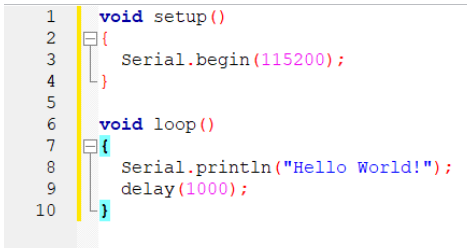
Activity 2	**Exploring ESP32 with Arduino IDE**	**31/1/2025**

In this activity, you will learn:

- How to write a simple program
- Understanding the serial operations
- Printing time as serial output
- Connecting to WiFi
- LED Blink project

Displaying 'Hello World' - serial output

Write the following code in Arduino IDE



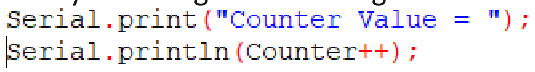
```
void setup()
{
  Serial.begin(115200);
}

void loop()
{
  Serial.println("Hello World!");
  delay(1000);
}
```

The delay() function accepts a single integer (or number) argument. This number represents the time (measured in milliseconds).

a. Choose the board, ESP32, COM port, hold down the BOOT button, click upload and keep your finger on the BOOT button pressed.
b. When the Arduino IDE starts sending the code, you can release the button and wait for the flashing process to be completed.
c. The ESP32 is flashed with the new firmware.

Open the Arduino serial monitor from the icon on the top right or from the tools menu and select Tools > Serial Monitor

Q1. What is baud rate?
Q2. Change the code and display 'Hello World' after every 5 milliseconds
Q3. Modify the code above by including the following lines before the delay() function

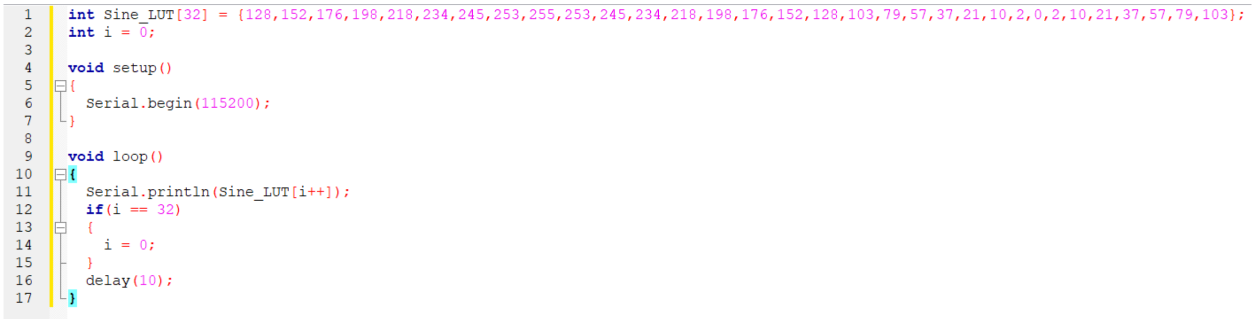


```
Serial.print("Counter Value = ");
Serial.println(Counter++);
```

Determine the output.

Using the serial plotter

Write the following code in Arduino IDE

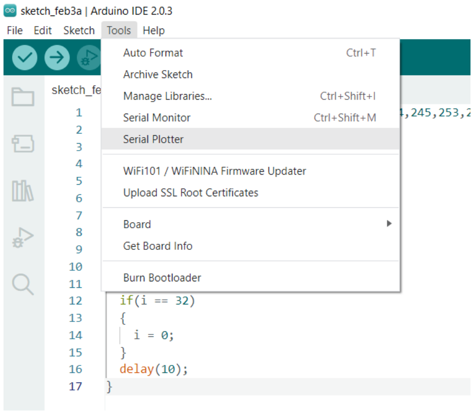


```
int Sine_LUT[32] = {128,152,176,198,218,234,245,253,255,253,245,234,218,198,176,152,128,103,79,57,37,21,10,2,0,2,10,21,37,57,79,103};
int i = 0;

void setup()
{
  Serial.begin(115200);
}

void loop()
{
  Serial.println(Sine_LUT[i++]);
  if(i == 32)
  {
    i = 0;
  }
  delay(10);
}
```

a. Choose the board, ESP32, COM port, hold down the BOOT button, click upload and keep your finger on the BOOT button pressed.

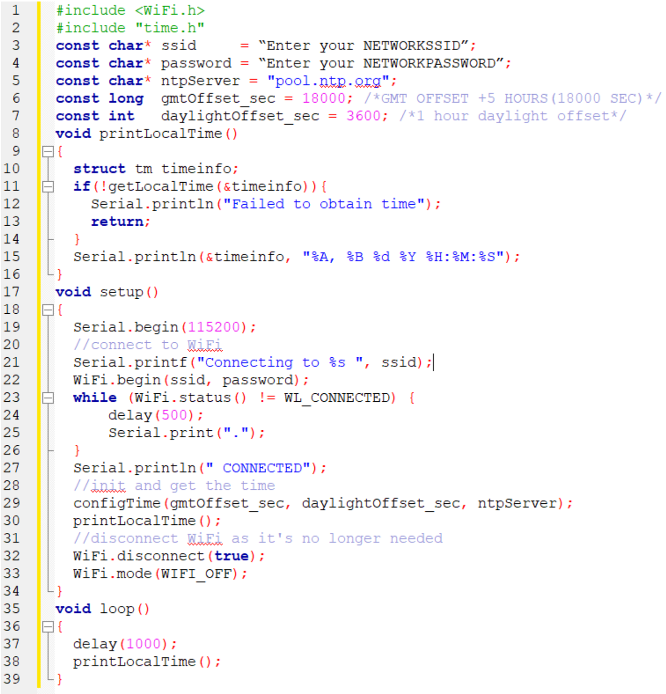
b. When the Arduino IDE starts sending the code, you can release the button and wait for the flashing process to be completed.

c. Display the Sine wave using the option displayed below.


sketch_feb3a | Arduino IDE 2.0.3
File
Edit
Sketch
Tools
Help
sketch_fe
Auto Format Ctrl+T
Archive Sketch
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter
WiFi101 / WiFiNINA Firmware Updater
Upload SSL Root Certificates
Board
Get Board Info
Burn Bootloader

```
,245,253,
  if(i == 32)
  {
    i = 0;
  }
  delay(10);
}
```

Printing the Time

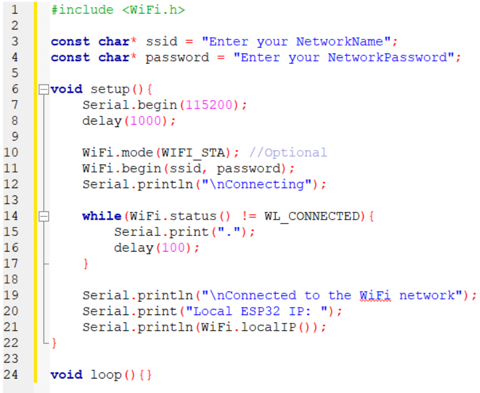


```
#include <WiFi.h>
#include "time.h"
const char* ssid      = “Enter your NETWORKSSID”;
const char* password = “Enter your NETWORKPASSWORD”;
const char* ntpServer = "pool.ntp.org";
const long  gmtOffset_sec = 18000; /*GMT OFFSET +5 HOURS(18000 SEC)*/
const int   daylightOffset_sec = 3600; /*1 hour daylight offset*/
void printLocalTime()
{
  struct tm timeinfo;
  if(!getLocalTime(&timeinfo)){
    Serial.println("Failed to obtain time");
    return;
  }
  Serial.println(&timeinfo, "%A, %B %d %Y %H:%M:%S");
}
void setup()
{
  Serial.begin(115200);
  //connect to WiFi
  Serial.printf("Connecting to %s ", ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
  }
  Serial.println(" CONNECTED");
  //init and get the time
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  printLocalTime();
  //disconnect WiFi as it's no longer needed
  WiFi.disconnect(true);
  WiFi.mode(WIFI_OFF);
}
void loop()
{
  delay(1000);
  printLocalTime();
}
```

Q4. Determine the output.

Connecting to WiFi

Write the following code in Arduino IDE



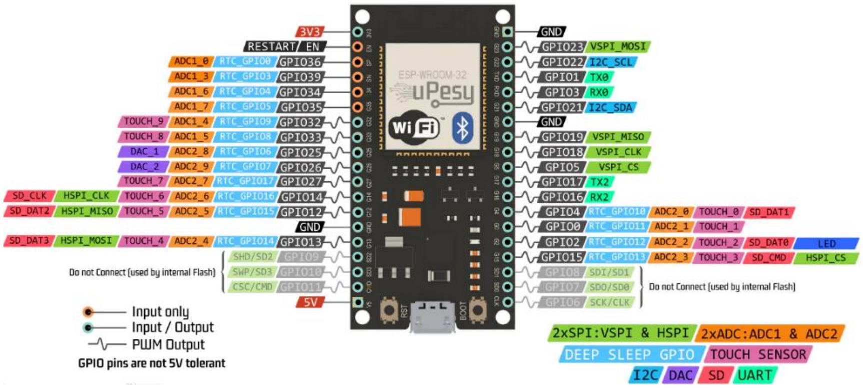
```
#include <WiFi.h>

const char* ssid = "Enter your NetworkName";
const char* password = "Enter your NetworkPassword";

void setup(){
    Serial.begin(115200);
    delay(1000);

    WiFi.mode(WIFI_STA); //Optional
    WiFi.begin(ssid, password);
    Serial.println("\nConnecting");

    while(WiFi.status() != WL_CONNECTED){
        Serial.print(".");
        delay(100);
    }

    Serial.println("\nConnected to the WiFi network");
    Serial.print("Local ESP32 IP: ");
    Serial.println(WiFi.localIP());
}

void loop(){}
```

Connect your ESP32 kit to WiFi using your Mobile data connection Hotspot

Q5. Determine the output if your module gets connected

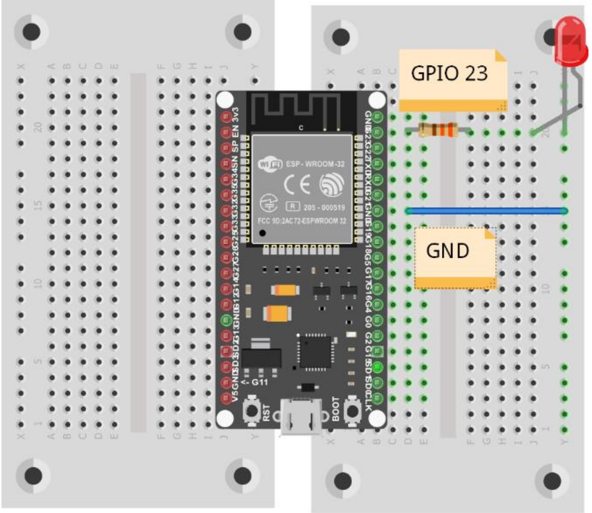
LED Blink Project

Following is the pinout of your ESP32 WROOM based MCU Kit

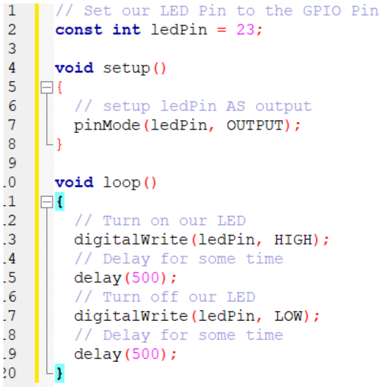

3V3
RESTART / EN
ADC1_0 / RTC_GPIO0 / GPIO36
ADC1_3 / RTC_GPIO3 / GPIO39
ADC1_6 / RTC_GPIO4 / GPIO34
ADC1_7 / RTC_GPIO5 / GPIO35
TOUCH_9 / ADC1_4 / RTC_GPIO9 / GPIO32
TOUCH_8 / ADC1_5 / RTC_GPIO8 / GPIO33
DAC_1 / ADC2_8 / RTC_GPIO6 / GPIO25
DAC_2 / ADC2_9 / RTC_GPIO7 / GPIO26
TOUCH_7 / ADC2_7 / RTC_GPIO17 / GPIO27
SD_CLK / HSPI_CLK / TOUCH_6 / ADC2_6 / RTC_GPIO16 / GPIO14
SD_DAT2 / HSPI_MISO / TOUCH_5 / ADC2_5 / RTC_GPIO15 / GPIO12
GND
SD_DAT3 / HSPI_MOSI / TOUCH_4 / ADC2_4 / RTC_GPIO14 / GPIO13
SHD/SD2 / GPIO9
SWP/SD3 / GPIO10
CSC/CMD / GPIO11
Do not Connect (used by internal Flash)
5V
Input only
Input / Output
PWM Output
GPIO pins are not 5V tolerant
ESP-WROOM-32
uPesy
Wi Fi
GND
GPIO23 / VSPI_MOSI
GPIO22 / I2C_SCL
GPIO1 / TX0
GPIO3 / RX0
GPIO21 / I2C_SDA
GND
GPIO19 / VSPI_MISO
GPIO18 / VSPI_CLK
GPIO5 / VSPI_CS
GPIO17 / TX2
GPIO16 / RX2
GPIO4 / RTC_GPIO10 / ADC2_0 / TOUCH_0 / SD_DAT1
GPIO0 / RTC_GPIO11 / ADC2_1 / TOUCH_1
GPIO2 / RTC_GPIO12 / ADC2_2 / TOUCH_2 / SD_DAT0 / LED
GPIO15 / RTC_GPIO13 / ADC2_3 / TOUCH_3 / SD_CMD / HSPI_CS
GPIO8 / SDI/SD1
GPIO7 / SDO/SD0
GPIO6 / SCK/CLK
Do not Connect (used by internal Flash)
2xSPI:VSPI & HSPI
2xADC:ADC1 & ADC2
DEEP SLEEP GPIO
TOUCH SENSOR
I2C
DAC
SD
UART

In the next class we will discuss in detail the kit specifications. However, we will use the *GPIO23* and *GND* pins to blink an external LED for understanding how to interface ESP32.

a. Use your breadboard and place a LED and 220 Resistor as follows (the figure may have different connections-use the one mentioned above):



b. Write the following program



```
// Set our LED Pin to the GPIO Pin
const int ledPin = 23;

void setup()
{
  // setup ledPin AS output
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  // Turn on our LED
  digitalWrite(ledPin, HIGH);
  // Delay for some time
  delay(500);
  // Turn off our LED
  digitalWrite(ledPin, LOW);
  // Delay for some time
  delay(500);
}
```

Q6. Determine the output and record the video.

Q7. Now write a program that flipflops onboard LED and external LED with a delay of 10secs