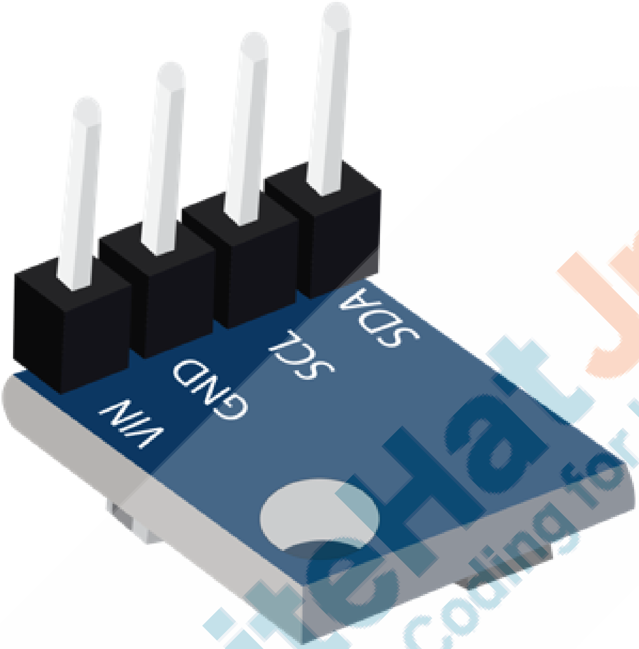
Topic	**MONITORING SYSTEM - 1**	
Class Description	**Students will be introduced to the BMP180 pressure sensor and how to interface BMP180 with ESP32 and how to design a cloud server on Adafruit.**	
Class	**PRO C248**	
Class time	**45 mins**	
Goal	● Introduction to BMP180 sensor ● I2C Communication ● Introduction to Adafruit cloud server	
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** **Teacher-Led Activity** **Student-Led Activity** **Wrap-Up**	**10 mins** **15 mins** **20 mins** **5 mins**
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.	
TEACHER-LED ACTIVITY - 10 mins	
Teacher Initiates Screen Share	
ACTIVITY ● **Introduction to Adafruit** ● **Introduction to libraries**	
Teacher Action	**Student Action**
Have you heard about atmospheric pressure? Atmospheric pressure is the pressure on earth that is caused by the weight of the air above us. Atmospheric pressure is also known as barometric pressure. What do you think about how we can measure	**ESR:** Yes! **ESR:** Sensors!

atmospheric pressure? Yes, Sensors! Atmospheric pressure sensors measure the absolute pressure of the air around us. This pressure varies with both the weather and altitude and according to that, we can predict weather conditions. Don’t you think it would be fun to measure live atmospheric pressure? To measure atmospheric pressure we have a **BMP180** sensor. Open your IoT kit and check the **BMP180 sensor**	



BMP180 Pin configuration :

VCC: Connected to +5V
GND: Connected to GND
SCL: Serial Clock pin
SDA: Serial Data pin (I2C interface)

SCL & SDA are used to communicate with the ESP32 module. The data is sent to the ESP32 or received from the ESP32 using these two pins.

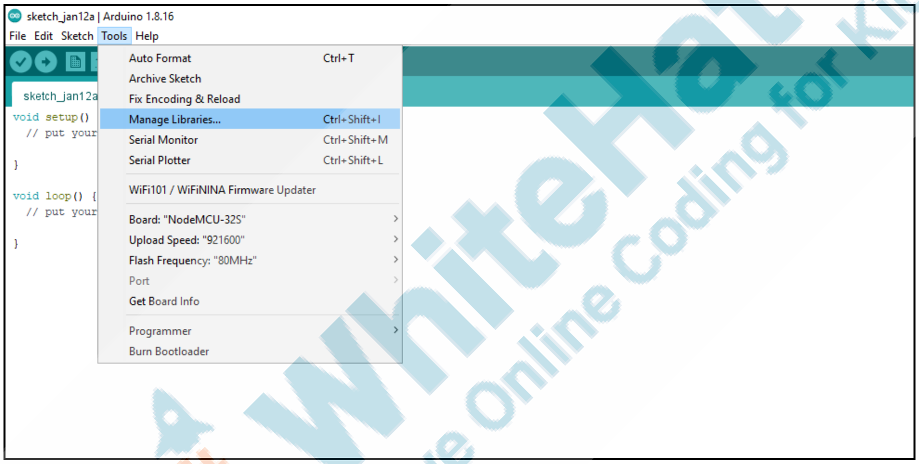
To communicate data to and fro we use the I2C communication protocol.

I2C communication Protocol

I2C communication is the short form for inter-integrated circuits. Using just two common wires, I2C allows data to be transferred between a central processor (ESP32) and several ICs on one circuit board.

Before using the **BMP180 sensor,** we need to install libraries first.

Open Arduino IDE, Go to **Tools,** and then **Manage Libraries**



```
void setup()
  // put your

}

void loop() {
  // put your

}
```

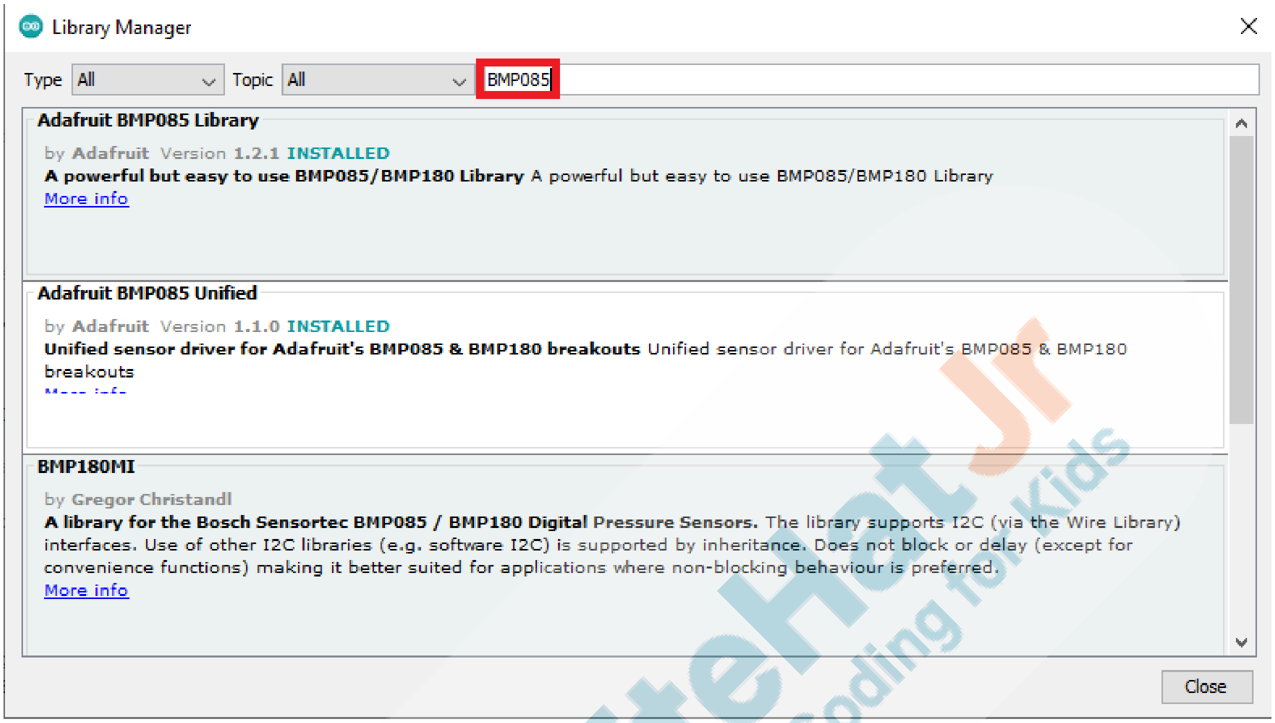
Type **BMP085** and then click on Install.

After Installing sensor library below window will appear:

Note:When running any Arduino program, if a library error appears, follow the same instructions to install libraries.

Install the same library on the student side too.

Install BMP180
Install MQTT library



Library Manager

Type All Topic All BMP085

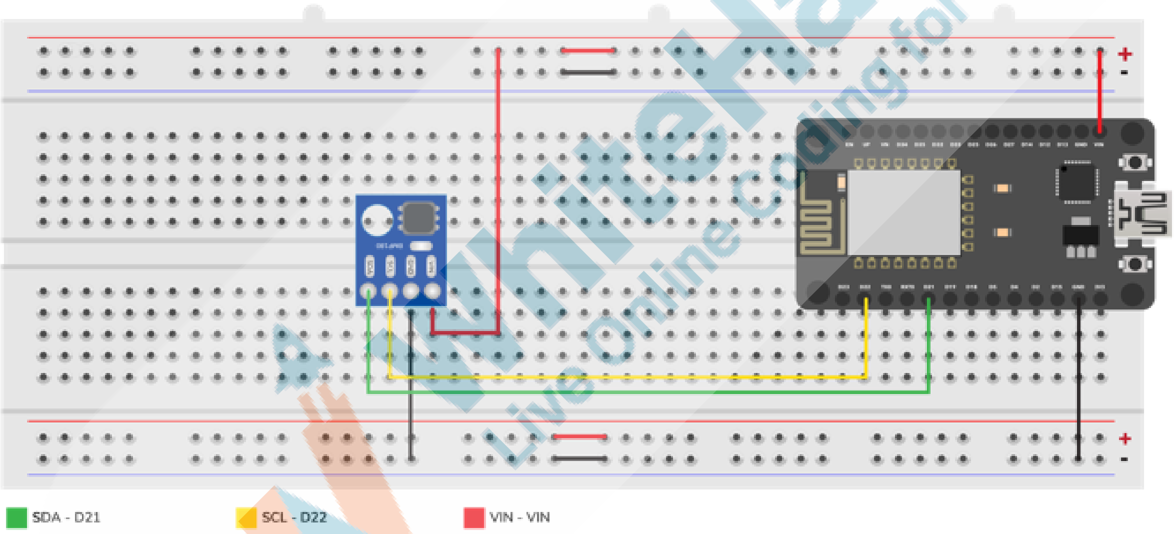
Adafruit BMP085 Library
by Adafruit Version 1.2.1 INSTALLED
A powerful but easy to use BMP085/BMP180 Library A powerful but easy to use BMP085/BMP180 Library
More info

Adafruit BMP085 Unified
by Adafruit Version 1.1.0 INSTALLED
Unified sensor driver for Adafruit's BMP085 & BMP180 breakouts Unified sensor driver for Adafruit's BMP085 & BMP180 breakouts
More info

BMP180MI
by Gregor Christandl
A library for the Bosch Sensortec BMP085 / BMP180 Digital Pressure Sensors. The library supports I2C (via the Wire Library) interfaces. Use of other I2C libraries (e.g. software I2C) is supported by inheritance. Does not block or delay (except for convenience functions) making it better suited for applications where non-blocking behaviour is preferred.
More info

Note : The teacher will guide the student in making connections: **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 BMP180	The student will collect the material from the IoT kit
Step -2: Let's do connections: *Note: Follow the same pins number as mentioned below*	*The student will do the connections as per teacher guidance*

BMP180 Pins	Wiring Connections
VCC	Connect with **3V3 PIN** of the ESP32
GND	Connect with **GND** of the ESP32
SCL	Connect with GPIO **PIN 22**
SDA	Connect with GPIO **PIN 21**



Teacher Stops Screen Share

So now it's your turn to write a program for barometric sensors.
Please share your screen with me.

We have one more class challenge for you.
Can you solve it?

Let's try. I will guide you through it.

STUDENT-LED ACTIVITY-1 - 10 mins

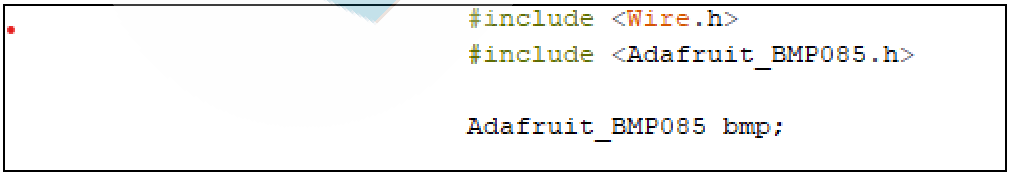
- Ask the student to press the ESC key to come back to the panel.
- Guide the student to start Screen Share.
- The teacher gets into Full Screen.

Student Initiates Screen Share

ACTIVITY

- Read Temperature
- Read Pressure
- Setup of Cloud server

Teacher Action	Student Action
Step-3 Let's write a code: Define the libraries ● **Wire.h library** is used to communicate with I2C devices. ● **Adafruit_BMP085.h** library is used for **pressure sensors.** *Note:BMP085/BMP180 both sensors can used this library* ● Create object **bmp** for **Adafruit_BMP085**	

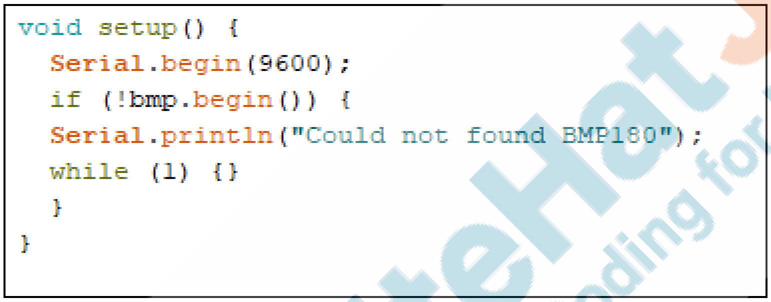


```
#include <Wire.h>
#include <Adafruit_BMP085.h>

Adafruit_BMP085 bmp;
```

Teacher Action	Student Action
Initialize the setup() ● **Serial. begin(9600)** is used for data exchange	

speed. 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.

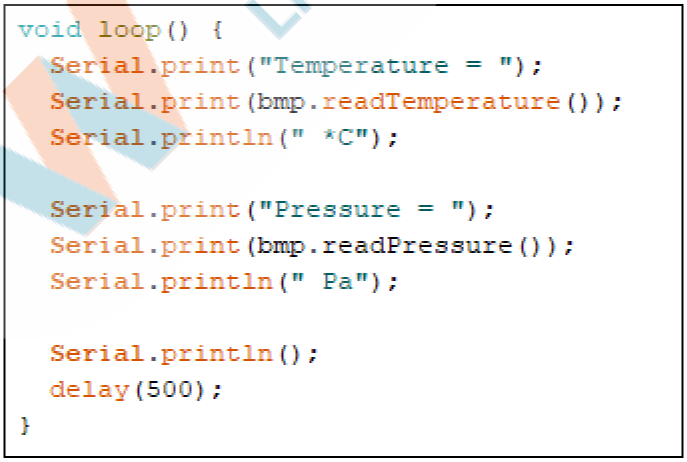
- **bmp.begin()** is used to begin the process.
- **Serial.println is** used to print data. Print (“Could **not found”,** if it fails to begin the process)



```
void setup() {
  Serial.begin(9600);
  if (!bmp.begin()) {
  Serial.println("Could not found BMP180");
  while (1) {}
  }
}
```

To execute the main process write the **void loop()**

- **Serial.print is** used to print data
- **readTemperature()** will read the temperature value.
- **readPressure()** will read the pressure value.
- Set the **delay** of **500 ms**



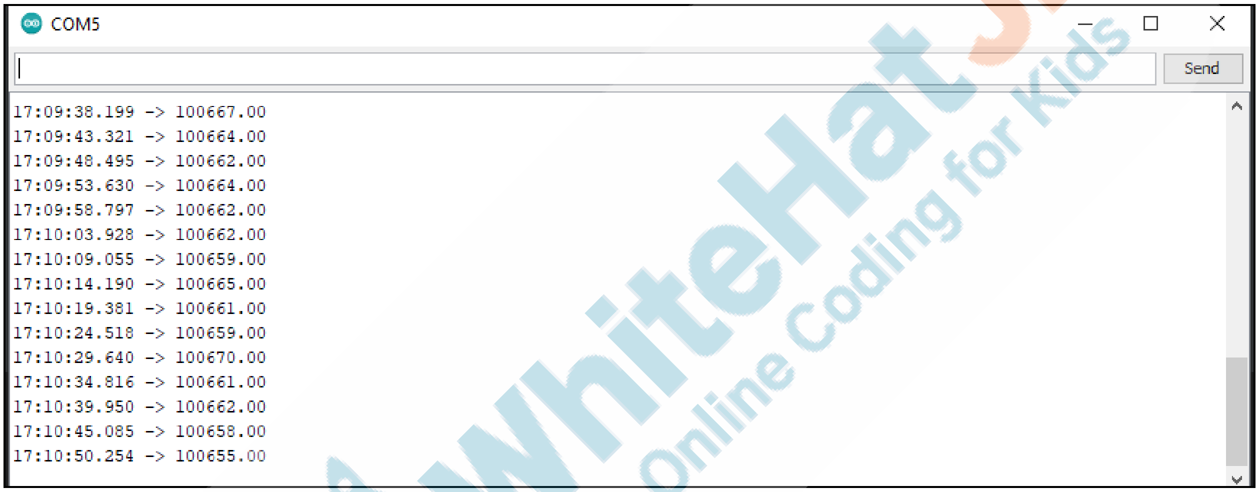
```
void loop() {
  Serial.print("Temperature = ");
  Serial.print(bmp.readTemperature());
  Serial.println(" *C");

  Serial.print("Pressure = ");
  Serial.print(bmp.readPressure());
  Serial.println(" Pa");

  Serial.println();
  delay(500);
}
```

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

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

Note: If the port is not selected, insert the USB cable in Computer’s port and select the port

- Go to Tools and select **Serial Monitor**
    - Pressure values will be displayed in Pascal. Pascal is a unit used to measure Pressure



```
17:09:38.199 -> 100667.00
17:09:43.321 -> 100664.00
17:09:48.495 -> 100662.00
17:09:53.630 -> 100664.00
17:09:58.797 -> 100662.00
17:10:03.928 -> 100662.00
17:10:09.055 -> 100659.00
17:10:14.190 -> 100665.00
17:10:19.381 -> 100661.00
17:10:24.518 -> 100659.00
17:10:29.640 -> 100670.00
17:10:34.816 -> 100661.00
17:10:39.950 -> 100662.00
17:10:45.085 -> 100658.00
17:10:50.254 -> 100655.00
```

So we see how the BMP180 sensor works, but still, a lot to do. We want this data to be sent on a cloud server.

What do you think it's possible?

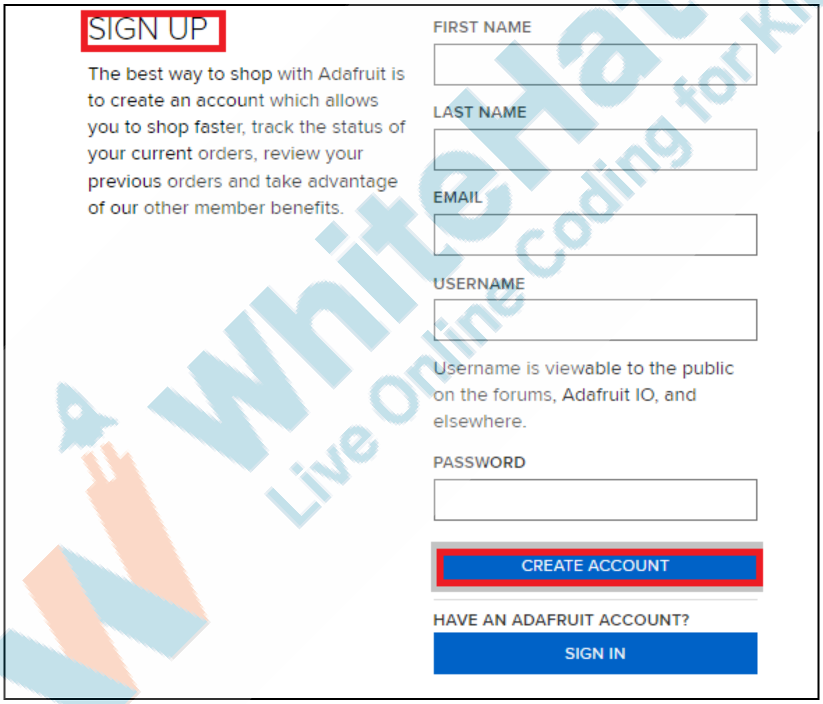
Yes, it is possible

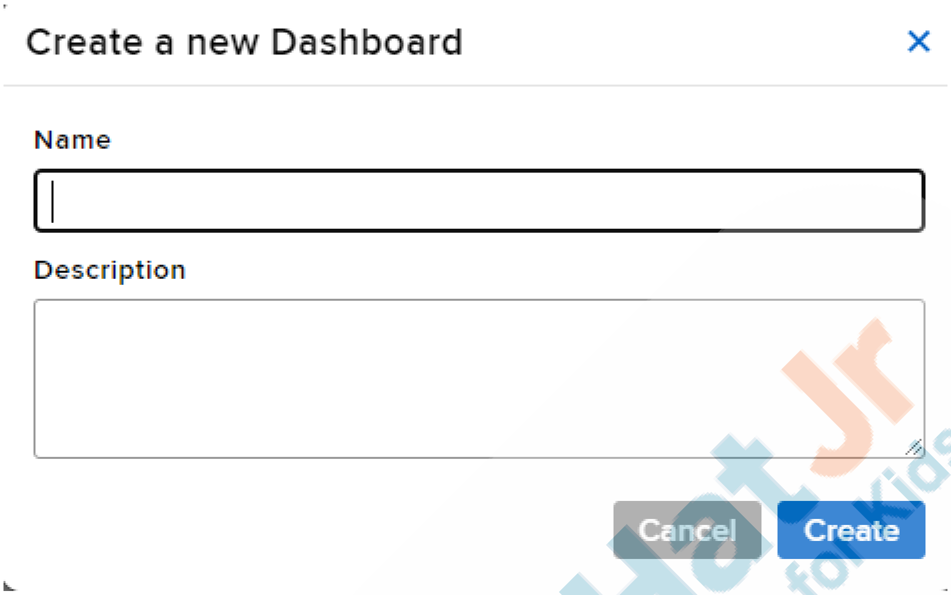
Let’s see how we can send BMP180 sensor data on the server.

We want to send data on a cloud server, for that we need to use an online server and to access an online server we need to use the platform Adafruit.

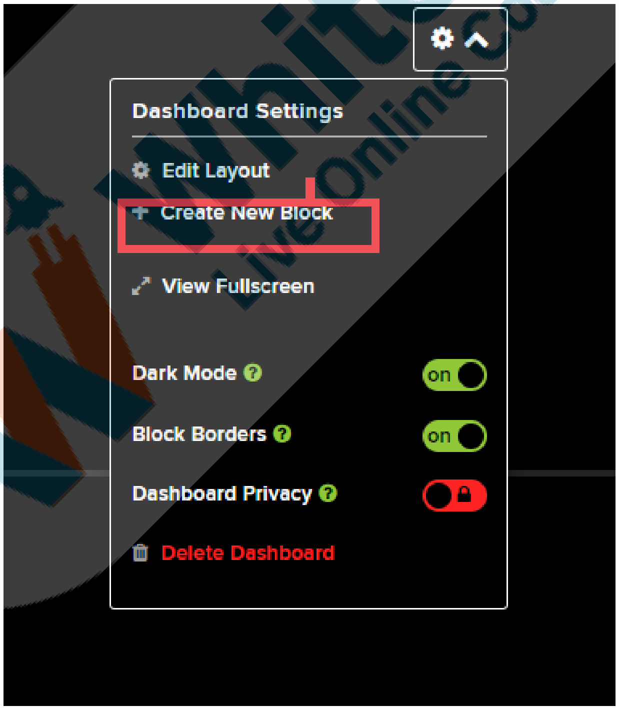
Adafruit will act as a broker between your device and

ESR: Varied!

server. Basically Adafruit is a neutral party that your Things can connect to send and receive messages. Let's set up an online server	
Teacher Clicks on ***Teacher Activity 2***	*Student Clicks on* ***Student Activity 2***
Click on **Sign Up** - Add your **Sign-in** details - Click on **Create Account**	
 SIGN UP The best way to shop with Adafruit is to create an account which allows you to shop faster, track the status of your current orders, review your previous orders and take advantage of our other member benefits. FIRST NAME LAST NAME EMAIL USERNAME Username is viewable to the public on the forums, Adafruit IO, and elsewhere. PASSWORD CREATE ACCOUNT HAVE AN ADAFRUIT ACCOUNT? SIGN IN	
- Click on **IO** - Go to **Dashboards** - Click on **New Dashboard** - Write the **Name (Environment Monitor System)** - Write **Description** if needed	



Click on **Create New Block**

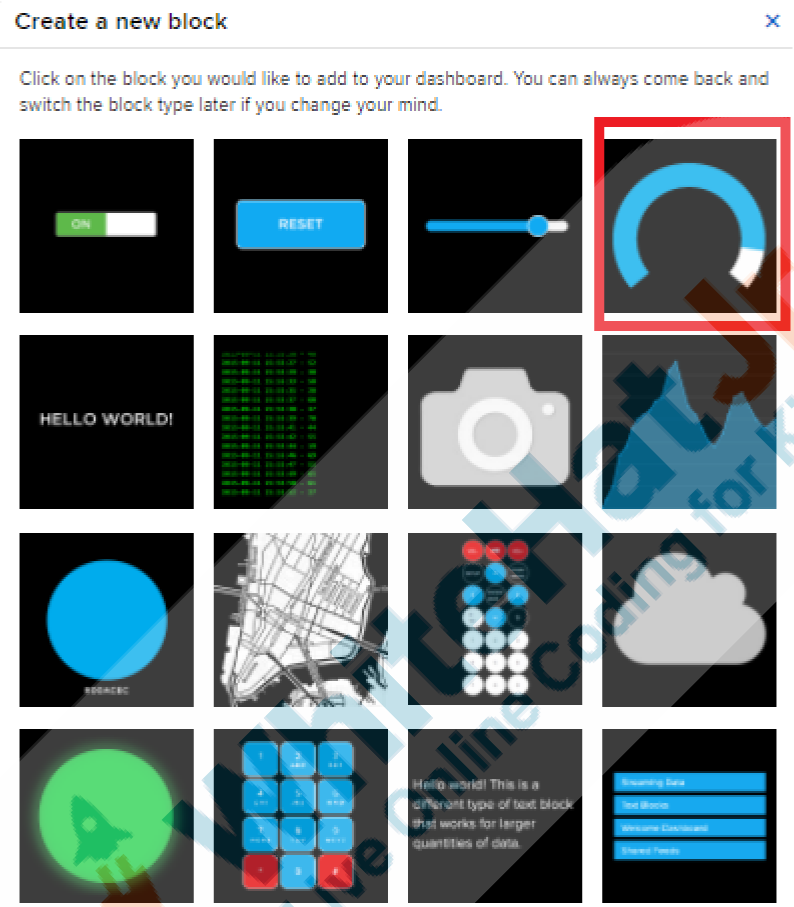


Dark Mode on

Block Borders on

Dashboard Privacy

Select the Gauge: Gauges are visual blocks to represent sensor values



Create a new block

Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.

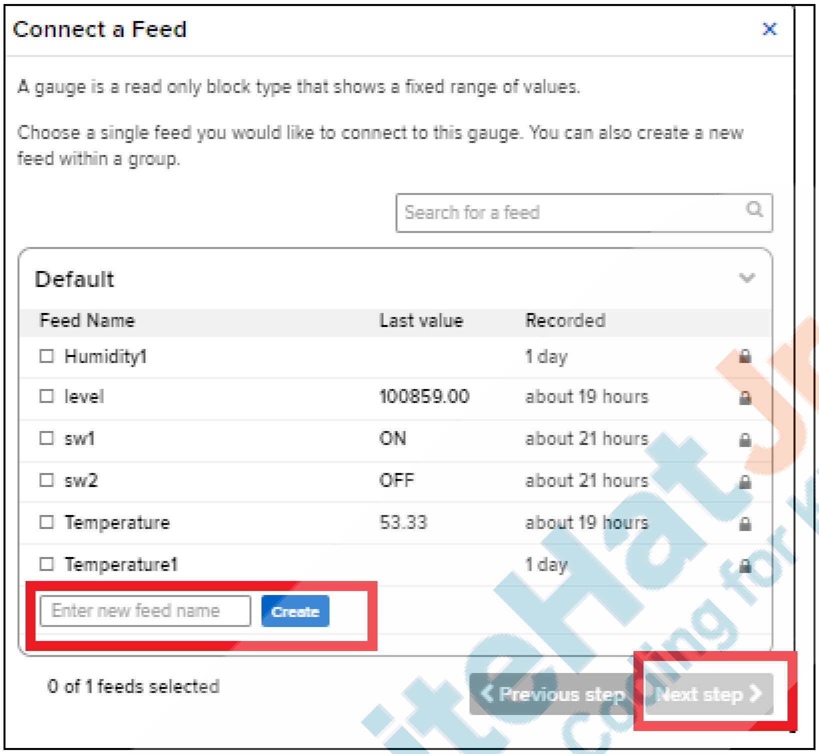
×
ON
RESET
HELLO WORLD!
Hello world! This is a different type of text block that works for larger quantities of data.

Connect a Feed

Feed - this is basically a set of data that you can read or write from like a sequential file. We can add data and we can receive the latest added data using feeds.

Note : During the first teacher/student experience, they may not see the screenshot below, they may only see the red highlighted one. Write down the name.

Note that we will use the same name later in our program. Try to write short names and without any space.



Connect a Feed

A gauge is a read only block type that shows a fixed range of values.

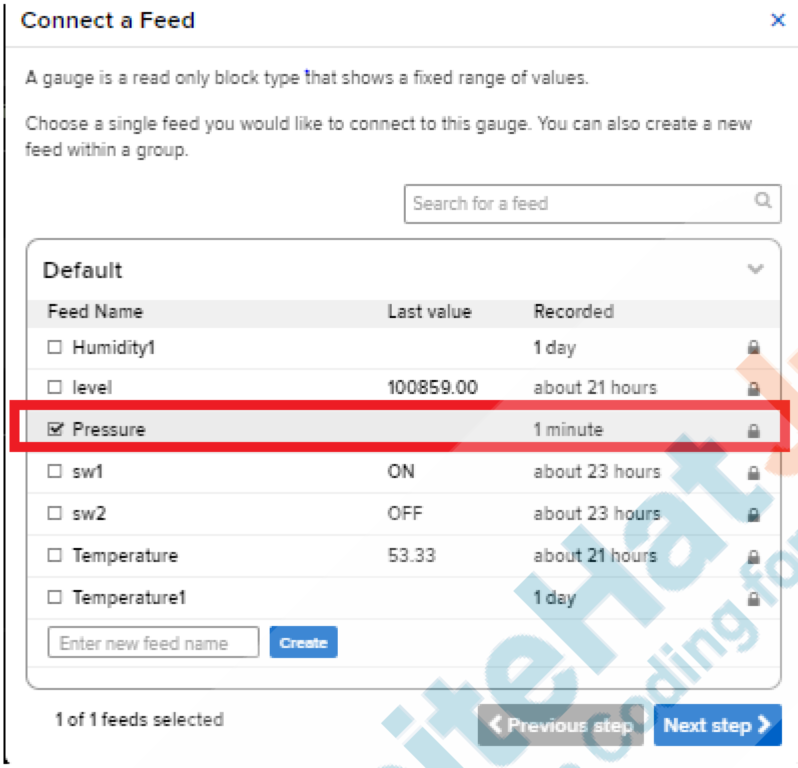
Choose a single feed you would like to connect to this gauge. You can also create a new feed within a group.

Default

Feed Name	Last value	Recorded
□ Humidity1		1 day
□ level	100859.00	about 19 hours
□ sw1	ON	about 21 hours
□ sw2	OFF	about 21 hours
□ Temperature	53.33	about 19 hours
□ Temperature1		1 day

0 of 1 feeds selected

Select the **Feed** which you have created just and then click on **Next Step**



Connect a Feed ×

A gauge is a read only block type that shows a fixed range of values.

Choose a single feed you would like to connect to this gauge. You can also create a new feed within a group.

Search for a feed

Default

Feed Name	Last value	Recorded
☐ Humidity1		1 day
☐ level	100859.00	about 21 hours
☑ Pressure		1 minute
☐ sw1	ON	about 23 hours
☐ sw2	OFF	about 23 hours
☐ Temperature	53.33	about 21 hours
☐ Temperature1		1 day

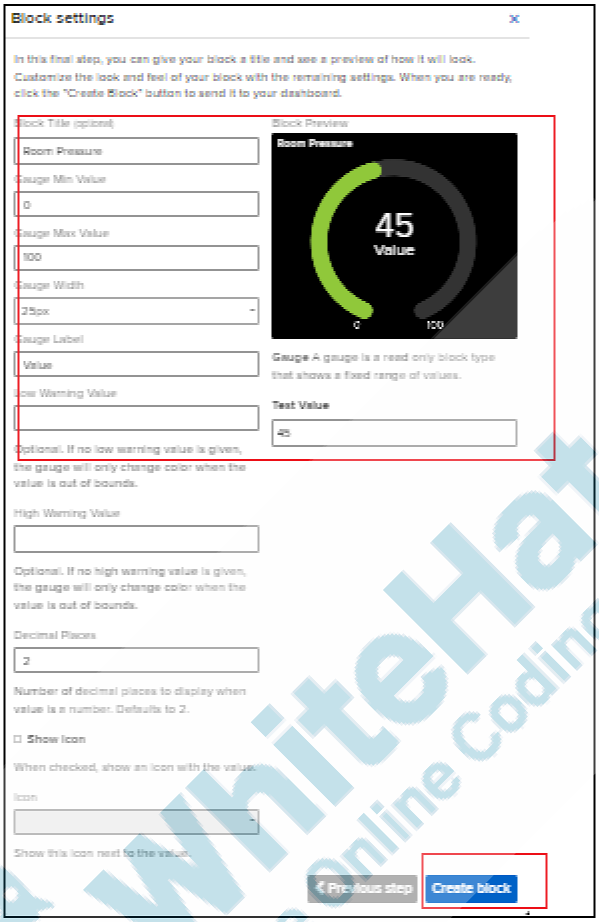
Enter new feed name Create

1 of 1 feeds selected

Select the **default values** and click on **Create Block**

In default, a thin type gauge will be used with min value 0 and max value 100. We can change the max value as per our wish.

Repeat the above steps to create one more **Gauge** for **Room Pressure**



Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional): Room Pressure

Gauge Min Value: 0

Gauge Max Value: 100

Gauge Width: 25px

Gauge Label: Value

Low Warning Value:

Optional. If no low warning value is given, the gauge will only change color when the value is out of bounds.

High Warning Value:

Optional. If no high warning value is given, the gauge will only change color when the value is out of bounds.

Decimal Places: 2

Number of decimal places to display when value is a number. Defaults to 2.

□ Show Icon

When checked, show an icon with the value.

Icon:

Show this icon next to the value.

Block Preview

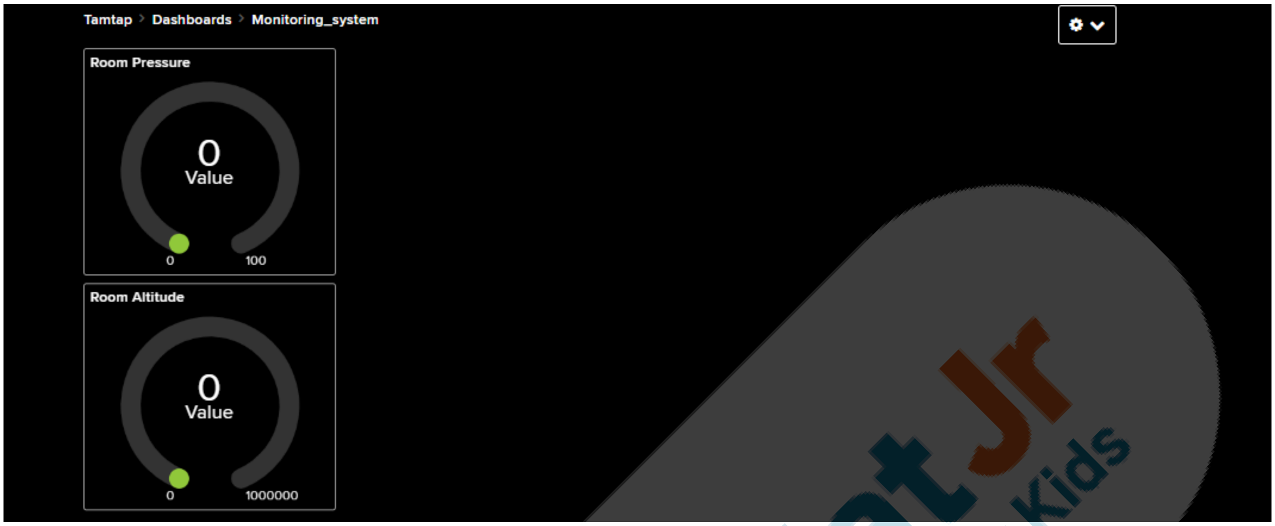
Gauge A gauge is a read only block type that shows a fixed range of values.

Test Value: 45

Repeat the same steps to create one more **Gauge** for **Room Temperature**

After creating two gauges one for **Room Pressure** and one for **Room Temperature** below the window will appear.

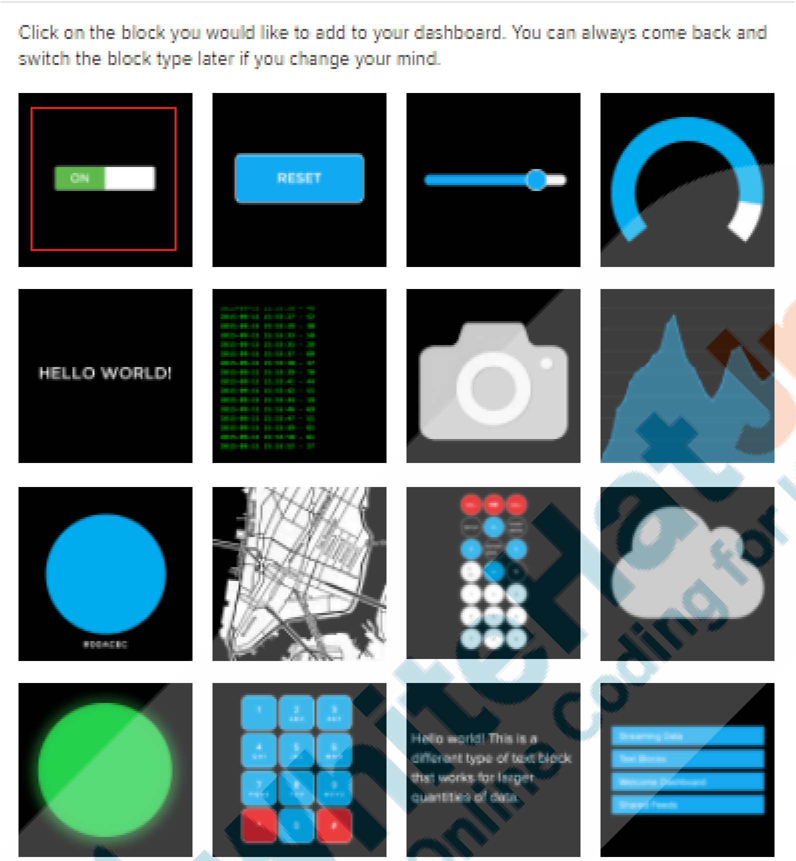
The gauges are now added to the dashboard



Next up, make another block, this time an on-off toggle switch for LED’s

Now select the two Toggle buttons for **Room AC** and **Room Light.**

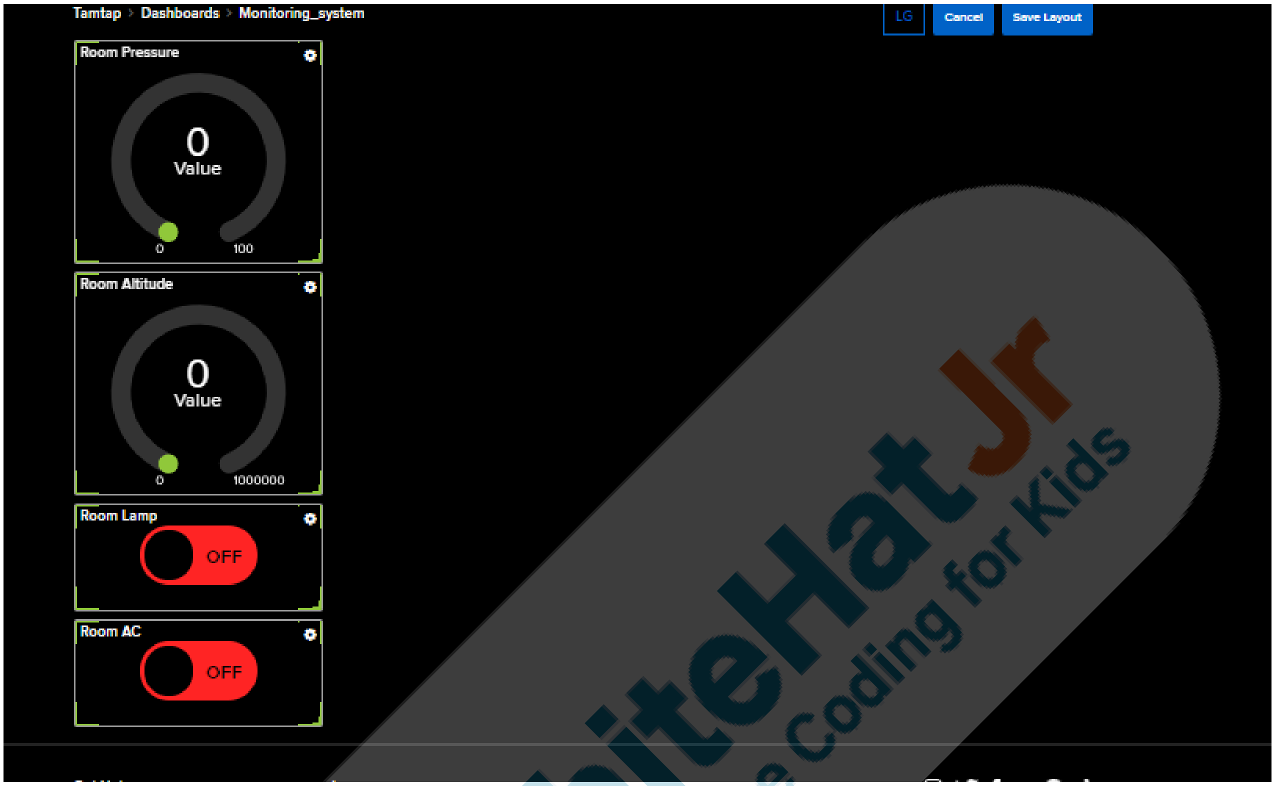
Set the default values **for** “**on**” **and** “**off**” texts



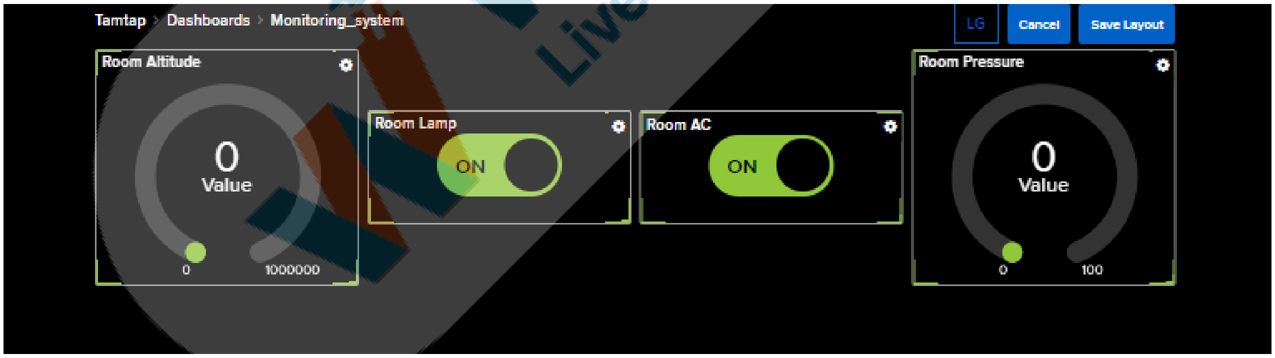
Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.

ON
RESET
HELLO WORLD!
#00ACEC
Hello world! This is a different type of text block that works for larger quantities of data.
Streaming Data
Text Blocks
Welcome Dashboard
Shared Feeds

Now after selecting **Toggle for AC & Light**, the below window will appear. Now, drag the **Feeds** to set the positions properly. Click on **Save Layout.**	



After Clicking on **Save Layout, the** window will appear like this:



Now we have prepared our cloud server on Adafruit, our next step is to integrate the **BMP180 sensor and LEDs.** After integration of **BMP180 sensor and LEDs** will send real time data on Adafruit **Dashboard.**

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)

FEEDBACK

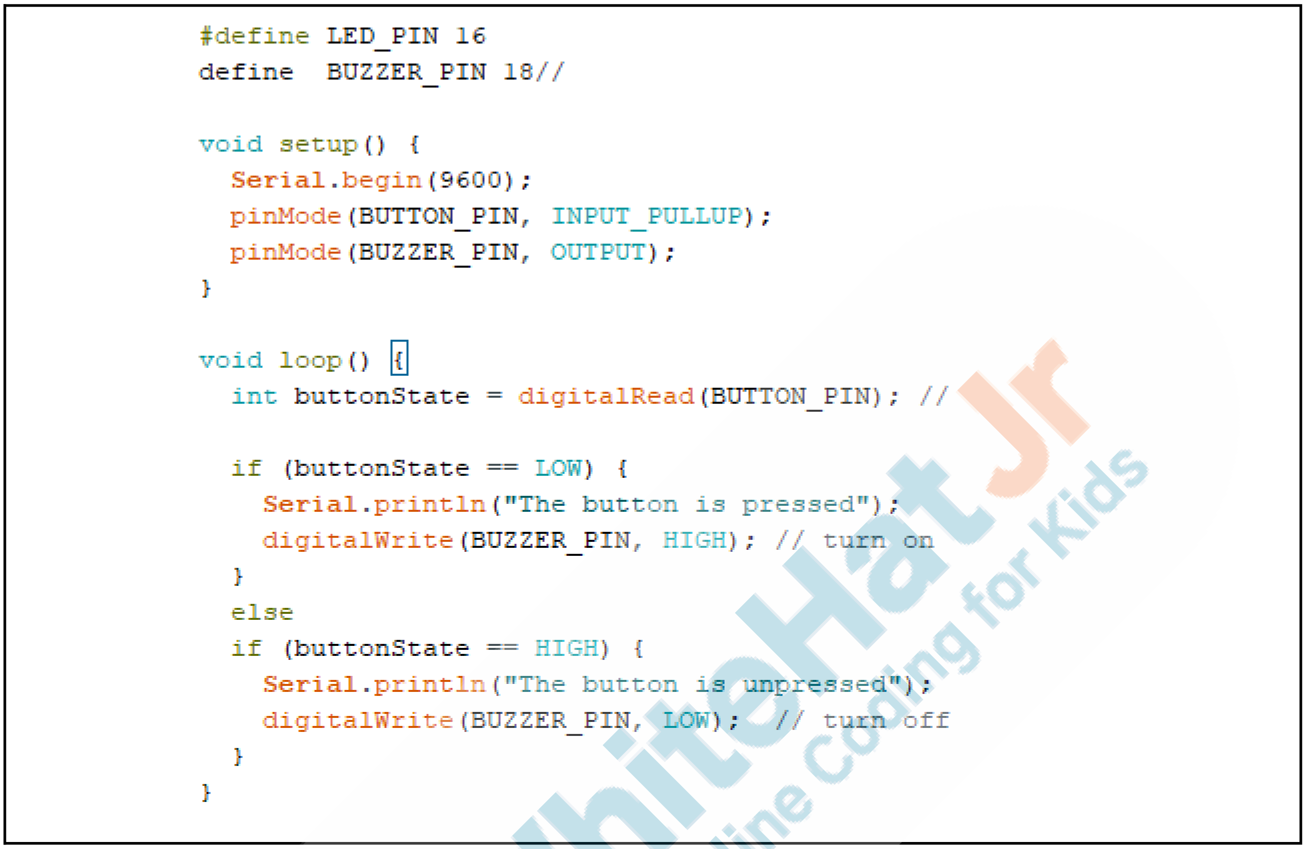
- **Appreciate and compliment the student for trying to learn a difficult concept.**
- **Get to know how they are feeling after the session.**
- **Review and check their understanding.**

Teacher Action	Student Action
You get “hats-off” for your excellent work! In the next class, we will learn about publishing & subscription of data	*Make sure you have given at least 2 hats-off during the class for:* Creatively Solved Activities +10

<table>
<tr><td></td><td>
</td></tr>
<tr><td colspan="2"><b>PROJECT OVERVIEW DISCUSSION</b>
Refer the document below in Activity Links Sections</td></tr>
<tr><td colspan="2"><b>Teacher Clicks</b>
</td></tr>
<tr><td colspan="2"><b>ADDITIONAL ACTIVITIES</b>
(Optional)</td></tr>
<tr><td><b>Additional Activities</b>

If still have time, tell the student to make a project in which it can control Buzzer using Push Button

● ESP32 makes a sound when the button is pressed and stop the sound when it is not pressed</td><td></td></tr>
</table>



```
#define LED_PIN 16
define  BUZZER_PIN 18//

void setup() {
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  pinMode(BUZZER_PIN, OUTPUT);
}

void loop() {
  int buttonState = digitalRead(BUTTON_PIN); //

  if (buttonState == LOW) {
    Serial.println("The button is pressed");
    digitalWrite(BUZZER_PIN, HIGH); // turn on
  }
  else
  if (buttonState == HIGH) {
    Serial.println("The button is unpressed");
    digitalWrite(BUZZER_PIN, LOW);  // turn off
  }
}
```

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Teacher Activity 1	Reference Code	https://github.com/procodingclass/PRO-C248-Reference-Code
Teacher Activity 2	Adafruit Account	https://accounts.adafruit.com/users/

		sign_in
Teacher Reference 1	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/a43f34f9-0dfb-4abe-ab19-0afd1a31cef8.docx
Student Activity 1	Adafruit Account	https://accounts.adafruit.com/users/sign_in