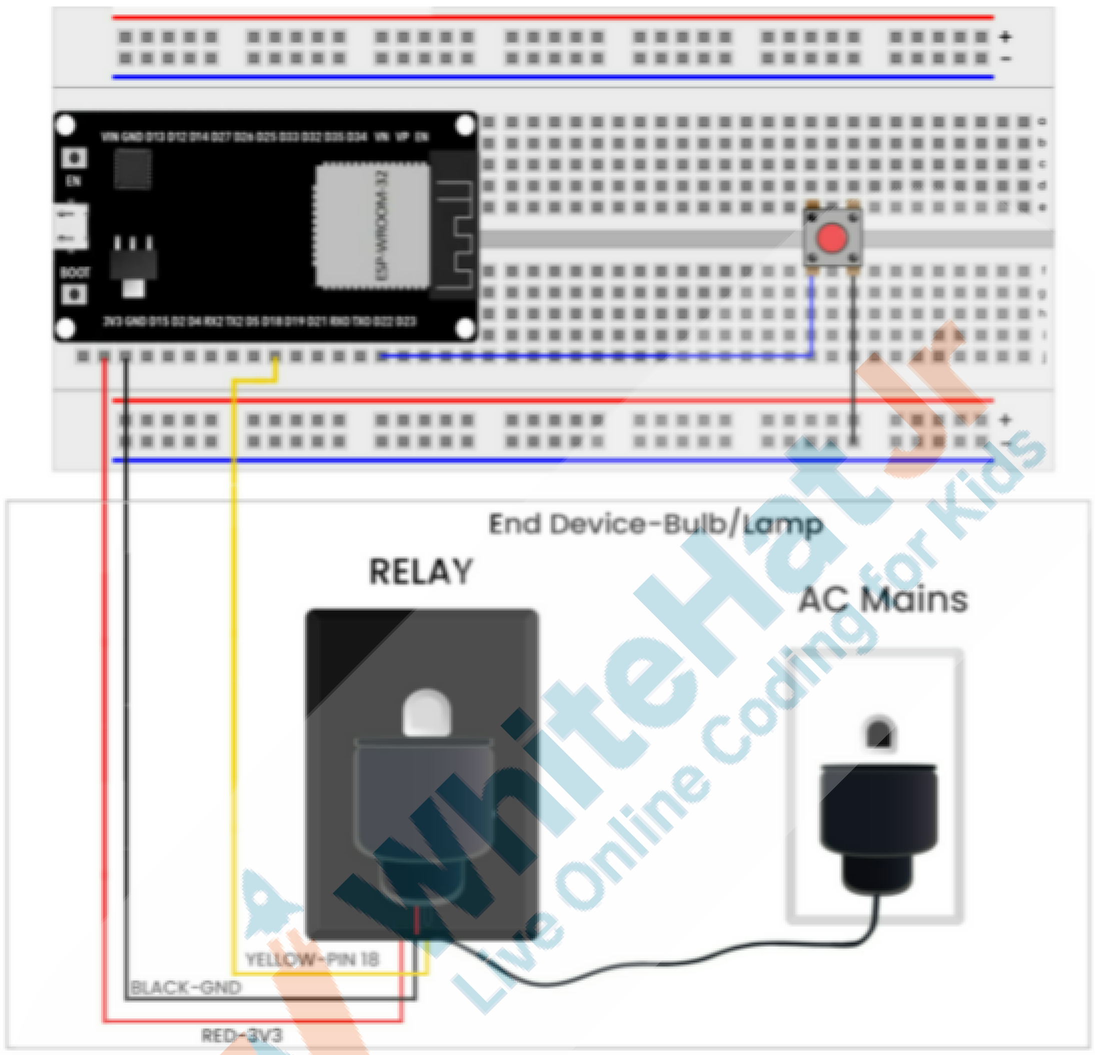
Press the push button once and keep it for several seconds and then release it and you'll see the light flickering of your connected device. You will see that you pressed the button once but your lamp/mosquito repellent LED will flicker on and off multiple times. It is called **BOUNCING** since it shows multiple stages of 0 and 1. Ideally, it should turn on and off once when a button is pressed. Consequently, it will give false signals. False signals cause a lot of problems with electronics circuits. To rectify this we must learn about **Debouncing** circuits	
Student Stops Screen Share	
We have Debouncing circuits challenge for you. Can you solve it? Let's try. I will guide you through it.	
STUDENT-LED ACTIVITY-2 - 15 mins	
• Ask the student to press the ESC key to come back to the panel. • Guide the student to start Screen Share. • The teacher gets into Full Screen.	
Student Initiates Screen Share	
ACTIVITY • Debouncing circuits	
Teacher Action	Student Action

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

So we must work on how to stop flickering. You must be surprised to know this, this type of problem usually occurs when we use buttons, especially when we run for the first time. To remove this type of issue we have a pre-defined library called **ezButton.**This library is usually used with pushbuttons and various Switches. Connections will remain the same as previous circuits.	

WhiteHat Jr Live Online Coding for Kids

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.



Define Pins

1. Define GPIO pin for pushbutton i.e **PUSHBUTTON_PIN 22**
2. Define GPIO pin for relay i.e **RELAY_PIN** 18

```
#include <ezButton.h>

#define BUTTON_PIN 22
#define RELAY_PIN  18
```

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

Define the datatypes for button_states,

- Int is used for integer values,
- declare **int** for **relay_state** and store the value **LOW**
- Create **ezbutton** object **button**

```
ezButton button(BUTTON_PIN);
int relay_state = LOW;
```

Initialize using **void setup()** function

- **Serial. begin(9600)** is used for data exchange speed. speed parameters. This tells the Arduino to get ready to exchange messages with the Serial Monitor at a data rate of 9600 bits per second. That's 9600 binary ones or zeros per second and is commonly called a baud rate.

- **pinMode()** configures the specified pin to behave either as input or output. Since we want this pin for output, we are writing **OUTPUT** here.

- **Syntax**: *pinMode(pin, mode)*
- **pin**: The pin do we need to set
- **mode**: Set the mode INPUT, OUTPUT,
- **setDebounceTime** of 50 ms

```
void setup() {
  Serial.begin(9600);
  pinMode(RELAY_PIN, OUTPUT);
  button.setDebounceTime(50);
}
```

Write the logic in **void loop()**

Call the loop function first

Variable **last_button_state** will store the current value of

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

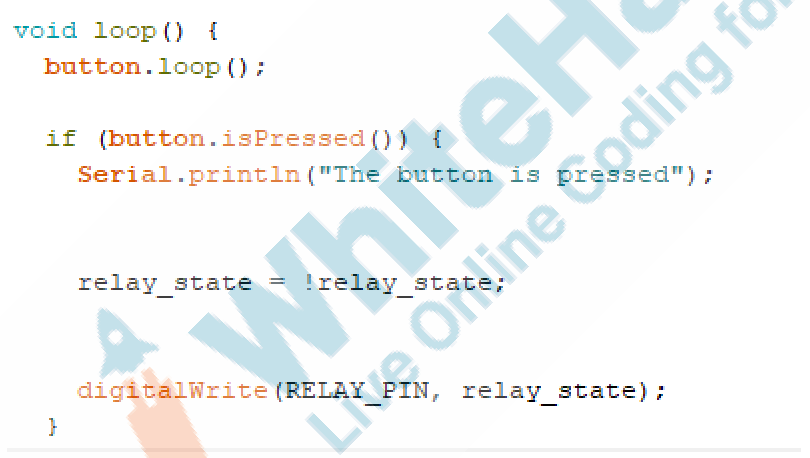
push button state

Serial.println is used to print the statement

! is used to toggle the state of the relay in case of flickering

The pin needs to be programmed to be either ON or OFF, that is, we can command it to be ON (output 5 volts), or OFF (output 0 volts).

To switch it on and off, we need to use a function called **digitalWrite().**



```
void loop() {
  button.loop();

  if (button.isPressed()) {
    Serial.println("The button is pressed");


    relay_state = !relay_state;


    digitalWrite(RELAY_PIN, relay_state);
  }
```

Output:
Compile and upload the program to ESP32 board using Arduino IDE

- Verify the program by clicking the Tick option
- Upload the program by clicking the arrow option

Note: If the port is not selected, insert the USB cable in Computer’s port and select the port.
Make sure hardware is connected properly.

Go to Tools and select **Serial Monitor**

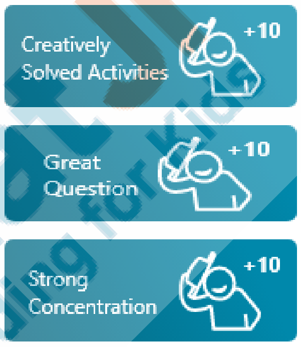
Press the push button once and keep it several seconds

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

and then release it and you'll see there is no light flickering. You will see that you pressed the button once in a result your lamp/mosquito repellent LED will turn on and off once only instead of multiple times It is called **DEBOUNCING** since it shows one stage of 0 and 1. This is the Ideal solution for the circuit.	
So, today we learned about bouncing & debouncing circuits	
Teacher Guides Student to Stop Screen Share	
WRAP-UP SESSION - 05 mins	
Activity details **Following are the WRAP-UP session deliverables:** ● Appreciate the student. ● Revise the current class activities. ● Discuss the quizzes.	
WRAP-UP QUIZ Click on In-Class Quiz	
Activity Details **Following are the session deliverables:** ● Explain the facts and trivia ● Next class challenge ● Project for the day ● Additional Activity (Optional)	
<u>FEEDBACK</u> ● **Appreciate and compliment the student for trying to learn a difficult concept.**	

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

● **Get to know how they are feeling after the session.** ● **Review and check their understanding.**

Teacher Action	**Student Action**
You get “hats-off” for your excellent work! In the next class, we will learn about web servers	*Make sure you have given at least 2 hats-off during the class for:*  Creatively Solved Activities +10 Great Question +10 Strong Concentration +10

PROJECT OVERVIEW DISCUSSION Refer the document below in Activity Links Sections
Teacher Clicks
ADDITIONAL ACTIVITIES (Optional)

Additional Activities	

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Student Activity 1	Reference Code -Bouncing	https://github.com/procodingclass/PRO-C253-Student-Activity-1
Student Activity 2	Reference Code -Debouncing	https://github.com/procodingclass/PRO-C253-Student-Activity-2
Teacher Reference 1	Project	https://s3-whjr-curriculum-uploads.whjr.online/7c70dace-2f9b-4921-8318-ca15733a6617.docx
Teacher Reference 2	Project Solution	https://github.com/procodingclass/PRO-C253-Project-Solution
Teacher Reference 4	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/c533855f-f224-4454-b97f-5c51e5ccce2c.docx

© 2021 - WhiteHat Education Technology Private Limited.
Note: This document is the original copyright of WhiteHat Education Technology Private Limited.
Please don't share, download or copy this file without permission.