Topic	ANALOG INPUT AND SENSORS	
Class Description	**Students will be introduced to the concept of analog input and sensors and how the sensor communicates with the ESP32 and shows the output in analog format**	
Class	**PRO C245**	
Class time	**50 mins**	
Goal	• Introduction to the analog input & potentiometer • Introduction to DTH11 sensor	
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** **15 mins** **10 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?	**ESR**: Hi, thanks! Yes, I am excited about it!

Following are the WARM-UP session deliverables: • Greet the student. • Revision of previous class activities. • Quizzes.	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 - 15 mins	
Teacher Initiates Screen Share	
ACTIVITY • **Introduction to Analog Input** • **Potentiometer**	
Teacher Action	**Student Action**
Have you ever noticed the type of meters on vehicles, such as the one on an Activa or a car's which will let you know about the distance and even fuel quantity? What type of meters are they? You must be surprised to know that these meters played an important role in electronics too. Working with electronics means dealing with a lot of meters,	**ESR: Varied!**

We learned about current, voltage, resistors when we started this module and I told you we measured these quantities too. To measure these quantities we have units assigned for each but now the thing is how we can measure these things. Any ideas? To measure we had a voltmeter, to measure current we had an ammeter, When we talk about these meters means we are providing input signals to these meters which measure and show output on meters. To measure these types of inputs and outputs signals we have analog and digital meters. In electronics circuits, Input & output matter a lot. One device output can be input for others You know everything around is you infinite, and then infinite signals are known as analog signals Can you give an example for analog signals? Yes, more examples are voltage, current In short, whichever signal we see in the waveform is called an **Analog signal** You must be curious to see an example of Analog signals: **In our IoT kit, we have the rotary potentiometer**	**ESR:** Audio, Video

Note: The teacher will take the potentiometer out from the IoT kit and show it to the student.

Potentiometer: A potentiometer is a 3 terminal device in which the resistance is manually varied to control the flow of electric current.

A potentiometer usually has 3 pins:

VCC pin: Connect this pin to **VCC (5V or 3.3v)**
GND pin: connect this pin to **GND (0V)**
DATA pin: Data pin is used to communicate between the potentiometer and ESP32, **Data** pin outputs the voltage to ESP32's input pin.

The GND and VCC will be interchangeable

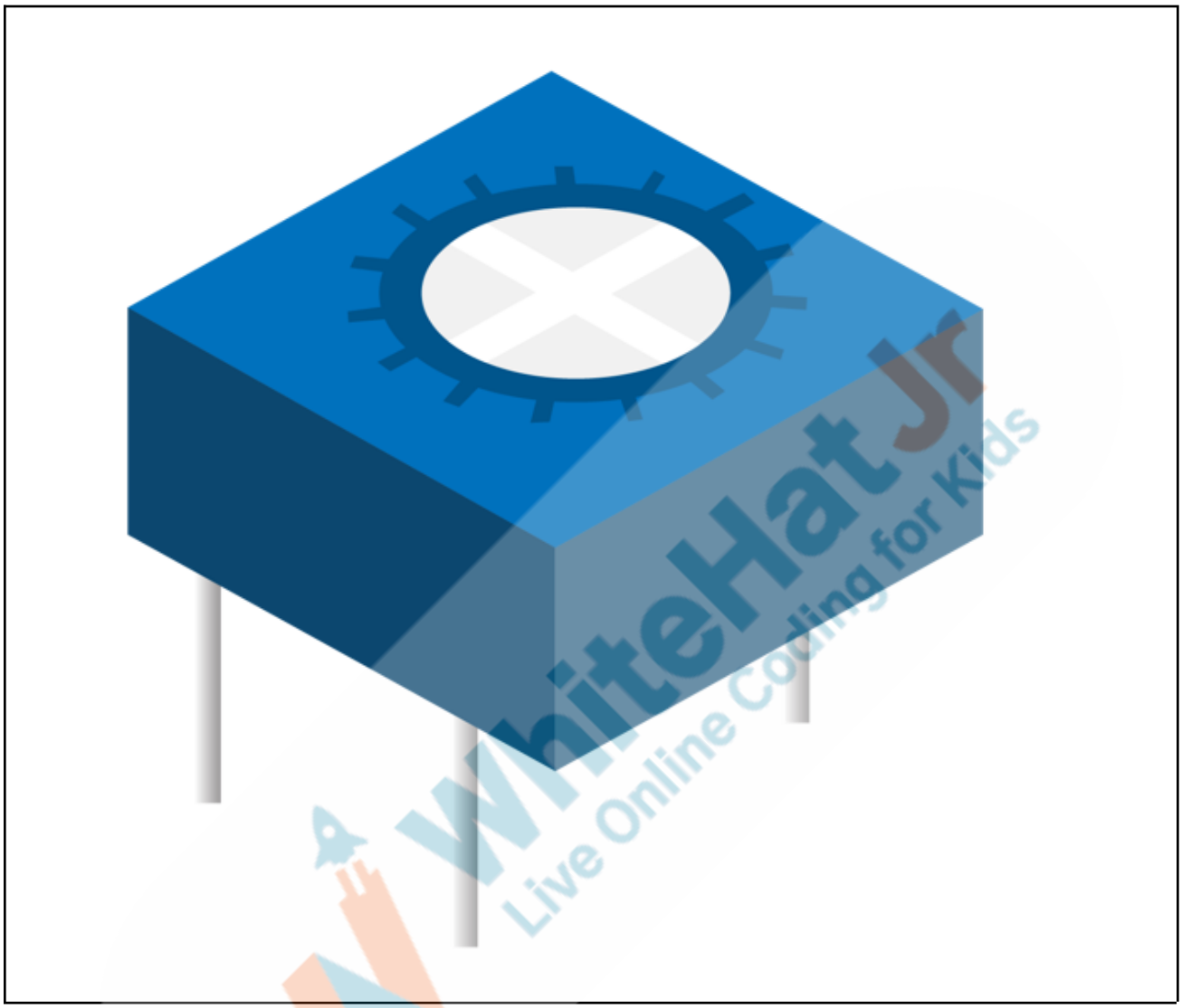
How does Potentiometer Work?

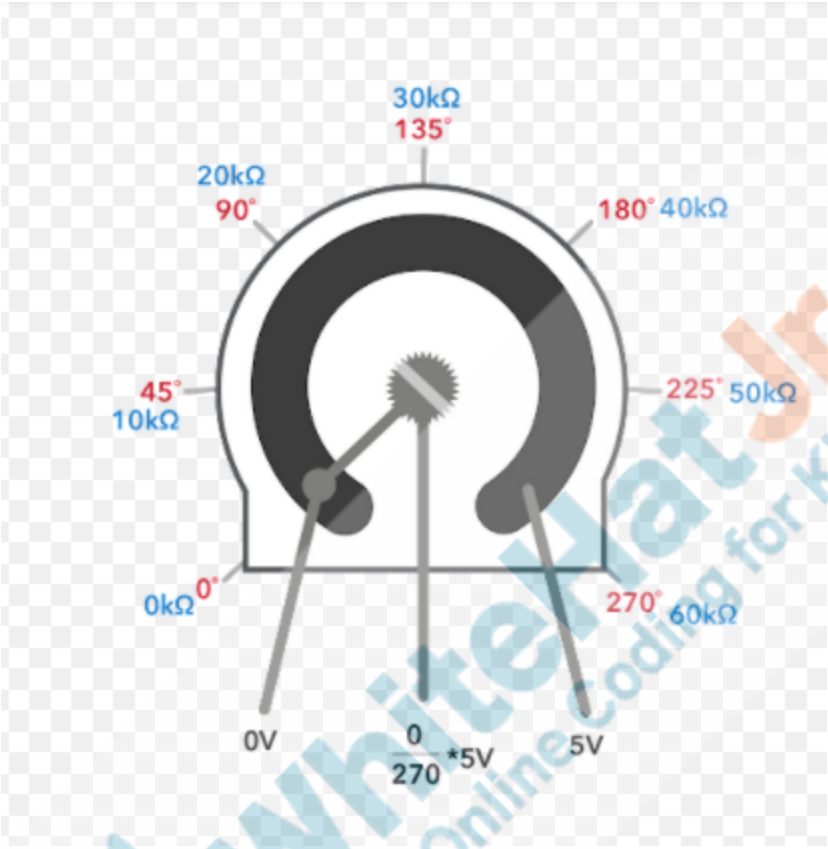
The potentiometer's shaft can be rotated from 0 (Minimum) to 270(Maximum)

The voltage in the output pin is in direct proportion to the angle varying from 0 to VCC.

The below table shows the relation between the angle and the voltage on the output pin:

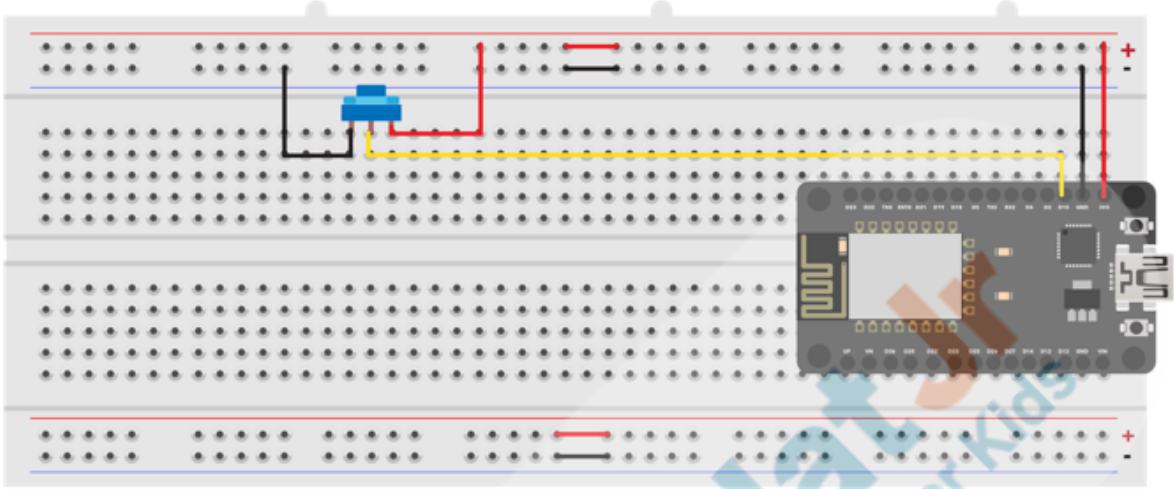
0 degree	***0 V***
270 degree	***VCC (5V)***





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 Potentiometer	
Step -2: Let's do connections: ● Supply positive**(VCC (+ve))** from the ESP 32 to breadboard **+ve** terminal ● Supply negative **(GND (-ve))** from the ESP 32 to	

breadboard negative terminal • Now our breadboard have (VCC (+ve)) and (GND (-ve)) supply • Insert potentiometer into the breadboard • Keep track of all three pins, Ist and last pin of potentiometer are used for (VCC (+ve)) and (GND (-ve)) respectively *Note : (VCC (+ve)) and (GND (-ve)) of the potentiometer can be interchangeable.* • Provide supply to the potentiometer from the breadboard • *Note: In Breadboards power rail first five are connected, to make other connected provide jumper wires as shown in the circuit diagram.* • Now it's time to connect the middle pin i.e output of potentiometer. connect the potentiometer's output pin to an ESP32's analog input pin 36. The ESP32's analog input pin converts the voltage (between 0v and 3.3V) into integer values (between 0 and 4095), or analog values. *Note: Multifunctionality of GPIO input pins, here pin 36 will acts as an Analog pin*	



Step-3 Let's write a code:

To Map value of voltage with potentiometer we need to use **float map()** function. It will map 0 to 3.3V/5V according to the rotation of the potentiometer. It will map value from low to high or high to low.

floatMap() is used to get all decimal value

```
float floatMap(float x, float in_min, float in_max, float out_min, float out_max) {
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
```

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.

`` void setup() { Serial.begin(9600); } ```	
To execute the main process write the **void loop()** ● The output pin of the potentiometer is connected to the ESP32 analog input pin. To read this value we need to use the **analogRead()** function. **analogRead()** function is used to read the value of the potentiometer reading. This means that it will map input voltages between 0 and the operating voltage(5V or 3.3V) into integer values between 0 and 1023 ● Rescale to the potentiometer's angle by using the **float map()** function. ● Rescale to the potentiometer's voltage: ● **Serial. print()** is used to print the value, print Analog, and then analog value, voltage ● **Serial.println(volage*1000)** is used to show voltage in terms of 1.00	

```
void loop() {

  int analogValue = analogRead(4);

  float voltage = floatMap(analogValue, 0, 4095, 0, 3.3);

  Serial.print("Analog: ");
  Serial.print(analogValue);
  Serial.print(", Voltage: ");
  Serial.println(voltage*1000);

  delay(1000);
}
```

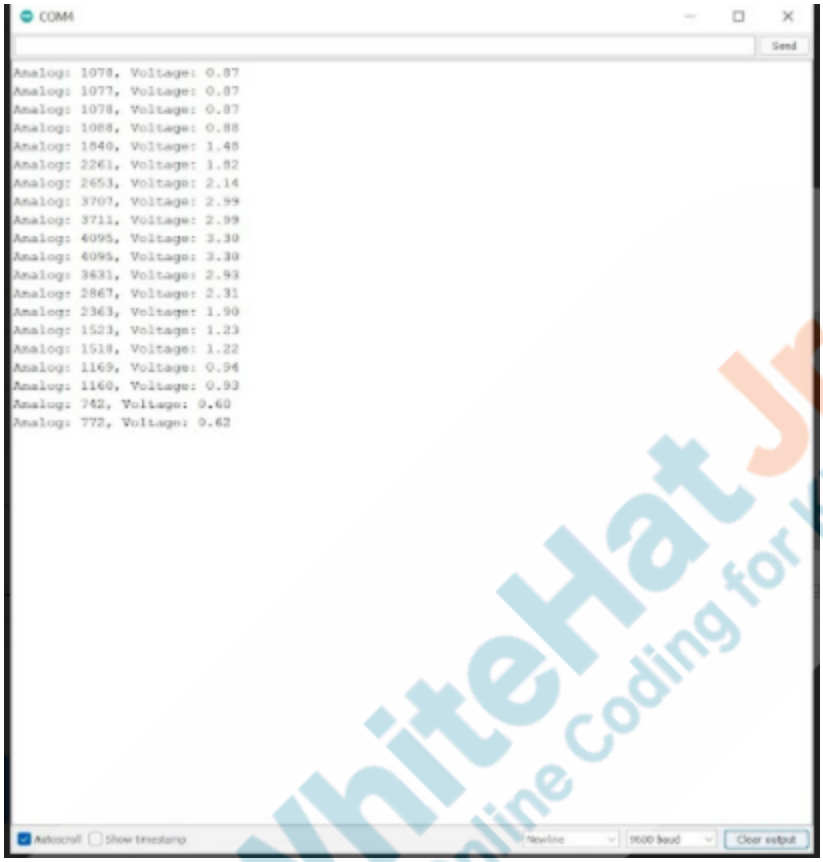
Output:

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

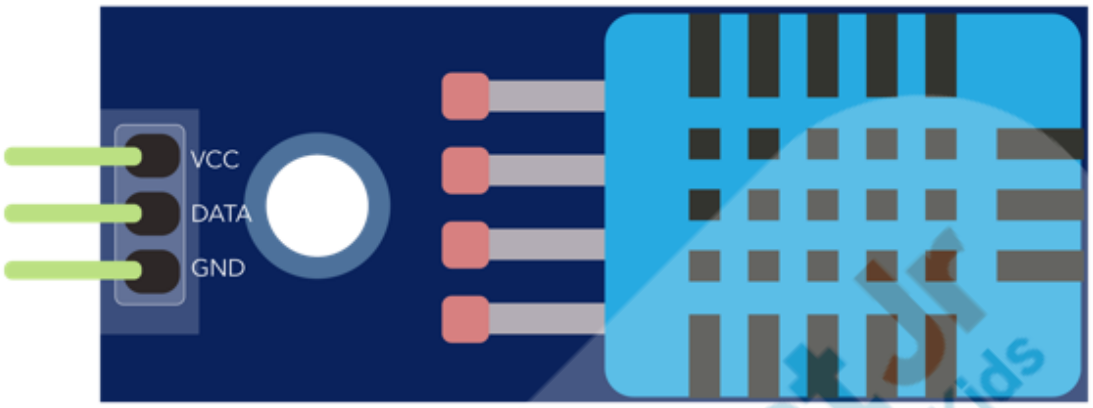
- Go to **Tools** and select **Serial Monitor**
- Rotate the **potentiometer** and see the output



By rotating the potentiometer knob, the voltage value will rise or fall.	
Teacher Stops Screen Share	
So now it's your turn. Please share your screen with me.	
Teacher Starts Slideshow  Slide 14-15 Refer to speaker notes and follow the instructions on each slide.	
We have one more class challenge for you.	

Can you solve it? Let's try. I will guide you through it.	
Teacher Ends Slideshow 	
STUDENT-LED ACTIVITY - 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 • Student Activity description (in bullet points).	
Teacher Action	Student Action
Can you see pressure or temperature? We can't see with our eyes but we can record them, Right! But how? Yes, we need to use sensors! **Sensors:** A sensor is a device that measures a physical quantity like pressure temperature and converts it into a 'signal' which can be read by an observer or by an instrument.	**ESR:** Varied! **ESR:** Varied! **ESR:** Using Sensors!

Like current can be checked by Ammeter, voltage by voltmeter What if you want to measure pressure and humidity surrounding you? Yes, we need sensors. To measure Temperature and humidity sensors we have a **DTH11** sensor Open your Iot Kit and check the **DHT11** sensor So our main task is to read the temperature and humidity value from **DHT11** and print it to **Serial Monitor** and check the same in the form of analog input too.	**ESR**: We need sensors!
DHT11 Sensor: Use to measure Temperature and Humidity sensors **DHT11 include 4 pins:** **GND pin**: connect this pin to GND (0V) **VCC pin**: connect this pin to VCC (3.3V) **DATA pin:** the pin is used to communicate between the sensor and ESP32, outputs the voltage to ESP32's input pin.	



Step -1:Gather the material from the IoT kit:

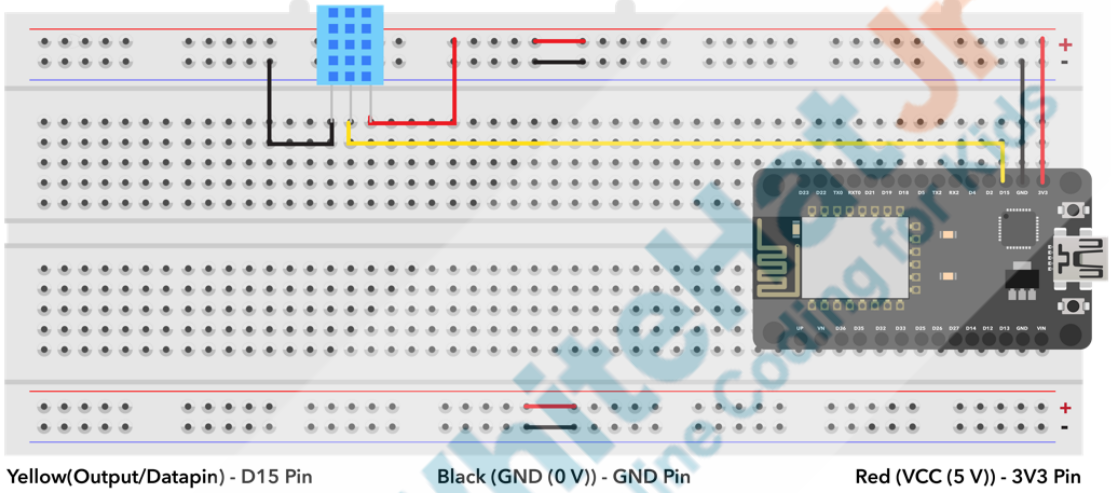
- 1 x ESP32
- 1 x USB Cable
- 1 x Breadboard
- 4 x Jumper wires
- 1 x Potentiometer
- 1 x Rotary Potentiometer

Step -2 : Let's do connections:

- Supply positive(VCC (+ve)) from the ESP 32 to breadboard terminal
- Supply negative(GND (-ve)) from the ESP 32 to breadboard negative terminal
- Now our breadboard have (VCC (+ve)) and (GND (-ve)) supply
- Take the DHT11 sensor, female jumper wires are already connected with DHT11
- Take three male jumper wires to insert into DHT11 sensor
- Now connect **VCC (+ve)** of DHT11 with **VCC (+ve)**

of the breadboard

- connect **GND(-ve)** of DHT11 with **GND(-ve)** of the breadboard
- Connect data/output pin of DHT11 with **D15** of the **ESP32**



Step-3 Let's write a code:

Define Pins

- define **DHTPIN 15**
- define **DHTPIN DHT11**

```
#define DHTPIN 15

#define DHTTYPE DHT11


DHT dht(DHTPIN, DHTTYPE);
```

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.
- **Serial.println is** used to print data. Print ("**DHT11 sensor!**")
- **dht. begin()** is used to begin the process

```
void setup() {
  Serial.begin(9600);
  Serial.println("DHT11 sensor!");
  //call begin to start sensor
  dht.begin();
}
```

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

- **Create float h and float t** variable to store decimal value
- **readHumidity()** will read the sensor's humidity value.
- **readTemperature()** will read the sensor's temperature value.
- Check **if** any reads failed **and exit** early using **isnan()**
- **Serial.print** used to print data. Print ("**Failed to read from DHT sensor!**")
- Return the process using **return()**
- **Serial.print()** is used to print the value, print **Humidity, (h), Temperature, (t)**
- Set **delay** of 2000ms

```
void loop() {
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // print the result to Terminal
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(",");
  Serial.print("Temperature: ");
  Serial.println(t);
  delay(2000);
}
```

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 port is not selected, insert the USB cable in Computer's port and select the port

- Go to **Tools** and select **Serial Monitor**
- Rotate the potentiometer and see the output

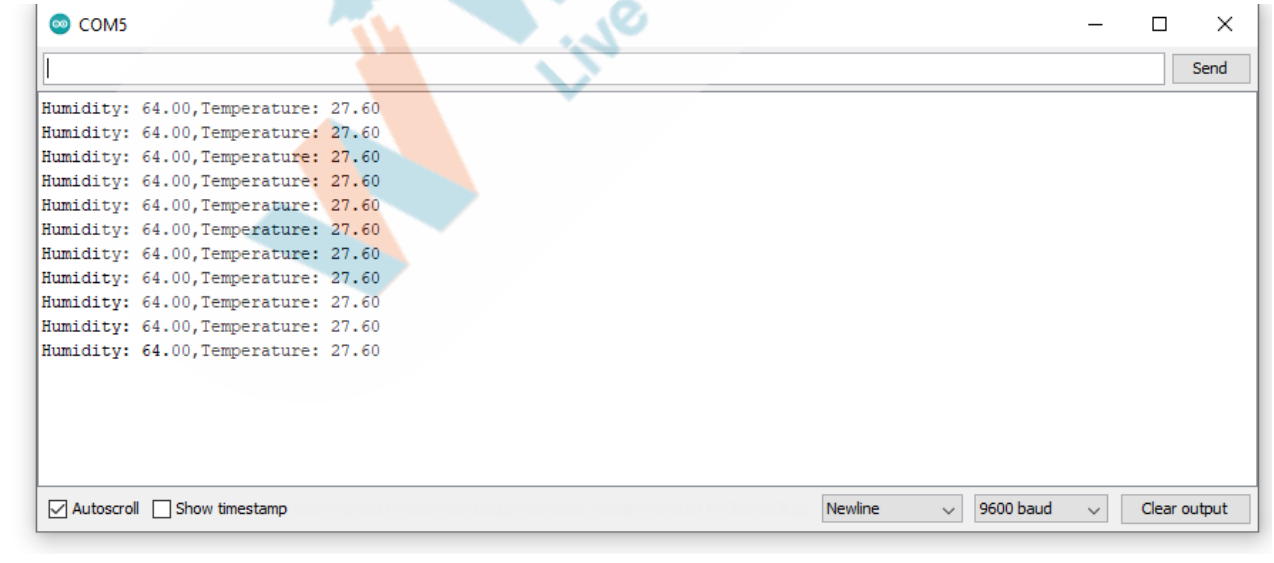
Error Message: If an error message comes like no such file or directory then use the below method to resolve this error

Go to Tools

- Click on Manage Libraries
- Write the component name which needs to install
- Click on Install

Go to Tools and select **Serial Monitor**

- See the **Humidity** and **Temperature** value



As we are learning analog input let's see analog waveform

- Go to Tools and select Serial Plotter
- See the **Humidity** and **Temperature** value

So, Today we made an arrangement where we saw the exact readings of the humidity and temperature around us using a **DHT11** sensor and even we learned how potentiometer works

That's fun!

Teacher Guides Student to Stop Screen Share

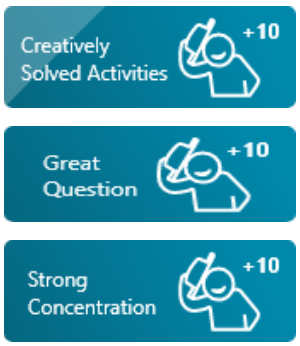
WRAP-UP SESSION - 10 mins

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! *Make sure you have given at least 2 hats-off during the class* In the next class, we will learn about RGB LED	
PROJECT OVERVIEW DISCUSSION Refer the document below in Activity Links Sections	
Teacher Clicks ✖ End Class	
ADDITIONAL ACTIVITIES (Optional)	

Additional Activities	

ACTIVITY LINKS		
Activity Name	**Description**	**Links**
Teacher Activity 1	Reference Code for Potentiometer	https://github.com/procodingclass/PRO-C245-Reference-Code
Teacher Reference 1	Reference Code for DHT11	https://github.com/procodingclass/PRO-C245-Reference-Code-SA
Teacher Reference 2	Project	https://s3-whjr-curriculum-uploads.whjr.online/41522a11-abea-4ebf-b443-a118183cd09d.docx
Teacher Reference 3	Project Solution	https://github.com/procodingclass/PRO-C245-Project-Solution
Teacher Reference 4	In-Class Quiz	https://s3-whjr-curriculum-uploads.whjr.online/27bfe922-fd27-4ac1-8fb0-cb9bcd235172.docx