**Assignment Completion Sheet**

|  |  |  |  |
| --- | --- | --- | --- |
| **SR. No** | **Assignment Name** | **Marks**  **(Out of 5)** | **Signature** |
| 1 | To write a program to sense the available networks using Arduino. |  |  |
| 2 | To write a program to measure the distance using ultrasonic sensor and make LED blink using Arduino. |  |  |
| 3 | To write a program to detects the vibration of an object with sensor using Arduino. |  |  |
| 4 | To write a program to sense a finger when it is placed on the board Arduino. |  |  |
| 5 | To write a program to connect with the available Wi-Fi using Arduino. |  |  |
| 6 | To write a program to get temperature notification using Arduino. |  |  |
| 7 | To write a program for LDR to vary the light intensity of LED using Arduino. |  |  |
| 8 | Start Raspberry Pi and try various Linux commands in command terminal window: ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron, chown, chgrp, pingetc. |  |  |
| 9 | Run some python programs on Pi like: a) Read your name and print Hello message with name b)Read two numbers and print their sum, difference, product and division. c) Word and character count of a given string. d)Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input. |  |  |
| 10 | Run some python programs on Pi like: a) Print a name 'n' times, where name and n are read from standard input, using for and while loops. b) Handle Divided by Zero Exception. c) Print current time for 10 times with an interval of10seconds. d) Read a fileline byline and print the word count of each line |  |  |
| 11 | Run some python programs on Pi like a) Light an LED through Python program b) Get input from two switches and switch on corresponding LEDs c) Flash an LED at a given on time and off time cycle, where the two times are taken from a file |  |  |
| Total out of 25 | |  |  |
| Total out of 05(Quiz/Viva) | |  |  |
| **Total (Out of 15)** | |  |  |

**This is to certify that Mr/Ms**  **University Exam Seat Number**  **has successfully completed the course work for CS-601-MJ (Internet Of Things) and has scored** **Marks out of**  **15.**

**Instructor Head**

**Internal Examiner External Examiner**

**Assignment 1**

**Assignment Name:** To write a program to sense the available networks using Arduino.

#include "WiFi.h" // For ESP32

// #include "ESP8266WiFi.h" // Uncomment this if you are using ESP8266

void setup() {

Serial.begin(115200);

// Start WiFi in station mode

WiFi.mode(WIFI\_STA);

WiFi.disconnect(); // Disconnect from any previous connections

delay(1000);

Serial.println("Scanning for available networks...");

}

void loop() {

// Perform WiFi network scan

int n = WiFi.scanNetworks();

if (n == 0) {

Serial.println("No networks found.");

} else {

Serial.println();

Serial.println("Networks found: ");

for (int i = 0; i < n; i++) {

// Print details of each network

Serial.print(i + 1);

Serial.print(": ");

Serial.print(WiFi.SSID(i)); // SSID of the network

Serial.print(" (");

Serial.print(WiFi.RSSI(i)); // Signal strength (RSSI)

Serial.print(" dBm) ");

Serial.print(" [");

Serial.print(WiFi.encryptionType(i)); // Encryption type

Serial.println("]");

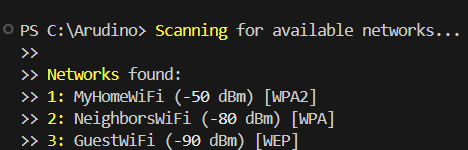
delay(10);

}

}

delay(5000);

}



**Assignment 2**

**Assignment Name: To write a program to measure the distance using ultrasonic sensor and make LED blink using Arduino.**

// Define pins for Ultrasonic sensor

const int trigPin = 9;

const int echoPin = 10;

// Define pin for LED

const int ledPin = 13;

// Variables to store the duration and distance

long duration;

int distance;

void setup() {

// Set trigPin as output and echoPin as input

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

// Set ledPin as output

pinMode(ledPin, OUTPUT);

// Start serial communication for debugging

Serial.begin(9600);

}

void loop() {

// Send a 10us pulse to the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

// Read the time for the pulse to return from the echoPin

duration = pulseIn(echoPin, HIGH);

// Calculate the distance in cm

distance = duration \* 0.034 / 2;

// Print distance to the serial monitor

Serial.print("Distance: ");

Serial.print(distance);

Serial.println(" cm");

// If the distance is less than 10 cm, blink the LED

if (distance < 10) {

digitalWrite(ledPin, HIGH); // Turn the LED on

delay(500); // Wait for 500ms

digitalWrite(ledPin, LOW); // Turn the LED off

delay(500); // Wait for 500ms

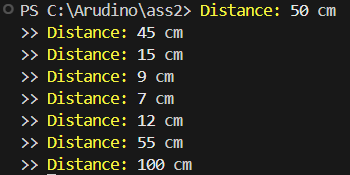
} else {

digitalWrite(ledPin, LOW); // Keep LED off if distance is greater than 10 cm

}

delay(1000); // Wait before next measurement

}



**Assignment 3**

**Assignment Name:** To write a program to detects the vibration of an object with sensor using Arduino.

**// Define the pins for the sensor, LED, and buzzer**

**int vibrationSensorPin = 2; // Vibration sensor is connected to digital pin 2**

**int ledPin = 13; // LED is connected to digital pin 13**

**int buzzerPin = 12; // Buzzer is connected to digital pin 12**

**void setup() {**

**pinMode(vibrationSensorPin, INPUT);**

**pinMode(ledPin, OUTPUT);**

**pinMode(buzzerPin, OUTPUT);**

**Serial.begin(9600);**

**}**

**void loop() {**

**// Read the state of the vibration sensor**

**int sensorValue = digitalRead(vibrationSensorPin);**

**// If the sensor detects vibration**

**if (sensorValue == HIGH) {**

**// Turn on the LED**

**digitalWrite(ledPin, HIGH);**

**// Turn on the buzzer**

**digitalWrite(buzzerPin, HIGH);**

**// Print message to the serial monitor**

**Serial.println("Vibration detected!");**

**} else {**

**// Turn off the LED**

**digitalWrite(ledPin, LOW);**

**// Turn off the buzzer**

**digitalWrite(buzzerPin, LOW);**

**// Print message to the serial monitor**

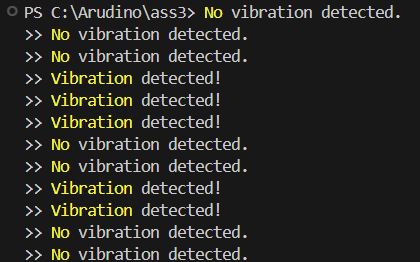
**Serial.println("No vibration detected.");**

**}**

**// Small delay for stability**

**delay(100);**

**}**



**Assignment 4**

**Assignment Name: To write a program to sense a finger when it is placed on the board Arduino.**

// Pin for the touch sensor

const int touchPin = 7;

const int ledPin = 13; // Built-in LED pin

void setup() {

// Initialize serial communication at 9600 baud rate

Serial.begin(9600);

// Set the touch pin as input

pinMode(touchPin, INPUT);

// Set LED pin as output

pinMode(ledPin, OUTPUT);

}

void loop() {

// Read the touch sensor value

int sensorValue = digitalRead(touchPin);

// Check if the sensor is touched

if (sensorValue == HIGH) {

Serial.println("Finger detected!");

digitalWrite(ledPin, HIGH); // Turn on LED when touched

} else {

Serial.println("No touch detected.");

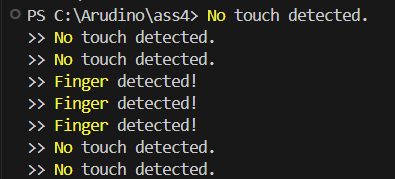
digitalWrite(ledPin, LOW); // Turn off LED when not touched

}

// Small delay to avoid excessive serial prints

delay(200);

}



**Assignment 5**

**Assignment Name: To write a program to connect with the available Wi-Fi using Arduino.**

  #include <ESP8266WiFi.h> // Include the ESP8266 library

// Replace these with your Wi-Fi credentials

const char\* ssid = "YOUR\_SSID";

const char\* password = "YOUR\_PASSWORD";

void setup() {

// Start serial communication for debugging

Serial.begin(115200);

delay(10);

// Connect to Wi-Fi

Serial.println();

Serial.println("Connecting to Wi-Fi...");

// Begin Wi-Fi connection

WiFi.begin(ssid, password);

// Wait until the connection is successful

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.print(".");

}

// Once connected, print the IP address

Serial.println();

Serial.println("Connected to Wi-Fi!");

Serial.print("IP Address: ");

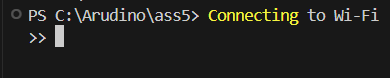
Serial.println(WiFi.localIP()); // Print the ESP8266 IP address

}

void loop() {

// Empty loop; the program only needs to connect to Wi-Fi and display the IP address

}



**Assignment 6**

**Assignment Name: To write a program to get temperature notification using Arduino.**

#include "DHT.h"

#define DHTPIN 2 // Pin where the DHT sensor is connected

#define DHTTYPE DHT11 // DHT11 or DHT22

#define LEDPIN 13 // Pin where LED is connected

// Set up the DHT sensor

DHT dht(DHTPIN, DHTTYPE);

float thresholdTemperature = 30.0; // Notification threshold

void setup() {

// Initialize serial communication for debugging

Serial.begin(9600);

// Initialize DHT sensor

dht.begin();

// Initialize LED pin as output

pinMode(LEDPIN, OUTPUT);

}

void loop() {

// Wait a few seconds between measurements

delay(2000);

// Reading temperature or humidity takes about 250ms!

float temperature = dht.readTemperature(); // Get temperature in Celsius

// Check if the reading is valid

if (isnan(temperature)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

// Print temperature to Serial Monitor

Serial.print("Temperature: ");

Serial.print(temperature);

Serial.println(" °C");

// Check if the temperature exceeds the threshold

if (temperature > thresholdTemperature) {

Serial.println("Temperature exceeded threshold! Turning ON notification LED.");

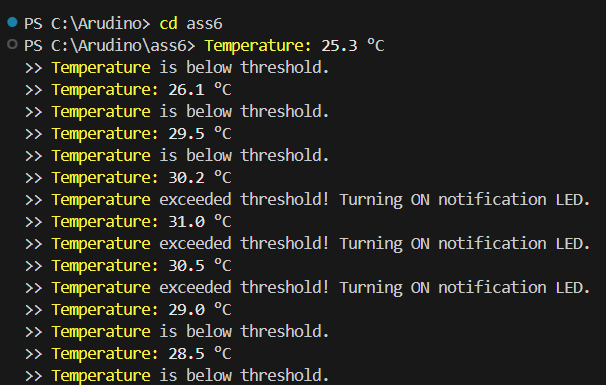
digitalWrite(LEDPIN, HIGH); // Turn on LED

} else {

Serial.println("Temperature is below threshold.");

digitalWrite(LEDPIN, LOW); // Turn off LED

} }



**Assignment 7**

**Assignment Name: To write a program for LDR to vary the light intensity of LED using Arduino.**

// Pin Definitions

const int ldrPin = A0; // LDR connected to analog pin A0

const int ledPin = 9; // LED connected to digital pin 9

void setup() {

// Initialize the LED pin as an output

pinMode(ledPin, OUTPUT);

// Begin Serial Communication for debugging

Serial.begin(9600);

}

void loop() {

// Read the value from the LDR

int ldrValue = analogRead(ldrPin);

// Map the LDR value to a range suitable for PWM (0-255)

int ledBrightness = map(ldrValue, 0, 1023, 0, 255);

// Set the brightness of the LED

analogWrite(ledPin, ledBrightness);

// Print the LDR value and the LED brightness for debugging

Serial.print("LDR Value: ");

Serial.print(ldrValue);

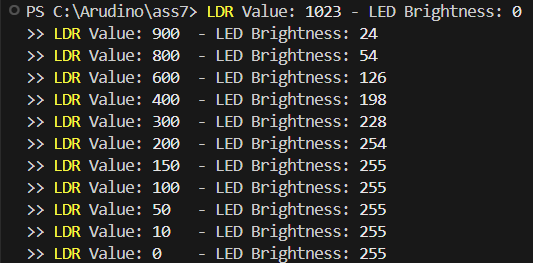
Serial.print(" - LED Brightness: ");

Serial.println(ledBrightness);

// Small delay for stability

delay(100);

}



**Assignment 8**

**Assignment Name :- Start Raspberry Pi and try various Linux commands in command terminal window: ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron, chown, chgrp, pingetc.**

**List directory contents.**

**ls**

**Change the current directory.**

cd /home/pi

**Create a new empty file.**

touch testfile.txt

**Move or rename files or directories.**

mv testfile.txt newfile.txt

**Remove a file.**

rm newfile.txt

**Display the manual for a command.**

man ls

**Create a new directory.**

mkdir newfolder

**Remove an empty directory.**

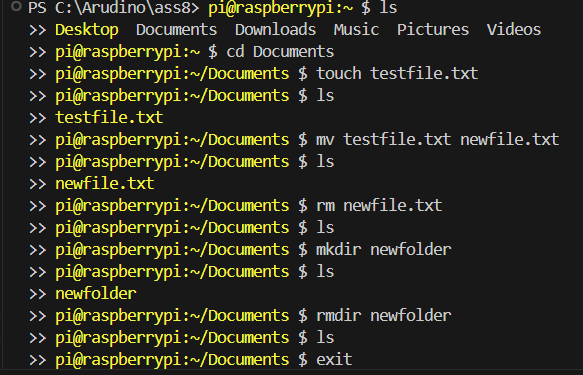
rmdir newfolder

**Archive files.**

tar -cvf archive.tar file1.txt file2.txt

**Schedule tasks**

crontab -e



**Assignment 9**

**Assignment Name: Run some python programs on Pi like: a) Read your name and print Hello message with name b)Read two numbers and print their sum, difference, product and division. c) Word and character count of a given string. d)Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input.**

# Program to read name and print a Hello message

name = input("Enter your name: ")

print(f"Hello, {name}!")

# Program to read two numbers and perform arithmetic operations

num1 = float(input("Enter first number: "))

num2 = float(input("Enter second number: "))

sum\_result = num1 + num2

difference = num1 - num2

product = num1 \* num2

division = num1 / num2 if num2 != 0 else "undefined (cannot divide by zero)"

print(f"Sum: {sum\_result}")

print(f"Difference: {difference}")

print(f"Product: {product}")

print(f"Division: {division}")

# Program to count words and characters in a string

input\_string = input("Enter a string: ")

word\_count = len(input\_string.split())

character\_count = len(input\_string)

print(f"Word Count: {word\_count}")

print(f"Character Count: {character\_count}")

# Program to calculate the area of a given shape

shape = input("Enter the shape (rectangle/triangle/circle): ").lower()

if shape == "rectangle":

length = float(input("Enter the length: "))

width = float(input("Enter the width: "))

area = length \* width

print(f"Area of the rectangle: {area}")

elif shape == "triangle":

base = float(input("Enter the base: "))

height = float(input("Enter the height: "))

area = 0.5 \* base \* height

print(f"Area of the triangle: {area}")

elif shape == "circle":

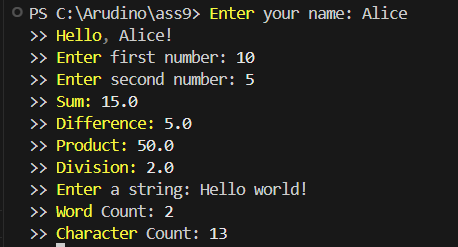
radius = float(input("Enter the radius: "))

area = 3.14159 \* radius \*\* 2

print(f"Area of the circle: {area}")

else:

print("Invalid shape")



**Assignment 10**

**Assignment Name: Run some python programs on Pi like: a) Print a name 'n' times, where name and n are read from standard input, using for and while loops. b) Handle Divided by Zero Exception. c) Print current time for 10 times with an interval of10seconds. d) Read a fileline byline and print the word count of each line**

# Function to print a name 'n' times using for loop

def print\_name\_for\_loop():

name = input("Enter a name: ")

n = int(input("Enter how many times to print the name: "))

print("\nUsing for loop:")

for \_ in range(n):

print(name)

# Function to print a name 'n' times using while loop

def print\_name\_while\_loop():

name = input("Enter a name: ")

n = int(input("Enter how many times to print the name: "))

count = 0

print("\nUsing while loop:")

while count < n:

print(name)

count += 1

print\_name\_for\_loop()

print\_name\_while\_loop()

def divide\_numbers():

try:

num1 = float(input("Enter the numerator: "))

num2 = float(input("Enter the denominator: "))

result = num1 / num2

print(f"The result of {num1} divided by {num2} is: {result}")

except ZeroDivisionError:

print("Error: Cannot divide by zero!")

divide\_numbers()

import time

from datetime import datetime

def print\_current\_time():

for \_ in range(10):

current\_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

print(f"Current time: {current\_time}")

time.sleep(10)

print\_current\_time()

def word\_count\_in\_file(filename):

try:

with open(filename, 'r') as file:

print("\nWord count for each line:")

for line\_number, line in enumerate(file, start=1):

word\_count = len(line.split())

print(f"Line {line\_number}: {word\_count} words")

except FileNotFoundError:

print("Error: The file does not exist!")

# Create a sample file for demonstration

sample\_file = 'sample.txt'

with open(sample\_file, 'w') as file:

file.write("Hello world\n")

file.write("This is a test file.\n")

file.write("Python is fun!\n")

word\_count\_in\_file(sample\_file)

def word\_count\_in\_file(filename):

try:

with open(filename, 'r') as file:

print("\nWord count for each line:")

for line\_number, line in enumerate(file, start=1):

word\_count = len(line.split())

print(f"Line {line\_number}: {word\_count} words")

except FileNotFoundError:

print("Error: The file does not exist!")

# Create a sample file for demonstration

sample\_file = 'sample.txt'

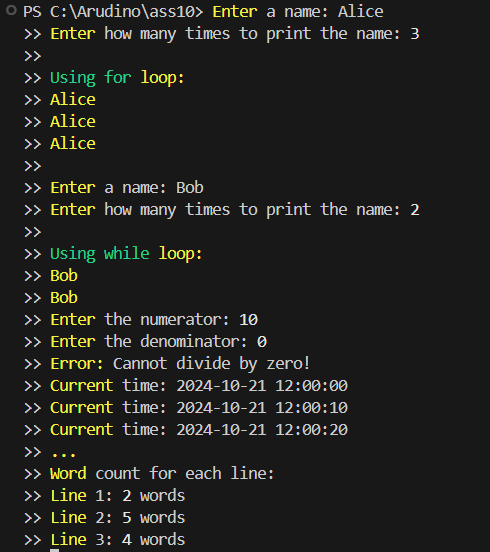
with open(sample\_file, 'w') as file:

file.write("Hello world\n")

file.write("This is a test file.\n")

file.write("Python is fun!\n")

word\_count\_in\_file(sample\_file)



**Assignment 11**

**Assignment Name: Run some python programs on Pi like a) Light an LED through Python program b) Get input from two switches and switch on corresponding LEDs c) Flash an LED at a given on time and off time cycle, where the two times are taken from a file**

import RPi.GPIO as GPIO

import time

# Set up the GPIO pin for the LED

LED\_PIN = 18 # Use the GPIO pin you connected the LED to

GPIO.setmode(GPIO.BCM) # Use Broadcom pin numbering

GPIO.setup(LED\_PIN, GPIO.OUT) # Set the LED pin as output

# Light up the LED

GPIO.output(LED\_PIN, GPIO.HIGH)

print("LED is ON")

time.sleep(5) # Keep the LED on for 5 seconds

# Turn off the LED

GPIO.output(LED\_PIN, GPIO.LOW)

print("LED is OFF")

# Clean up

GPIO.cleanup()

import RPi.GPIO as GPIO

import time

# Set up GPIO pins

SWITCH1\_PIN = 17 # First switch pin

SWITCH2\_PIN = 27 # Second switch pin

LED1\_PIN = 18 # LED for the first switch

LED2\_PIN = 22 # LED for the second switch

GPIO.setmode(GPIO.BCM) # Use Broadcom pin numbering

GPIO.setup(SWITCH1\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

GPIO.setup(SWITCH2\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)

GPIO.setup(LED1\_PIN, GPIO.OUT)

GPIO.setup(LED2\_PIN, GPIO.OUT)

try:

while True:

if GPIO.input(SWITCH1\_PIN) == GPIO.HIGH:

GPIO.output(LED1\_PIN, GPIO.HIGH) # Turn on LED1

print("LED1 is ON")

else:

GPIO.output(LED1\_PIN, GPIO.LOW) # Turn off LED1

print("LED1 is OFF")

if GPIO.input(SWITCH2\_PIN) == GPIO.HIGH:

GPIO.output(LED2\_PIN, GPIO.HIGH) # Turn on LED2

print("LED2 is ON")

else:

GPIO.output(LED2\_PIN, GPIO.LOW) # Turn off LED2

print("LED2 is OFF")

time.sleep(0.1) # Small delay for debounce

except KeyboardInterrupt:

print("Exiting...")

finally:

GPIO.cleanup()

import RPi.GPIO as GPIO

import time

# Set up the GPIO pin for the LED

LED\_PIN = 18 # Use the GPIO pin you connected the LED to

GPIO.setmode(GPIO.BCM) # Use Broadcom pin numbering

GPIO.setup(LED\_PIN, GPIO.OUT) # Set the LED pin as output

# Read timings from the file

with open('timings.txt', 'r') as file:

on\_time, off\_time = map(int, file.read().strip().split())

try:

while True:

GPIO.output(LED\_PIN, GPIO.HIGH) # Turn on LED

print(f"LED is ON for {on\_time} seconds")

time.sleep(on\_time)

GPIO.output(LED\_PIN, GPIO.LOW) # Turn off LED

print(f"LED is OFF for {off\_time} seconds")

time.sleep(off\_time)

except KeyboardInterrupt:

print("Exiting...")

finally:

GPIO.cleanup()

