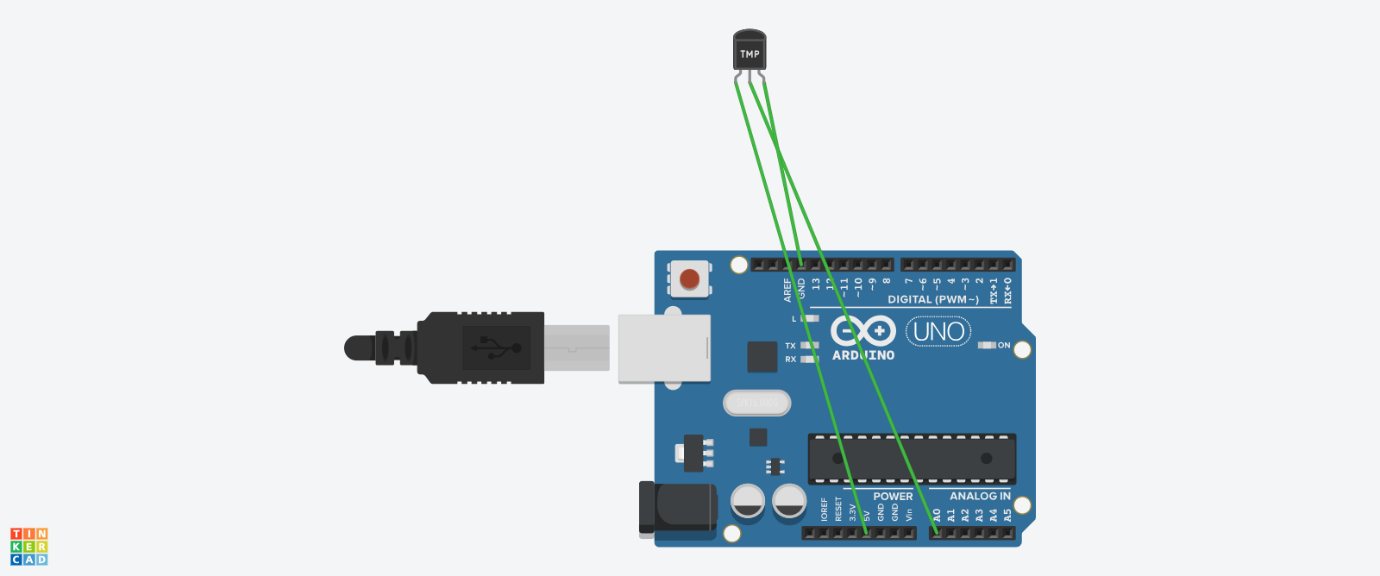
1. Temprature Senser (slip no .6 , 17 thinkerCad)



Code –

int baselineTemp = 0;

int celsius = 0;

int fahrenheit = 0;

void setup()

{

pinMode(A0, INPUT);

Serial.begin(9600);

}

void loop()

{

baselineTemp = 40;

celsius = map(((analogRead(A0) - 20) \* 3.04), 0, 1023, -40, 125);

fahrenheit = ((celsius \* 9) / 5 + 32);

Serial.print(celsius);

Serial.print(" C, ");

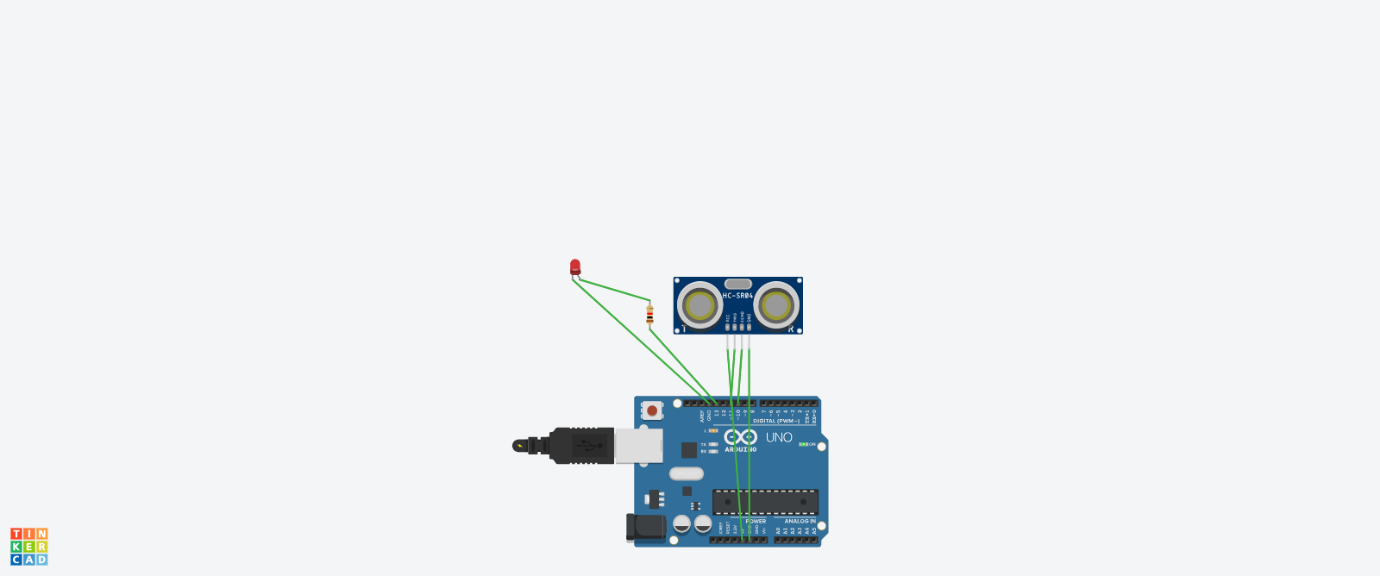
Serial.print(fahrenheit);

Serial.println(" F");

delay(1000);

}

2.Ultrasonic sensor (slip no 2 and 13 , thinkerCad )



Code -

// this constant won't change. It's the pin number

// of the sensor's output:

const int pingPin = 7;

const int ledPin = 13;

void setup() {

// initialize serial communication:

Serial.begin(9600);

pinMode(ledPin, OUTPUT);

}

void loop() {

// establish variables for duration of the ping,

// and the distance result in inches and centimeters:

long duration, cm;

// The PING))) is triggered by a HIGH pulse of 2 or more microseconds.

// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:

pinMode(pingPin, OUTPUT);

digitalWrite(pingPin, LOW);

delayMicroseconds(2);

digitalWrite(pingPin, HIGH);

delayMicroseconds(5);

digitalWrite(pingPin, LOW);

// The same pin is used to read the signal from the PING))): a HIGH

// pulse whose duration is the time (in microseconds) from the sending

// of the ping to the reception of its echo off of an object.

pinMode(pingPin, INPUT);

duration = pulseIn(pingPin, HIGH);

// convert the time into a distance

cm = microsecondsToCentimeters(duration);

// Print the distance

Serial.print("Distance: ");

Serial.print(cm);

Serial.print("cm");

Serial.println();

// Turn on the LED if the object is too close:

if(cm < 100) {

digitalWrite(ledPin, HIGH);

}

else {

digitalWrite(ledPin, LOW);

}

delay(100);

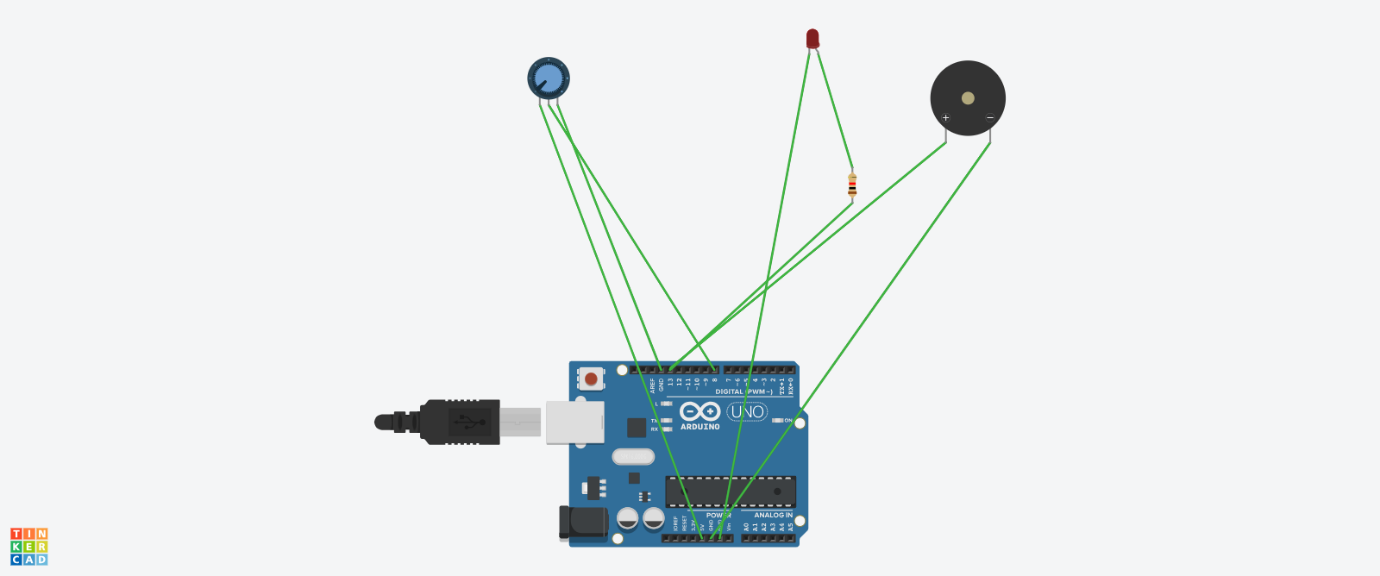
}

long microsecondsToCentimeters(long microseconds) {

return microseconds / 29 / 2;

}

3.Vibration Detatector (Slip no 3, 14 )



Code –

//Define pin numbers

#define VIBRATION\_PIN 8 // Digital pin connected to the Vibration Sensor

#define BUZZER\_PIN 13 // Digital pin connected to the Buzzer

int buzzerDurationMillis = 1000; // Duration for which the buzzer will be active (in milliseconds)

void setup() {

// Initialize the Vibration Sensor pin as an input

pinMode(VIBRATION\_PIN, INPUT);

// Initialize the Buzzer pin as an output

pinMode(BUZZER\_PIN, OUTPUT);

}

void loop() {

// Read the value from the Vibration Sensor

if (digitalRead(VIBRATION\_PIN) == HIGH) {

// Turn the buzzer on

digitalWrite(BUZZER\_PIN, HIGH);

// Keep the buzzer on for the specified duration

delay(buzzerDurationMillis);

// Turn the buzzer off

digitalWrite(BUZZER\_PIN, LOW);

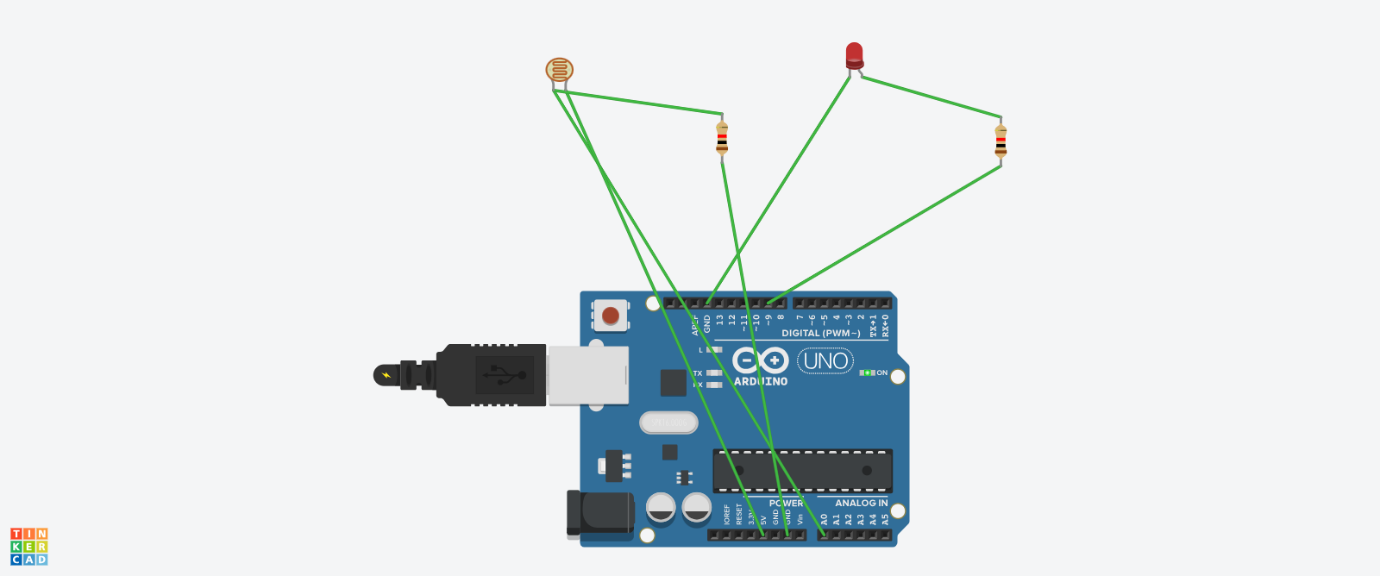
}

// Small delay to avoid rapid triggering

delay(100);

}

4. LDR (instead of LDR Use Photoresistor) –(slip no – 7 , 18 )



Code –

// Pin definitions

const int ldrPin = A0; // Analog pin connected to the LDR

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

void setup() {

// Initialize the LED pin as an output

pinMode(ledPin, OUTPUT);

// Initialize the LDR pin as an input (it's an analog pin by default)

pinMode(ldrPin, INPUT);

}

void loop() {

// Read the LDR value (analog input 0-1023)

int ldrValue = analogRead(ldrPin);

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

// The LDR value is in the range 0-1023, so map it to 0-255

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

// Set the LED brightness

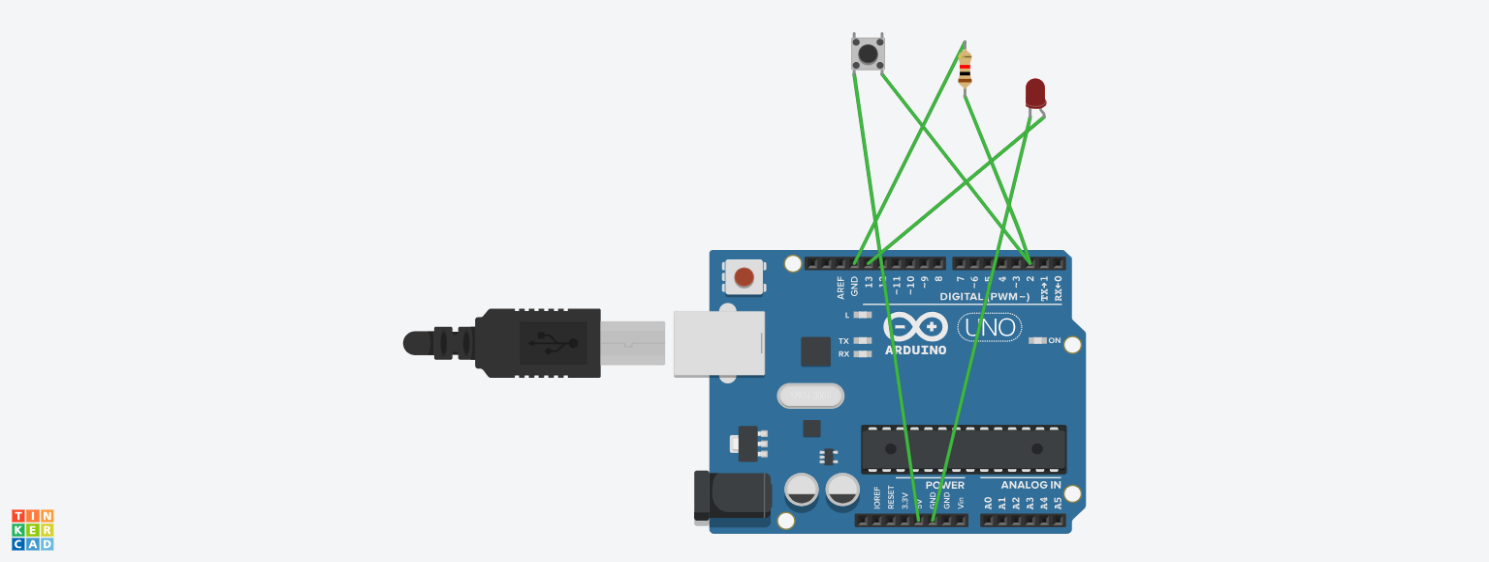
analogWrite(ledPin, ledBrightness);

// Short delay for stability

delay(50);

}

5. Finger Detection System Using a Push Button(slip no=4,15)



Code –

const int buttonPin = 2; // Pin connected to the push button

const int ledPin = 13; // Onboard LED pin (or external LED)

void setup() {

pinMode(buttonPin, INPUT); // Set the button pin as an input

pinMode(ledPin, OUTPUT); // Set the LED pin as an output

Serial.begin(9600); // Start serial communication for debugging

}

void loop() {

int buttonState = digitalRead(buttonPin); // Read the state of the button

// If the button is pressed (simulating finger detection)

if (buttonState == HIGH) {

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

Serial.println("Finger Detected!"); // Print to Serial Monitor

} else {

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

Serial.println("No Finger.");

}

delay(100); // Small delay to prevent flooding the serial monitor

}

6.(Thonny Use kara):- slip no= 9,20

Code-

Run some python programs on Pi like:

**a) Read your name and print Hello message with name**

name = input("Please enter your name: ")

# Print a hello message with the name

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

**b) Read two numbers and print their sum, difference, product and division.**

# Read two numbers from the user

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

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

# Calculate sum, difference, product, and division

sum\_result = num1 + num2

difference = num1 - num2

product = num1 \* num2

# Handle division by zero

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

# Print the results

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

print(f"Difference: {difference}")

print(f"Product: {product}")

print(f"Division: {division}")

**c) Word and character count of a given string.**

# Read a string from the user

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

# Count words and characters

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

character\_count = len(input\_string)

# Print the results

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

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

**d)Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input.**

import math

# Get the shape from the user

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 = math.pi \* radius \*\* 2

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

7.(Thonny Use kara) = Slip no 10

Code-

**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.**

**# Read the name and the number of times to print**

**name = input("Enter the name: ")**

**n = int(input("Enter the number of times to print the name: "))**

**# Using a for loop**

**print("\nUsing for loop:")**

**for \_ in range(n):**

**print(name)**

**# Using a while loop**

**print("\nUsing while loop:")**

**count = 0**

**while count < n:**

**print(name)**

**count += 1**

**b) Handle Divided by Zero Exception.**

**# Function to perform division**

**def divide\_numbers():**

**try:**

**# Read two numbers from the user**

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

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

**# Perform division**

**result = num1 / num2**

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

**except ZeroDivisionError:**

**print("Error: Cannot divide by zero. Please enter a non-zero denominator.")**

**except ValueError:**

**print("Error: Please enter valid numbers.")**

**# Call the function**

**divide\_numbers()**

**c) Print current time for 10 times with an interval of10seconds.**

import time

from datetime import datetime

# Print current time 10 times with a 10-second interval

for \_ in range(10):

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

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

time.sleep(10) # Wait for 10 seconds

**d) Read a fileline byline and print the word count of each line**

**# test1.txt file save on same location were to save your .py file**

# File path to be read

filename = "test1.txt" # Replace with your actual file name

try:

# Open the file for reading

with open(filename, 'r', encoding='utf-8') as file:

# Read the file line by line

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

# Strip leading/trailing whitespace and check if the line is not empty

stripped\_line = line.strip()

if stripped\_line: # If the line is not empty

# Count the words in the current line by splitting on spaces

word\_count = len(stripped\_line.split())

# Print the word count of the current line

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

else:

# If the line is empty or just spaces, skip this line

continue # This will skip printing for empty lines

except FileNotFoundError:

print(f"Error: The file '{filename}' was not found.")

except Exception as e:

print("An error occurred:", e)

8.( proteous Use kara) = Slip no =11

**Raspberry pi to LED blink=**

* **STEP BY STEP:**

1.Open proteous

2.Go files

3.Go New project

4.Enter project name + Next

5.Next

6.Next

7.Select = Create Firmware Project

7.1. Go family

7.2 Select = Raspberry pi + Next

8.Click finish

9.Pahila code desel tyat ha code taka

**CODE**-

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

GPIO.setup(20,GPIO.OUT)

while True:

GPIO.output(20,GPIO.HIGH)

time.sleep(1)

GPIO.output(20,GPIO.LOW)

time.sleep(1)

10.In this same screen

10.1.Go left side

10.2.Right click on Peripherals

10.3.Go Add Peripherals

10.4.Category = Select = Breakout Peripherals

10.5.Click LED(Blue) + Add

11.Run Code