# Practical : 20

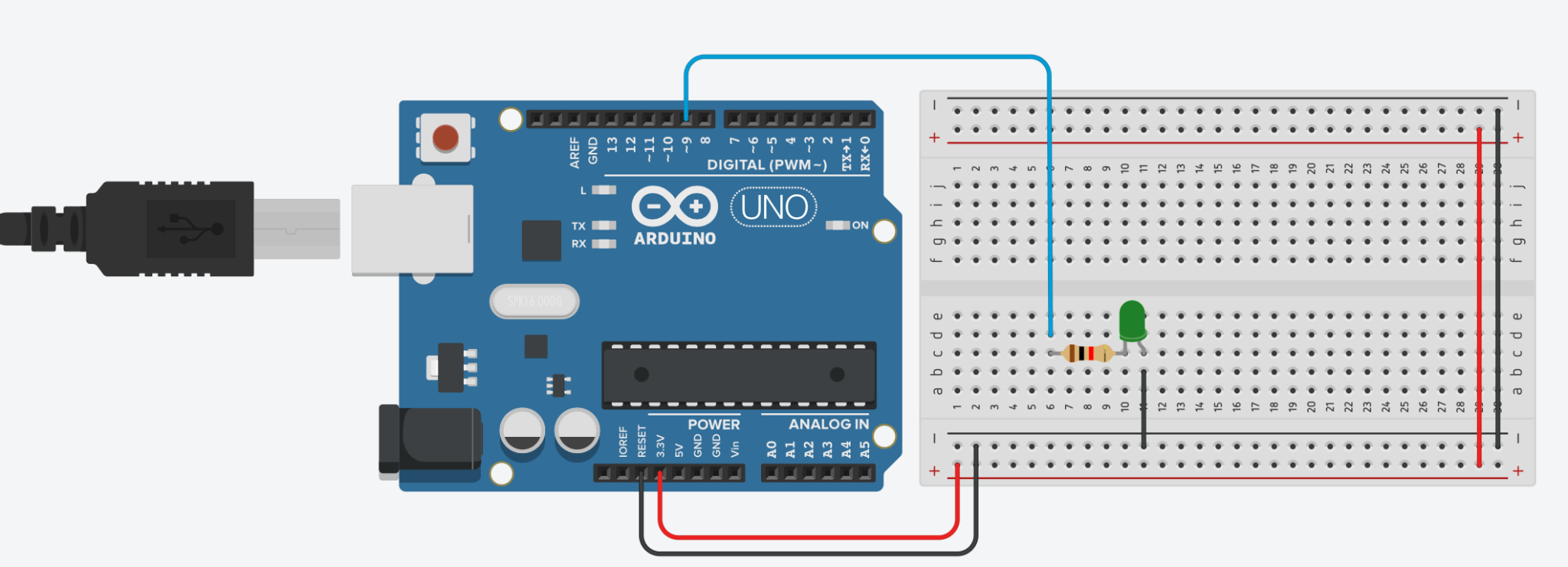
**Aim:** WAP to interface external LED with Breadboard & Arduino and write a program to Fade LED With Arduino Analog Output.

## Objectives:

* To learn Arduino UNO basics.
* To Learn Breadboard basics.
* Write a program to blink Arduino onboard LED and to interface external LED with Breadboard & Arduino.
* Write a program to Fade LED With Arduino Analog Output.

**Components:** Arduino UNO R3,Breadboard,1 Resistor,1 LED, Wires

## Circuit:

****

**Program Code:**

int brightness = 0; void setup()

{

pinMode(9, OUTPUT);

}

void loop()

{

for (brightness = 0; brightness <= 1000; brightness += 5)

{

analogWrite(9, brightness);

delay(50); // Wait for 30 millisecond(s)

}

for (brightness = 1000; brightness >= 0; brightness -= 5)

{

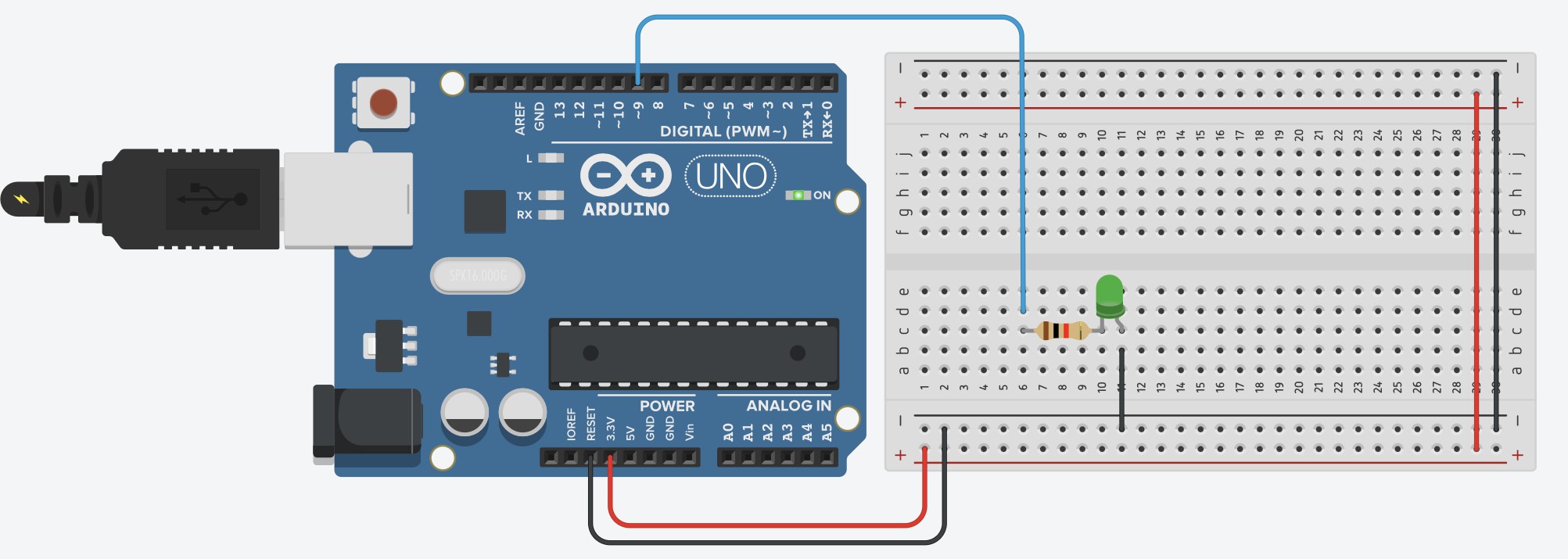
analogWrite(9, brightness);

delay(50); // Wait for 30 millisecond(s)

}

}

## Output:

****

**Conclusion:**

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard , LED’s and interfacing Fade LED With Arduino Analog Output with connections of resistors, breadboard, Jumper wires & Arduino.

# Practical : 21

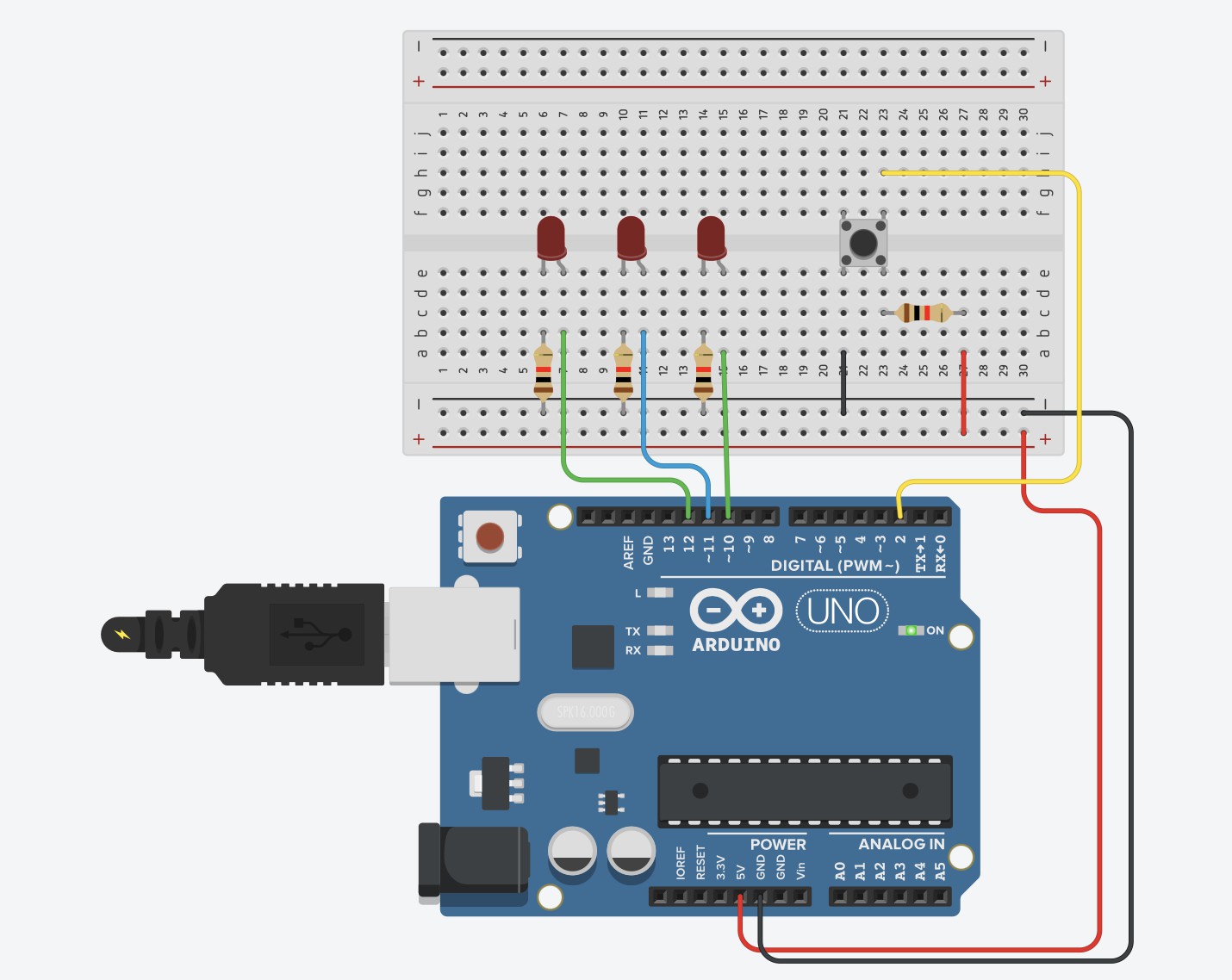
**Aim:** WAP to interface Push Button with Arduino and write a program to turn ON LEDs when push button is pressed and gets OFF when push button is pressed again.

## Objectives:

* To learn Arduino UNO basics.
* To Learn Breadboard basics.
* To Learn Push buttons / switches.
* Programming of interfacing Push Button with Arduino
* Write a program to turn ON LEDs when push button is pressed and gets OFF gets OFF when push button is pressed again.

**Components:** Arduino UNO R3,Breadboard,4 Resistor,3 LED, 1 Push Button, Wires

## Circuit:

****

**Program Code:**

int ledPin1 = 10; int ledPin2 = 11; int ledPin3 = 12; int keyPin = 2;

bool ledState = false; // Track LED ON/OFF state

bool lastButtonState = HIGH; // Store previous button state void setup()

{

pinMode(ledPin1, OUTPUT); pinMode(ledPin2, OUTPUT); pinMode(ledPin3, OUTPUT);

pinMode(keyPin, INPUT\_PULLUP); // Enable internal pull-up resistor

}

void loop()

{

bool currentButtonState = digitalRead(keyPin);

// Detect button press (falling edge: HIGH to LOW)

if (lastButtonState == HIGH && currentButtonState == LOW)

{

ledState = !ledState; // Toggle LED state digitalWrite(ledPin1, ledState ? HIGH : LOW); delay(500);

digitalWrite(ledPin2, ledState ? HIGH : LOW); delay(500);

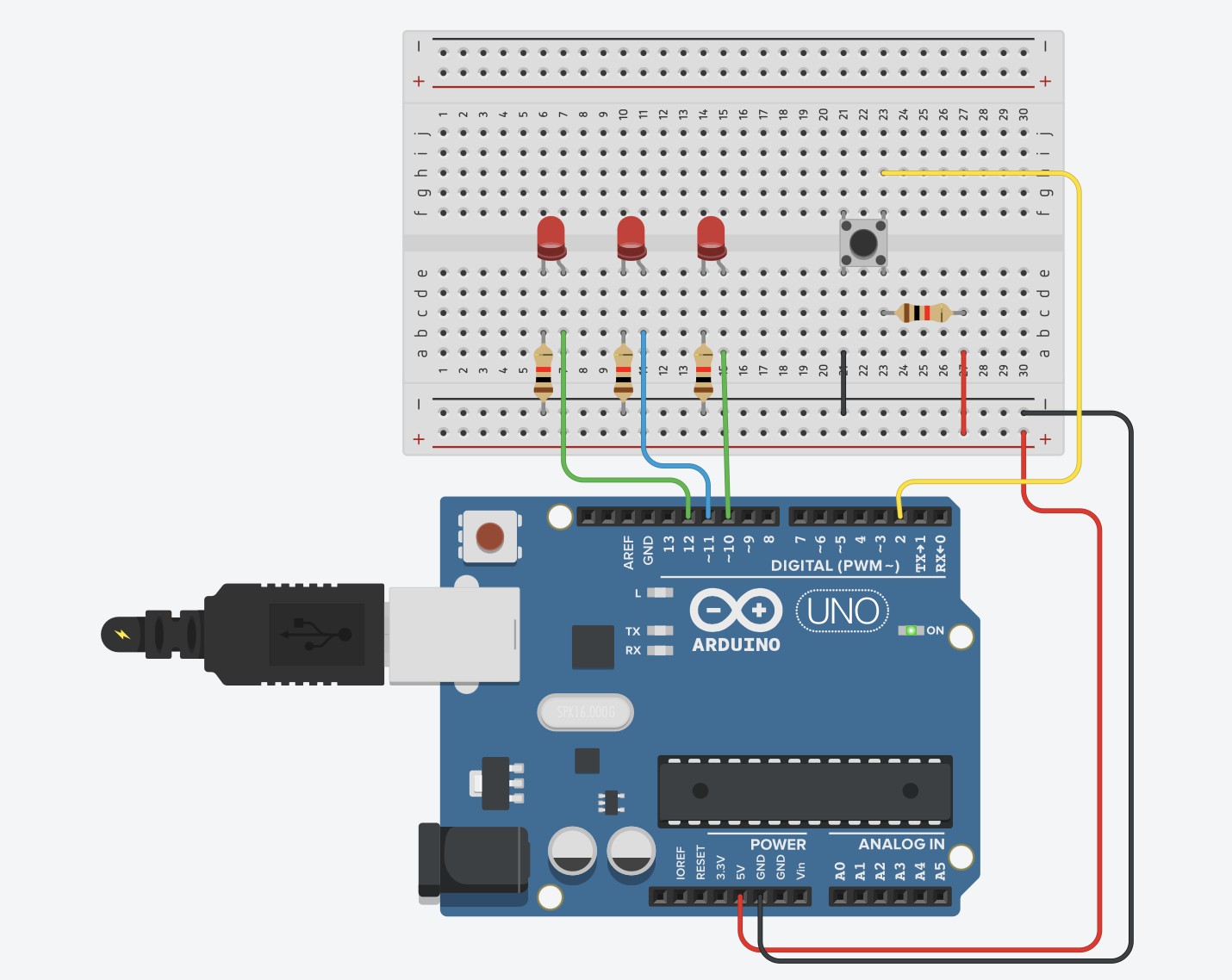
digitalWrite(ledPin3, ledState ? HIGH : LOW); delay(500);

}

lastButtonState = currentButtonState;

}

## Output:

****

**Conclusion:**

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard , LED’s and interfacing Fade LED With Arduino Analog Output with connections of resistors, breadboard, Jumper wires & Arduino.

# Practical No. 22

**AIM:** To interface 16×2 LCD, Push Button, Potentiometer with Arduino and WAP to display message on LCD when push button is pressed.

## Objectives:

* To learn Arduino UNO basics
* To Learn Breadboard basics
* To Learn about 16x2 LCD.
* To Learn about Potentiometer
* WAP to display message on LCD when push button is pressed.

**Components:** Arduino UNO R3,Breadboard,1 16x2 LCD,4 Resistor,1 Potentiometer, wires.

## Circuit:

**Program Code:**

// include the library code:

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup()

{

pinMode(A0, INPUT); pinMode(A1, INPUT); pinMode(A2, INPUT);

}

void loop()

{

// set the cursor to column 0, line 1

//(note: line 1 is the second row, since counting begins with 0):

lcd.setCursor(0, 1);

if (digitalRead(A0) == HIGH)

{

lcd.print( "VIT MCA");

}

if (digitalRead(A1) == HIGH)

{

lcd.print("Sub-IoT");

}

if (digitalRead(A2) == HIGH)

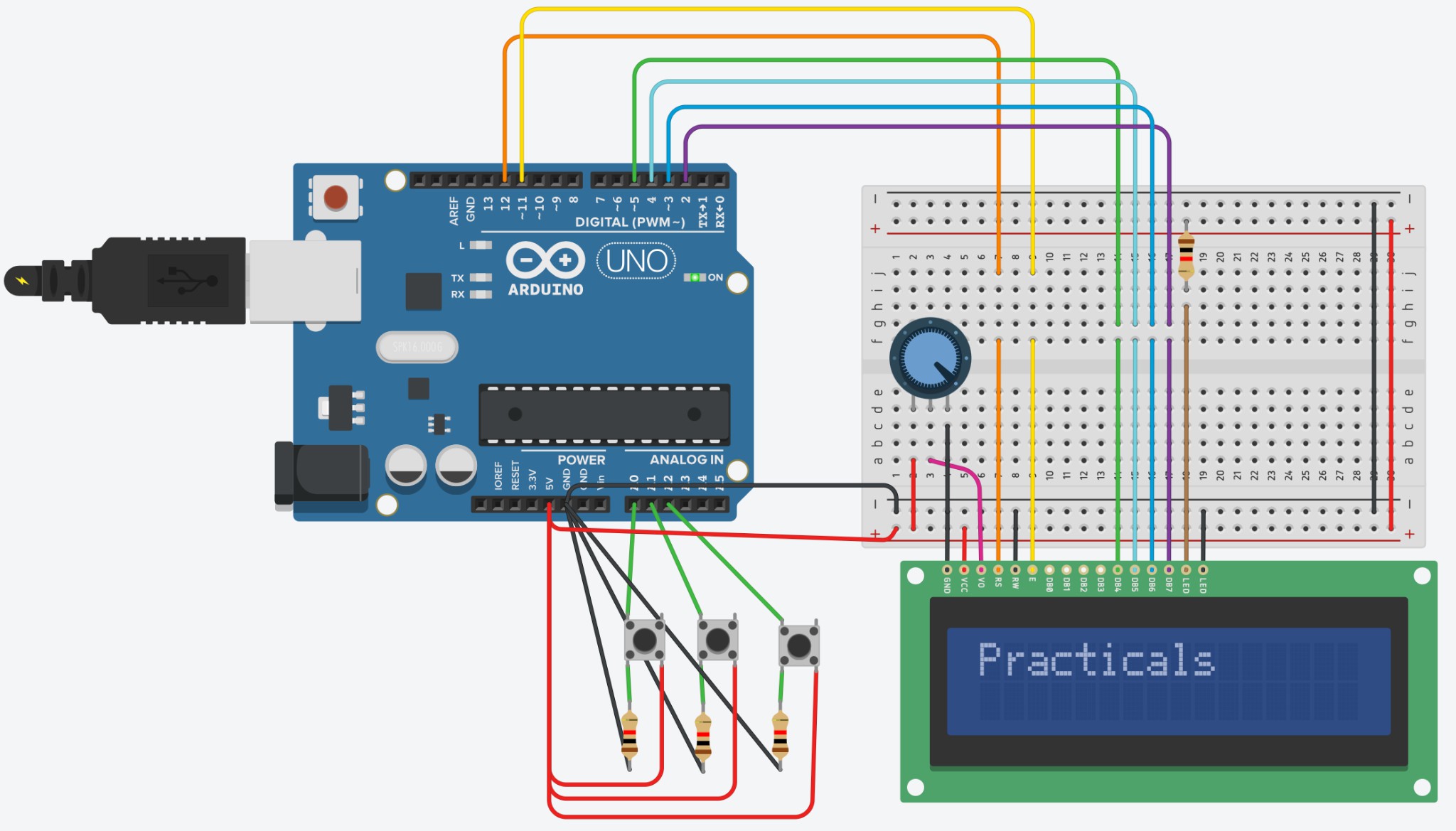
{

lcd.print("Practicals");

}

}

## Output:

****

**Conclusion:**

Thus, learnt about basic components of IoT like Arduino UNO (blink Arduino onboard LED), Breadboard , 7 segment display and interfacing 16×2 LCD, Push Button, Potentiometer with Arduino and displaying message on LCD when push button pressed with connections of resistors, breadboard, Jumper wires & Arduino.

# Practical No. 23

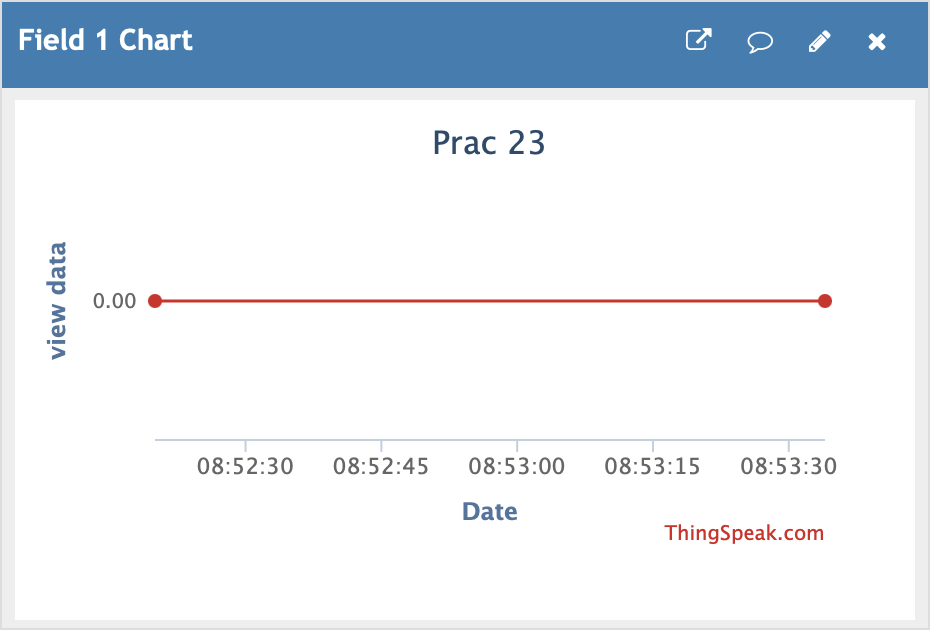
**Part 1 AIM:** To upload data on Thingspeak cloud **Manually.Steps:**

1. Go to Google and search for Thingspeak.
2. If you are new to Thingspeak, Do sign up and make sure you are on your Channel page.
3. Click on the NEW CHANNEL button (Green color) and create a new channel.
4. Enter a channel name, any description of your choice, and make sure one field is selected or ticked and give that field a name of your choice. Click on save.
5. Now in the private view, make sure you see a graph (empty).
6. Now click on API KEYS tab, scroll down to find API requests section and in that copy the link of Write a Channel Feed and paste it in the Address bar of your browser. And press enter to get a blank screen with a number which indicates the number of data uploaded manually. Following is the example link: https://api.thingspeak.com/update?api\_key=7JWB3X9H4O4VQGL7&field1=0
7. Suppose you want to change the data to be entered in the graph, just change the

8. =0 to any value of your choice in the link. Above is the link, where we had changed 0 to 40.

https://api.thingspeak.com/update?api\_key=7JWB3X9H4O4VQGL7&field1=40

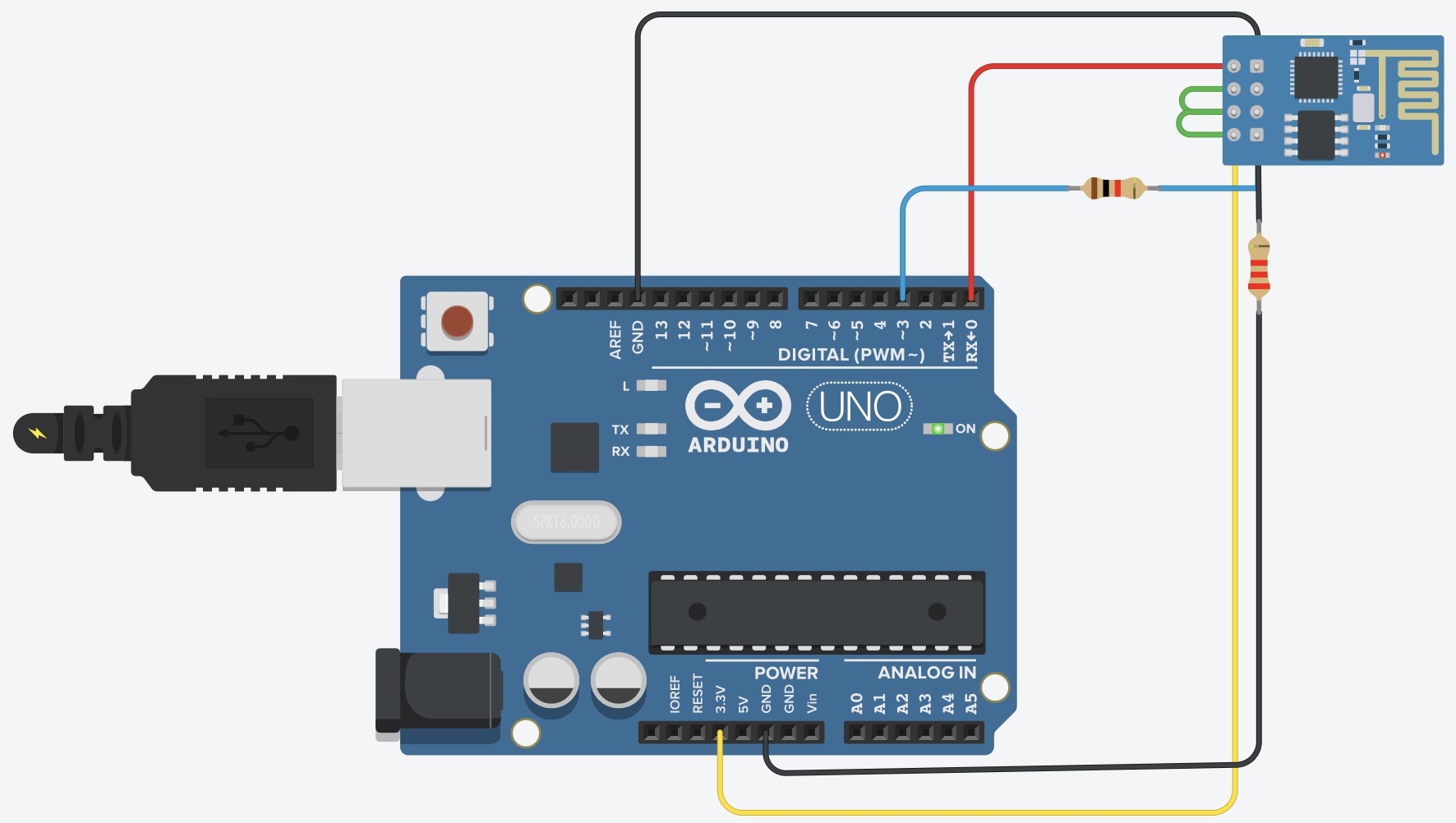
9. As a result, to see the visualization (graph), go to the private view and see the graph.



**Part 2 AIM:** To update readings to Thingspeak from Arduino **using Tinkercad.Steps:**

1. Click on the NEW CHANNEL button (Green colour) and create a new channel.
2. Enter a channel name, any description of your choice, and make sure one field is selected or tickedand give that field a name of your choice. Click on save.
3. Now in the API Keys tab copy the Write API Key and Paste it in your program

## Circuit:

****

**Program Code:**

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

Serial.println("AT+CWJAP=\"Simulator Wifi\",\"\"");

delay(5000); // wait for connection

}

void loop() {

Serial.println("AT+CIPSTART=\"TCP\",\"a pi.thingspeak.com\",80");

delay(3000);

String cmd = "GET

/update?api\_key=7JWB3X9H4O4VQGL7& field1=99 HTTP/1.1\r\n";

cmd += "Host: api.thingspeak.com\r\n"; cmd += "Connection: close\r\n\r\n";

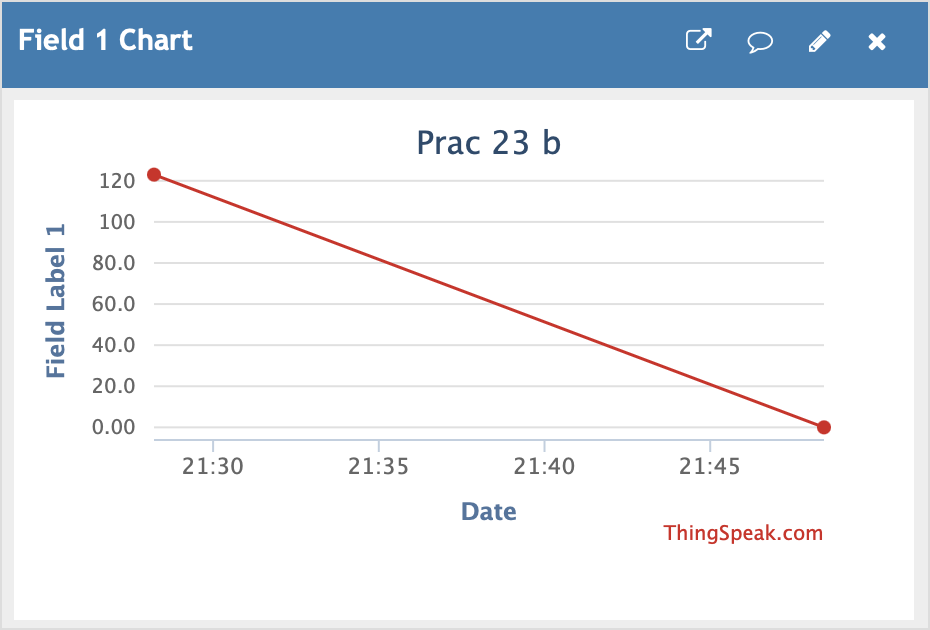
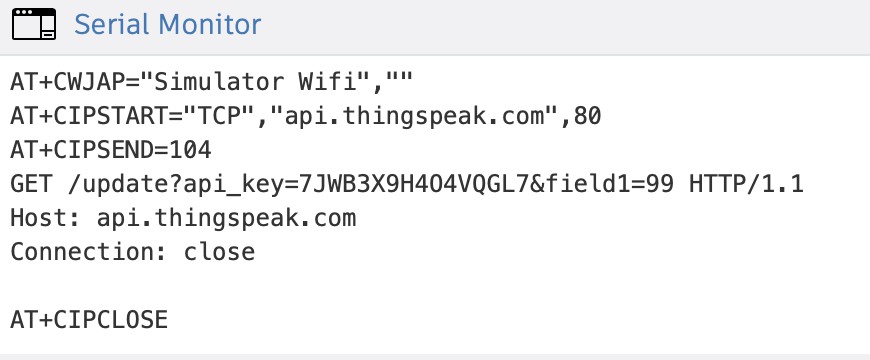
Serial.print("AT+CIPSEND="); Serial.println(cmd.length()); delay(2000);

Serial.print(cmd);

delay(5000); Serial.println("AT+CIPCLOSE"); delay(20000); // wait before sending again

}

## Output:

****

**NOTE:**

1. TEXT IN RED COLOUR IS YOUR WRITE API KEY
2. Make sure you see your Serial Monitor and check if Data is sent to Thingspeak.
3. To check result about data upload, go to thingspeak, click on private view and see the graph.

## Part 3

**Aim:** To interface Temperature sensor and ESP8266 with Arduino and update temperature reading toThingspeak.

## Steps:

1. Click on the NEW CHANNEL button (Green colour) and create a new channel.
2. Enter a channel name, any description of your choice, and make sure one field is selected or tickedand give that field a name of your choice. Click on save.
3. Now in the API Keys tab copy the Write API Key and Paste it in your program

## Circuit:

**PROGRAM Code:**

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

// Connect to Wi-Fi Serial.println("AT+CWJAP=\"Simulator

Wifi\",\"\"\r\n");

delay(3000); // Wait for connection

// Convert voltage to Celsius (assuming TMP36 sensor)

float tempC = (volt - 0.5) \* 100; // Adjust based on your sensor's characteristics

// Print the temperature to the Serial Monitor

Serial.println(tempC);

}

void loop() {

// Read the analog value from the sensor int sensorValue = analogRead(A0);

// Convert the analog value to voltage float volt = (sensorValue / 1024.0) \* 5.0;

// Use 1024 for 10-bit resolution and 5V reference

// Start TCP connection to ThingSpeak

Serial.println("AT+CIPSTART=\"TCP\",\"a pi.thingspeak.com\",80\r\n");

delay(5000); // Wait for connection to establish

// Prepare the HTTP GET request String cmd = "GET

/update?api\_key=63Z3U807XQRKWFWX

&field1=" + String(tempC) + " HTTP/1.1\r\n";

cmd += "Host: api.thingspeak.com\r\n"; cmd += "Connection: close\r\n\r\n";

// Send the length of the request

int len = cmd.length(); // Calculate the length of the command

Serial.print("AT+CIPSEND="); Serial.println(len);

delay(10); // Wait for the command to be processed

// Send the actual command Serial.print(cmd);

delay(100); // Wait for the request to be sent

// Close the TCP connection Serial.println("AT+CIPCLOSE\r\n"); delay(60000); // Wait before sending

again (60 seconds)

}

## Output:-

**NOTE:**

1. TEXT IN RED COLOUR IS YOUR WRITE API KEY
2. Make sure you see your Serial Monitor and check if Data is sent to Thingspeak.
3. To check result about data upload, go to thingspeak, click on private view and see the graph.

# Practical No. 24

**Aim:** To interface LDR sensor, LED and ESP8266 with Arduino and update light intensity values to Thingspeak and tweet “LIGHT ON” message on tweeter when light intensity value is less than 300.

## Steps:

1. Click on the NEW CHANNEL button (Green colour) and create a new channel.
2. Enter a channel name, any description of your choice, and make sure one field is selected or tickedand give that field a name of your choice. Click on save.
3. Now in the API Keys tab copy the Write API Key and Paste it in your program

## Circuit:-

**PROGRAM:**

int ldr=A0;//Set A0(Analog Input) for LDR. int value=0;

void setup()

{

Serial.begin(115200); pinMode(13,OUTPUT); delay(1000);

Serial.println("AT+CWJAP=\"Simulator Wifi\",\"\"\r\n");delay(3000);

}

void loop()

{

{

value=analogRead(ldr); Serial.println("LDR value is :"); Serial.println(value); if(value<300)

{

digitalWrite(13,HIGH);

}

else

{

digitalWrite(13,LOW);//Turns the LED OFF in Light.

}

Serial.println("AT+CIPSTART=\"TCP\",\"th ingspeak.com\",80");delay(5000);

int len = 65; Serial.print("AT+CIPSEN D=");Serial.println(len); delay(10);

Serial.print("GET/update?api\_key=6WEDQNF N3GBKNCQ3&field1="+String(value)

+"HTTP/1.1\r\n");

delay(100); Serial.println("AT+CIPCLOSE=0\r\n");dela

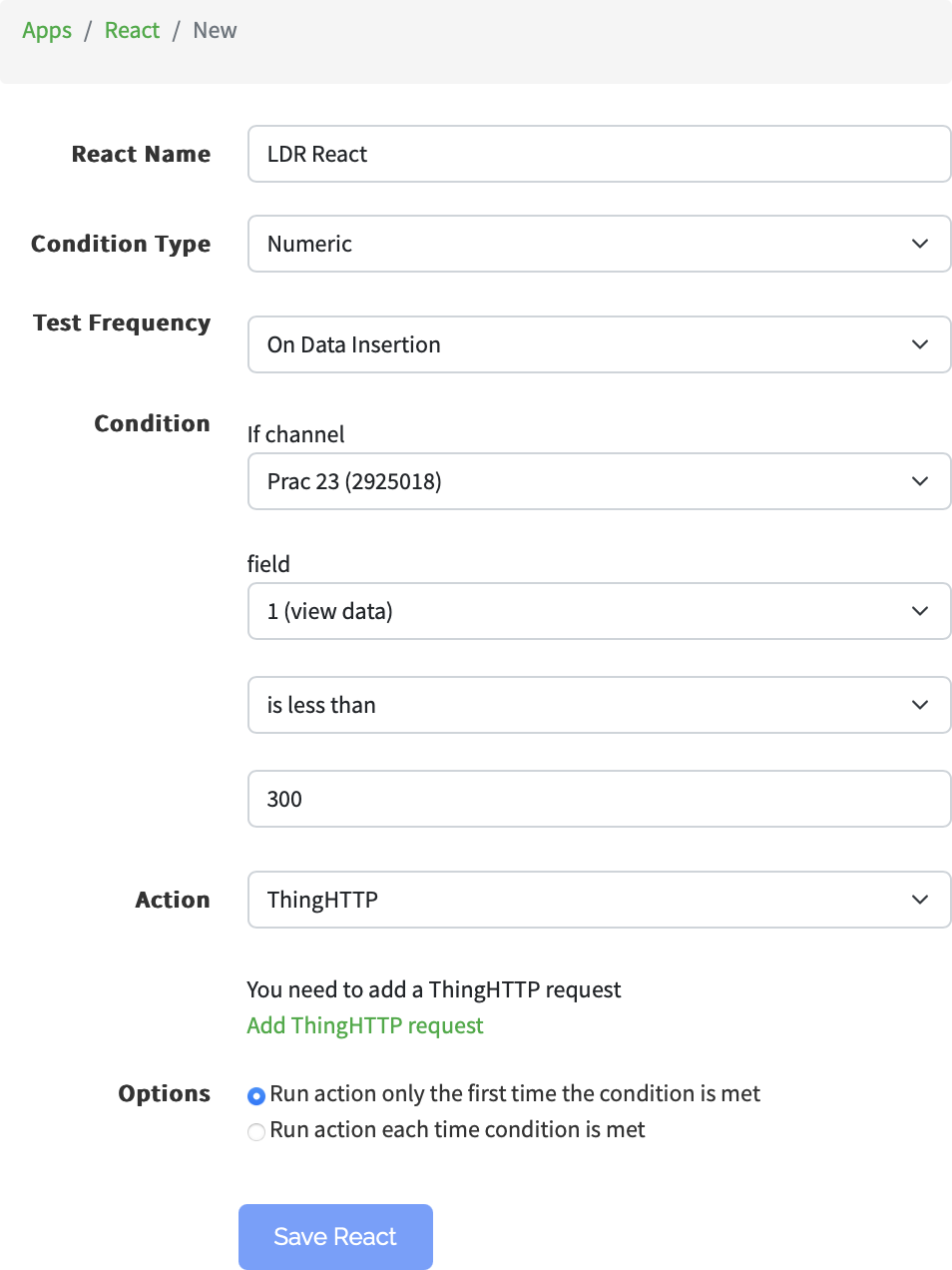
y(6000); }

}

## NOTE:

1. TEXT IN RED COLOUR IS YOUR WRITE API KEY
2. Make sure you see your Serial Monitor and check if Data is sent to Thingspeak.
3. To check result about data upload, go to thingspeak, click on private view and see the graph.
4. Once you finish doing the above steps go back to Thingspeak and next to the CHANNELS tab

,click on the APPS tab and select React option.

1. Click on the NEW REACT button (Green colour) and give a React name. Here it is LDR React and do following settings:
2. In the Action tab in the above figure, select ThingTweet option and then it will ask to link with your twitter account
3. Link and Choose your Twitter Account.
4. Save the React.
5. Make sure your intensity of LDR is less than 300 in the TinkerCad LDR slider and then check your Twitter account for the Tweet of “LIGHT ON” message.

# Practical No. 25

**Aim:** To interface servo motor / DC motor with Arduino and WAP to sweep a servo back and forth through its full range of motion/to control a DC motor.

**Components:** Arduino UNO R3,Breadboard,4 Resistor,3 LED, 1 Push Button, Wires

## Circuit:

**Code:**

const int poten = A3; int var;

void setup()

{

Serial.begin(9600); pinMode(6, OUTPUT);

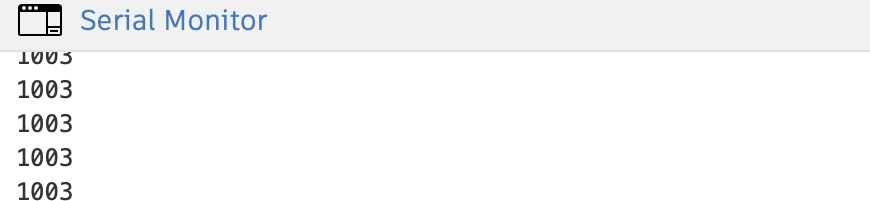
}

void loop()

{

var = analogRead(poten); analogWrite(6, var); Serial.println(var);

}

**Output:**