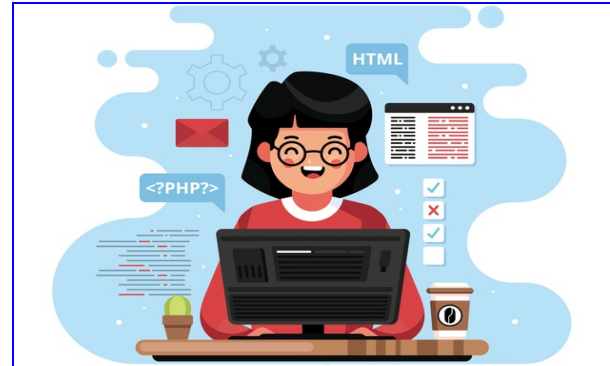


## MONITORING SYSTEM-1



### What is our GOAL for this CLASS?

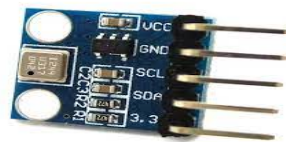
In this class, we were introduced to the **BMP180** pressure sensor, how to interface BMP180 with ESP32 and we designed a cloud server on **Adafruit**.

### What did we ACHIEVE in the class TODAY?

- We were introduced to the **BMP180** sensor.
- We learned about **I2C Communication**.
- We learned how to design a cloud server on **Adafruit**.

### Which CONCEPTS/ CODING BLOCKS did we cover today?

- We used the **BMP180** sensor.
  - It is used to measure atmospheric pressure.
  - We learned about the BMP180 Pin configuration.



- VCC: Connected to +5V
- GND: Connected to GND
- SCL: Serial Data pin (I2C interface)
- SDA: Serial Clock pin
- SCL & SDA are used to communicate with the ESP32 module. The data is

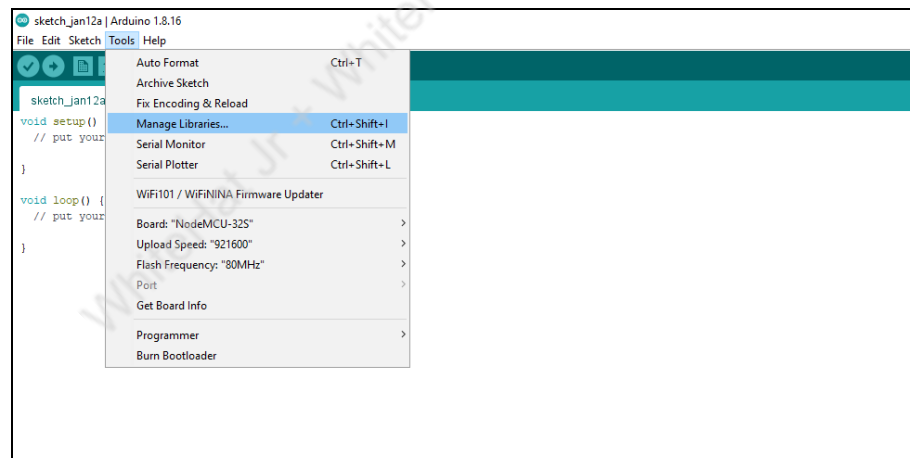
sent to the ESP32 or received from the ESP32 using these two pins.

- We learned about **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.
- We **installed libraries** for BMP180 Sensor
- We wrote a program for **barometric sensors**.
- We used the **Adafruit platform**
  - 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 server. It is a neutral party that your things can connect to send and receive messages.

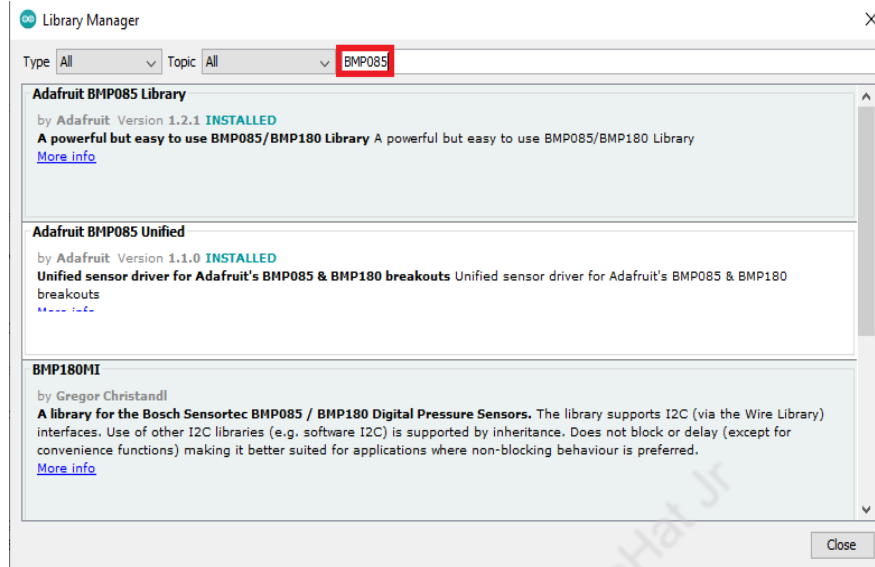
### How did we DO the activities?

#### 1. Install libraries

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



- Type BMP085 and then **click on Install**.
- Type MQTT and then **click on Install**.
- After Installing the sensor library below window will appear:



2. Gather the material from the IoT kit:

- 1 x ESP32
- 1 x USB Cable
- 1 x Breadboard
- 4 x Jumper wires
- 1 x BMP180

3. Connections for **Circuit Diagram**

- **BMP180 VCC pin:** Connect with 3V3 PIN of the ESP32
- **BMP180 GND pin:** Connect with GND of the ESP32
- **BMP180 SCL pin:** Connect with GPIO PIN 22
- **BMP180 SDA pin:** Connect with GPIO PIN 21

4. write a code for barometric sensors:

- Define the libraries
  - **Wire.h library** is used to communicate with I2C devices.
  - **Adafruit\_BMP085.h** library is used for pressure sensors.
- Create object bmp for **Adafruit\_BMP085**

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

Adafruit_BMP085 bmp;
```

5. Initialize the **setup()**

- **Serial.begin (9600)** is used for data exchange 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**.
- **bmp.begin()** is used to begin the process.
- **Serial.println** is used to print data. Print ("Could not found", if it fail to begin the process)

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

6. 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);  
}
```

## 7. 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

- 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
8. To send data on a cloud server, use an online server, and to access an online server use the platform **Adafruit**.

**9. Set up an online server**

- Click on **SIGN UP**
- Add your **SIGN IN** details
- Click on **CREATE ACCOUNT**



The screenshot shows the Adafruit 'SIGN UP' form. The 'SIGN UP' text is highlighted with a red box. The form includes a descriptive paragraph about creating an account, followed by input fields for 'FIRST NAME', 'LAST NAME', 'EMAIL', and 'USERNAME'. A note states 'Username is viewable to the public on the forums, Adafruit IO, and elsewhere.' Below these is a 'PASSWORD' field. A blue 'CREATE ACCOUNT' button is highlighted with a red box. At the bottom, there is a link for 'HAVE AN ADAFRUIT ACCOUNT?' with a blue 'SIGN IN' button.

- Click on **IO**
- Go to **Dashboards**
- Click on **New Dashboard**
- Write the **Name** (Environment Monitor System)
- Write **Description** if needed

Create a new Dashboard ×

Name

Description

Cancel

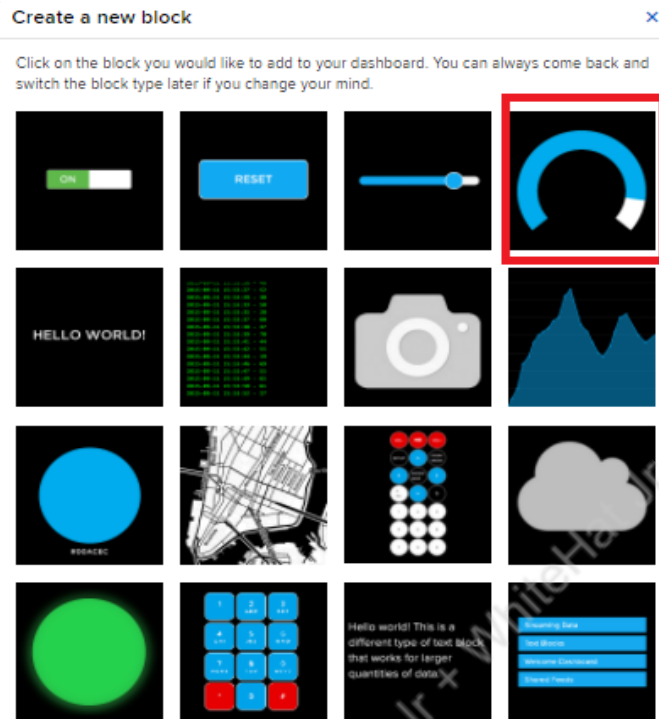
Create

10. Click on **Create New Block**



11. Select the **Gauge**

- **Gauges** are visual block to represent sensor values



## 12. Enter new Feed

- This is 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.
- During the experience, you may not see the screenshot below, you may only see the red highlighted one. Write down the name.
- Use the same name in the program. Write short names 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.

Search for a feed

**Default** ▼

Feed Name	Last value	Recorded	
<input type="checkbox"/> Humidity1		1 day	🔒
<input type="checkbox"/> level	100859.00	about 19 hours	🔒
<input type="checkbox"/> sw1	ON	about 21 hours	🔒
<input type="checkbox"/> sw2	OFF	about 21 hours	🔒
<input type="checkbox"/> Temperature	53.33	about 19 hours	🔒
<input type="checkbox"/> Temperature1		1 day	🔒

0 of 1 feeds selected

13. Select the **Feed** which you have created 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	
<input type="checkbox"/> Humidity1		1 day	🔒
<input type="checkbox"/> level	100859.00	about 21 hours	🔒
<input checked="" type="checkbox"/> Pressure		1 minute	🔒
<input type="checkbox"/> sw1	ON	about 23 hours	🔒
<input type="checkbox"/> sw2	OFF	about 23 hours	🔒
<input type="checkbox"/> Temperature	53.33	about 21 hours	🔒
<input type="checkbox"/> Temperature1		1 day	🔒

1 of 1 feeds selected

14. 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 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



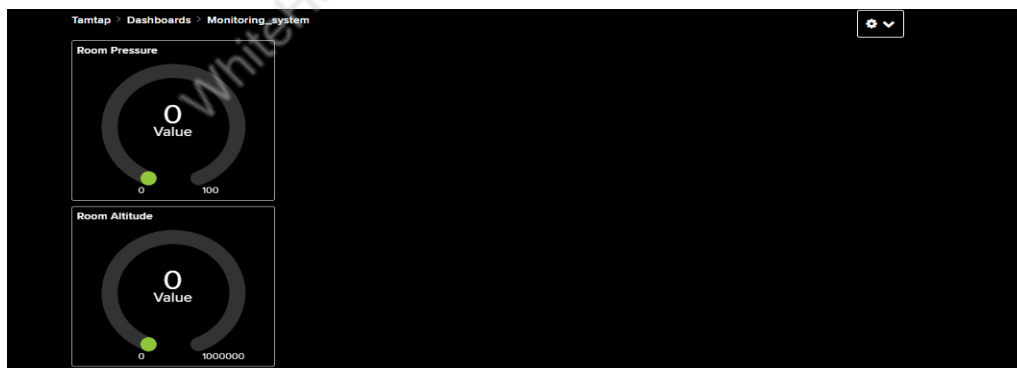
Gauge A gauge is a read only block type that shows a fixed range of values.

Test Value

45

< Previous step   **Create block**

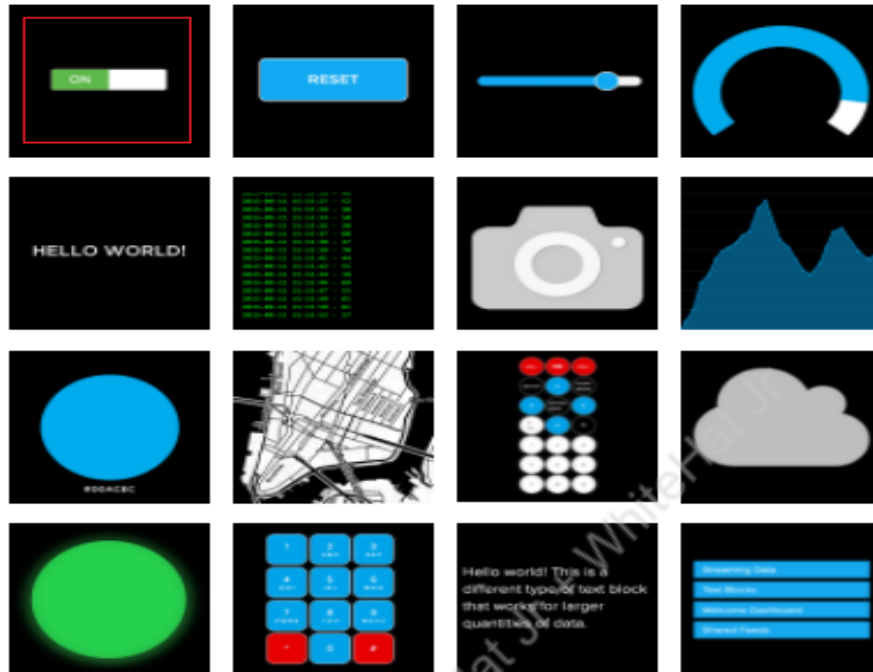
- Repeat the same steps to create one more Gauge for Room Temperature
- After creating two gauges one for Room Pressure and one for Room Temperature the below the window will appear.
- The gauges are now added to dashboard



15. 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

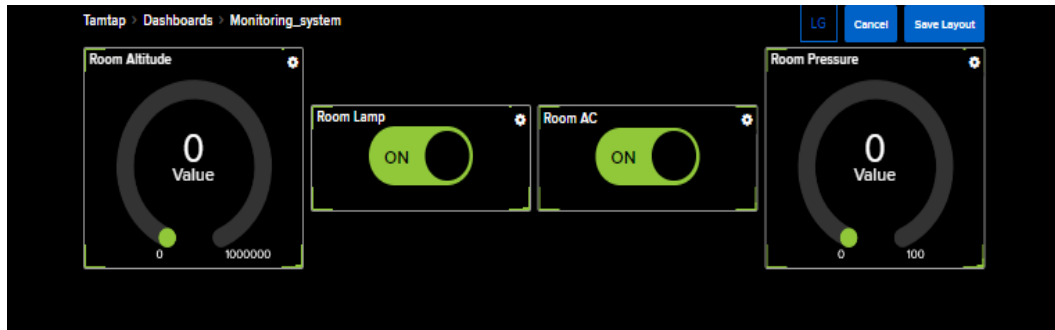
Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.



- 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:



- The **cloud server on Adafruit** is created, the next step is to integrate the BMP180 sensor and LEDs.
- After integration of **BMP180 sensor and LEDs** it will send real time data on Adafruit Dashboard.

### What's NEXT?

In the next class, we will learn **about publishing & subscription of data.**

### Expand Your Knowledge

To know more about cloud servers [click here](#).