Practical No: 1

Introduction to Arduino

Arduino is an open-source hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. Its hardware products are licensed under a CC BY-SA license, while software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL),[1] permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially from the official website or through authorized distributors.

Types of Arduino-

1)Arduino uno

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

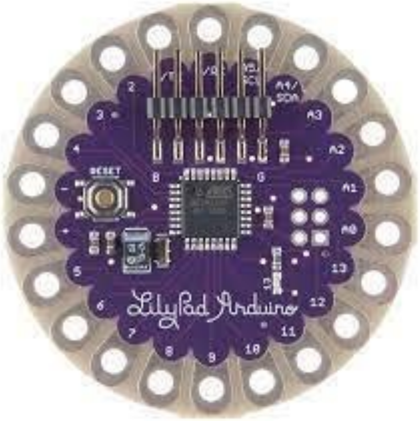
2) Arduino mega

This boards is considered as the microcontroller that uses the Atmega2560 in it. There are total 54 input pins and output pins in it in which 14 pins are of PWM output, 4 pins are of hardware port, 16 pins as analog inputs. The board also contain one USB connection, ICSP header, power jack and one REST pin.There are additional pins that act as crystal oscillator having frequency of 16 MHz. The board also has flash memory of 256KB size that uses to store the data in it. The Arduino Mega board can be attached to computer system via USB connection and power supply can be provided to board by using battery or AC to DC adapter. As the board has large number of pins fitted in it that make the board suitable for projects that requires more number of pins in it.

3) Arduino nano: The Arduino Nano is equipped with 30 male I/O headers, in a DIP-30-like configuration, which can be programmed using the Arduino Software integrated development environment (IDE), which is common to all Arduino boards and running both online and offline. The board can be powered through a type-B mini-USB cable or from a 9 V battery.[2]

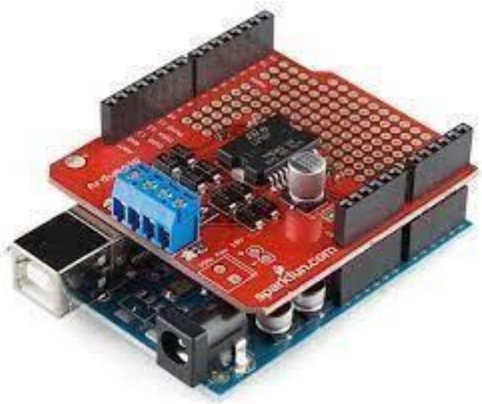
4) Lilypad Arduino:

The Lilypad Arduino is considered as other Arduino board type that is designed for integrating with wearable projects and e-textile projects. This board comes in round shape that helps to decrease the snagging and can be easily connected to other devices. This board uses the Atmega328 microcontroller and Arduino bootloader in it. This board uses very less external component in it that makes the design easy and compatible. The board requires 2 volt to 5 volt power supply and use large size pin holes so that it can be easily connect to other devices. This board is widely used for controlling different device that includes motor, light and switch.



5) Arduino Shields

The Arduino shields are considered as pre-build circuit boards that are used to connect other Arduino boards. The Arduino shield are placed on top of Arduino boards and enhance the capability of board to get connected to internet network, controlling of motor, controlling of LCD and also help to establish wireless communication. There are different type of shields available for the use. It includes Wireless Shields, Ethernet Shield, Proto Shield and GSM shield. This helps to increase the compatibility of the Arduino boards.



6) Arduino Red Board

The Arduino Red board is another type of Arduino board that uses the mini USB cable for getting programmed and the Arduino IDE is used for this purpose. This board is compatible with Windows 8 operating system and there is no need to change the security settings to make this board working. The Red board uses the FTDI chip and USB chip for the connection to other device. As the design of red board is very simple it can be easily integrate with other projects. The only requirement if to plug the red board and select appropriate option and can upload program in no time. The barrel jack can be used to control the USB cable of the Arduino Red board.

Wires and types

Wire:

Wire is a long, thin and flexible piece of metal. Wires are made in many different metals and sizes, and are used for many purposes. Wire rope has been used for centuries. Electrical wire is wire used to carry electricity.

Normal wire:

The default wires in Tinkercad circuits offer a tidy and colorful way to represent your circuit design. In most classrooms, however, circuit prototyping is done with reusable hookup/jumper wires or alligator clips, and we see no reason why your Tinkercad circuits shouldn’t use them too.To use this new option, simply create or select a wire in your design and use the properties window to change its type.

Hook up wire:

Hook up wires, also known as lead wires, are single core insulated cables used in the internal wiring of electronic and electrical equipment. Specific applications include motors, transformers, switchboard, panels, rectifier and electronic circuits.

Alligator wire:

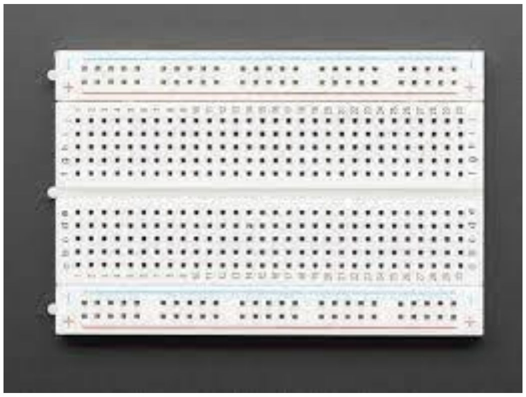
An alligator clip is a small, spring-loaded metal clip that can be used to make temporary connections between two wires or between a wire and the anode or cathode of a device. The clip has one end where a wire is screwed into place whilst the other end can be clipped or unclipped as needed

Automatic wire:

In the case of an automatic wire stripper, one side is held tight and, simultaneously, the other side is cut and removed. An automatic wire stripper can help even a novice cut and strip most wires quickly. However, it only works for certain size ranges of wires.

Breadboard

A breadboard is a solderless device for temporary prototype with electronics and test circuit designs. Most electronic components in electronic circuits can be interconnected by inserting their leads or terminals into the holes and then making connections through wires where appropriate. The breadboard has strips of metal underneath the board and connect the holes on the top of the board. The metal strips are laid out as shown below. Note that the top and bottom rows of holes are connected horizontally and split in the middle while the remaining holes are connected vertically.



Resistor

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to thevoltage across it. This is the well-known Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Practical No: 2

Aim: WAP to blink Arduino onboard LED & to interface external LED with Arduino and write a program to turn ON LED for 1 second after every 2 seconds.

Objectives:

- To learn Arduino UNO basics.
- Write a program to blink Arduino onboard LED and to interface external LED with Arduino.
- Write a program to turn ON LED for 1 second after every 2 seconds

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to thevoltage across it. This is the well-known Ohm's Law. In alternating-current

(AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

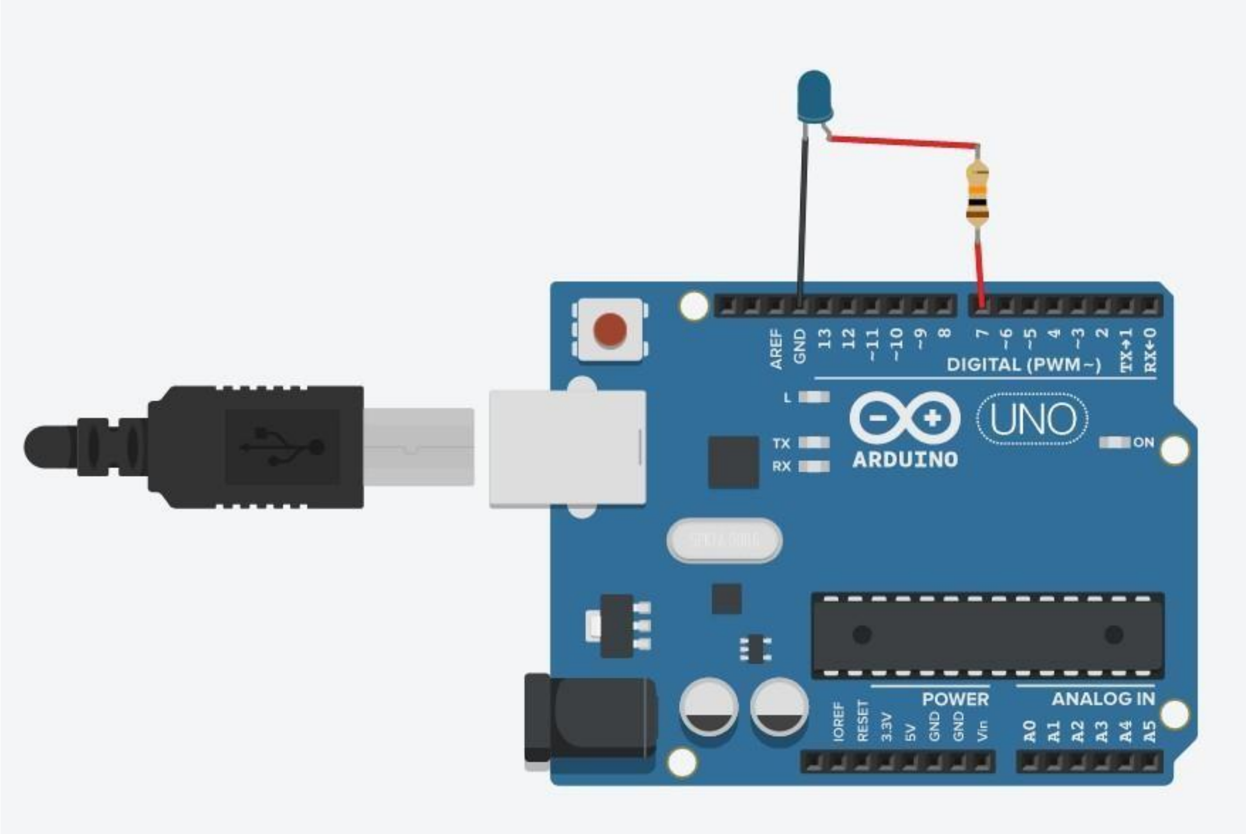
loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull- up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwriter() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW

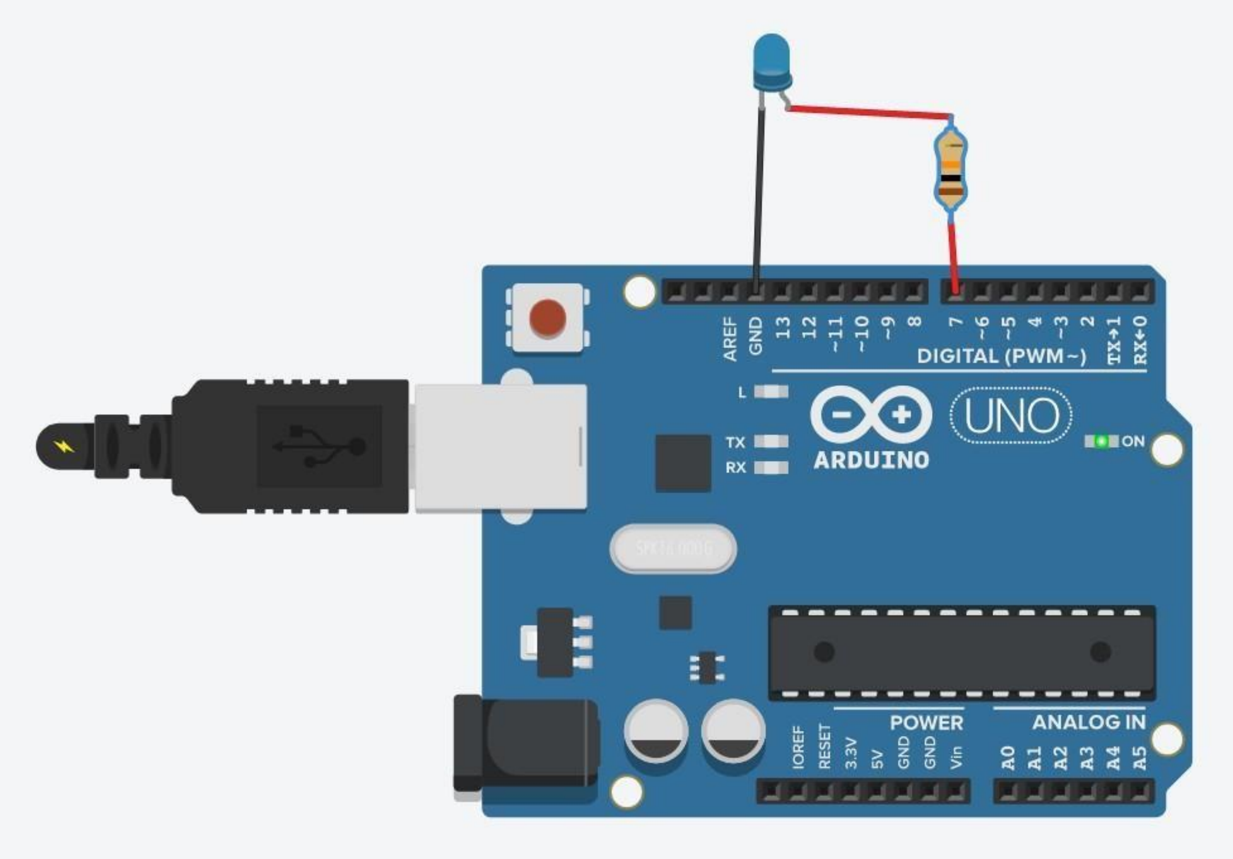
Circuit Diagram:



Code:

```
void setup()
{
  pinMode(7, OUTPUT);
}

void loop()
{
  digitalWrite(7, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(7, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Output:



Conclusion:

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), LED's and interfacing external LED and resistor with Arduino.

Practical No:3

Aim: WAP to blink Arduino onboard LED & to interface external LED with Breadboard & Arduino and write a program to turn ON LED for 1 second after every 2 seconds

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 turn ON LED for 1 second after every 2 seconds

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to thevoltage across it. This is the well-known Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

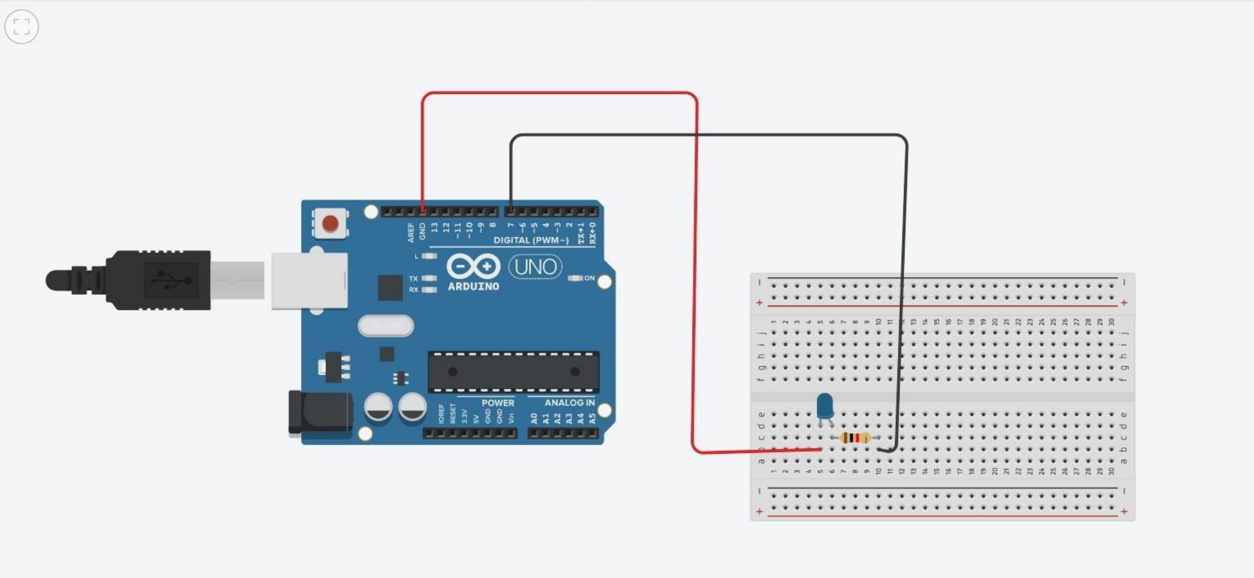
loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull- up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwriter() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

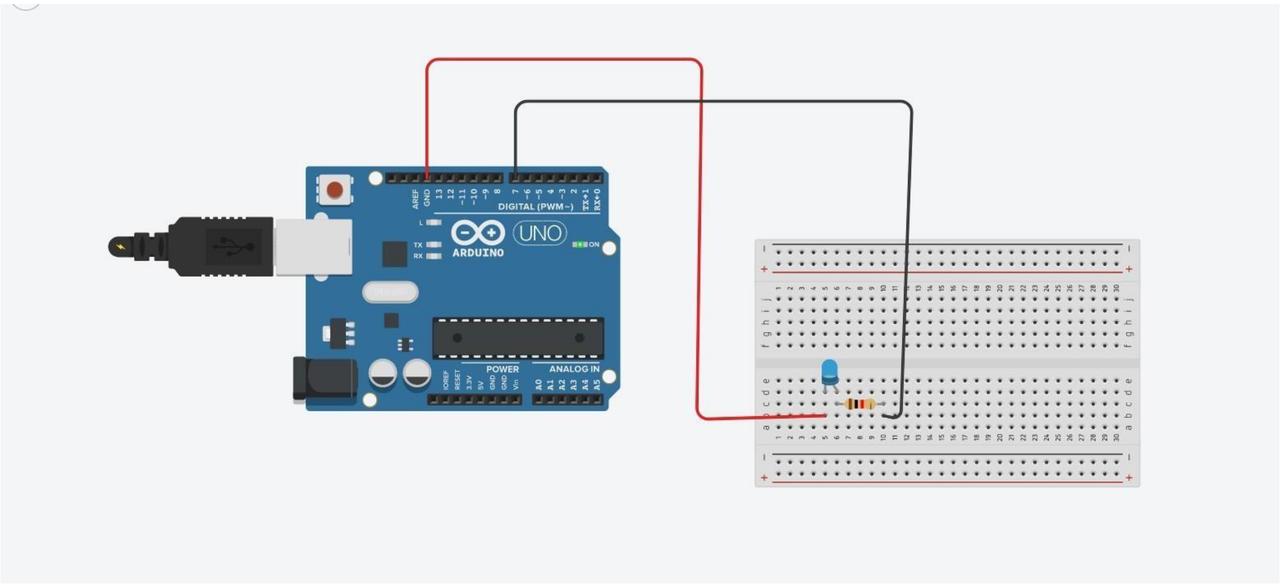
Circuit Diagram:



Code:

```
// C++ code
int Blink_LED=7;
void setup()
{
  pinMode(Blink_LED, OUTPUT);
}

void loop()
{
  digitalWrite(Blink_LED, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(Blink_LED, LOW);
  delay(2000); // Wait for 2000 millisecond(s)
}
```

Output:



Conclusion:

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED’s and interfacing external LED and resistor with breadboard & Arduino.

Practical No: 4

Aim: WAP to blink LEDs with Arduino & to interface external 5 LEDs with Breadboard & Arduino and write a program to turn ON/OFF LED.

Objective:

- 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 of blinking 5LEDs – test program to run 5 LEDs in a pattern

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to thevoltage across it. This is the well-known Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

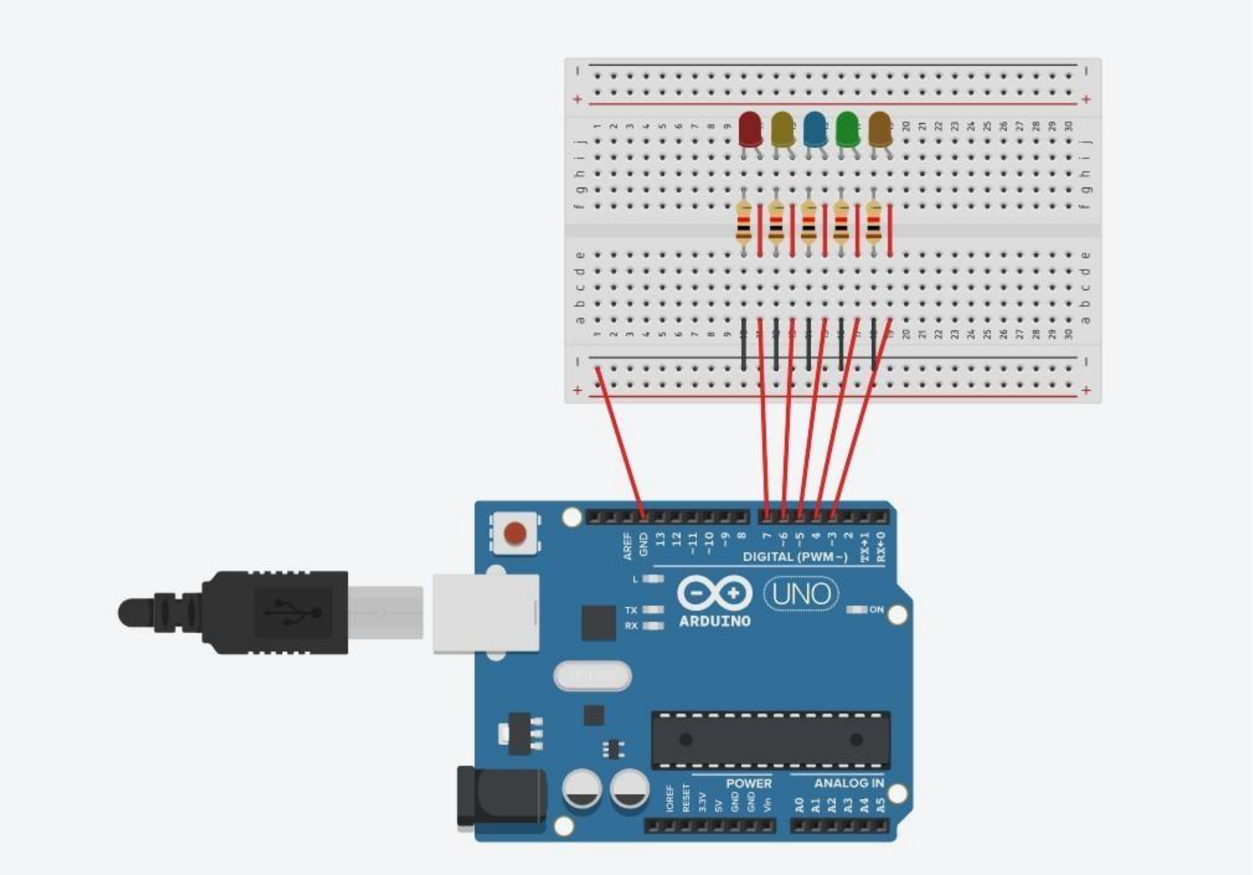
loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull- up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwriter() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

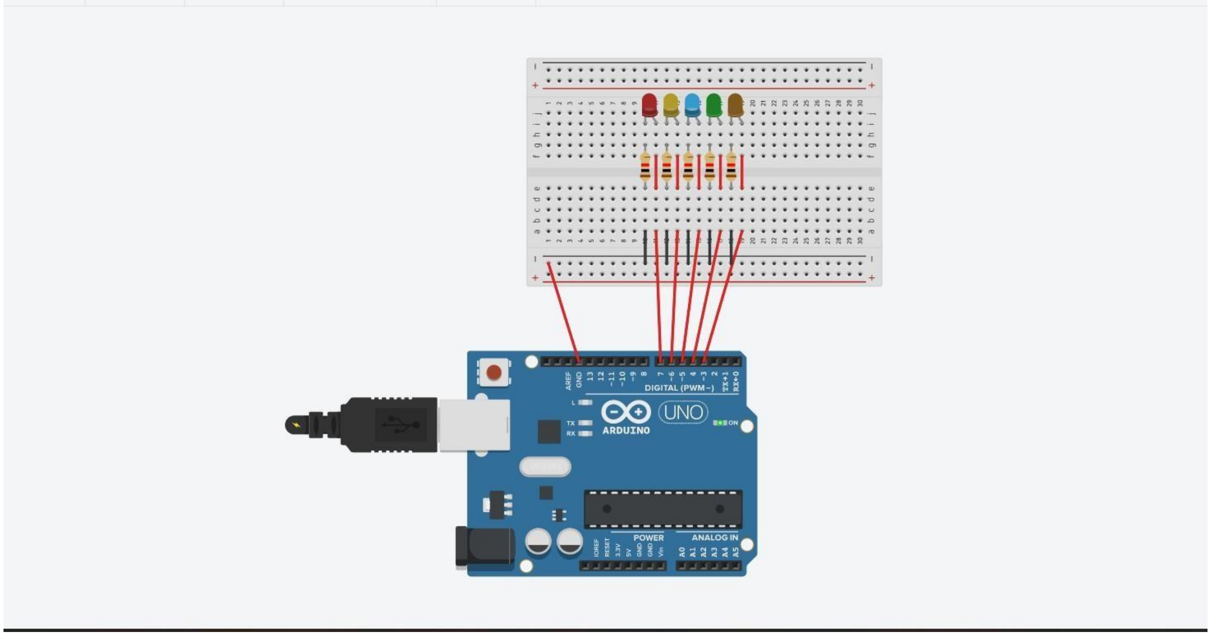
Circuit Diagram:



Code:

```
// C++ code
//
int   led1=7;
int   led2=6;
int   led3=5;
int   led4=4;
int   led5=3;
void setup()
{
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);

}

void loop()
{
  digitalWrite(led1, HIGH);
  delay(80);
  digitalWrite(led1, LOW);
  digitalWrite(led2, HIGH);
  delay(80);
  digitalWrite(led2, LOW);
  digitalWrite(led3, HIGH);
  delay(80);
  digitalWrite(led3, LOW);
  digitalWrite(led4, HIGH);
  delay(80);
  digitalWrite(led4, LOW);
  digitalWrite(led5, HIGH);
  delay(80);
  digitalWrite(led5, LOW);
  delay(500);
}
```

Output:



Conclusion:

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED's and interfacing external 5 LEDs blinking in a pattern of blinks with connections of resistors, breadboard, Jumper wires & Arduino.

Practical No: 5

Aim: WAP to interface external 6 LEDs with Breadboard & Arduino and write a program to blink 6 LEDs, one at a time, in a back and forth formation.

Objective:

- 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 blink 6 LEDs, one at a time. In a back and forth formation.

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to thevoltage across it. This is the well-known Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

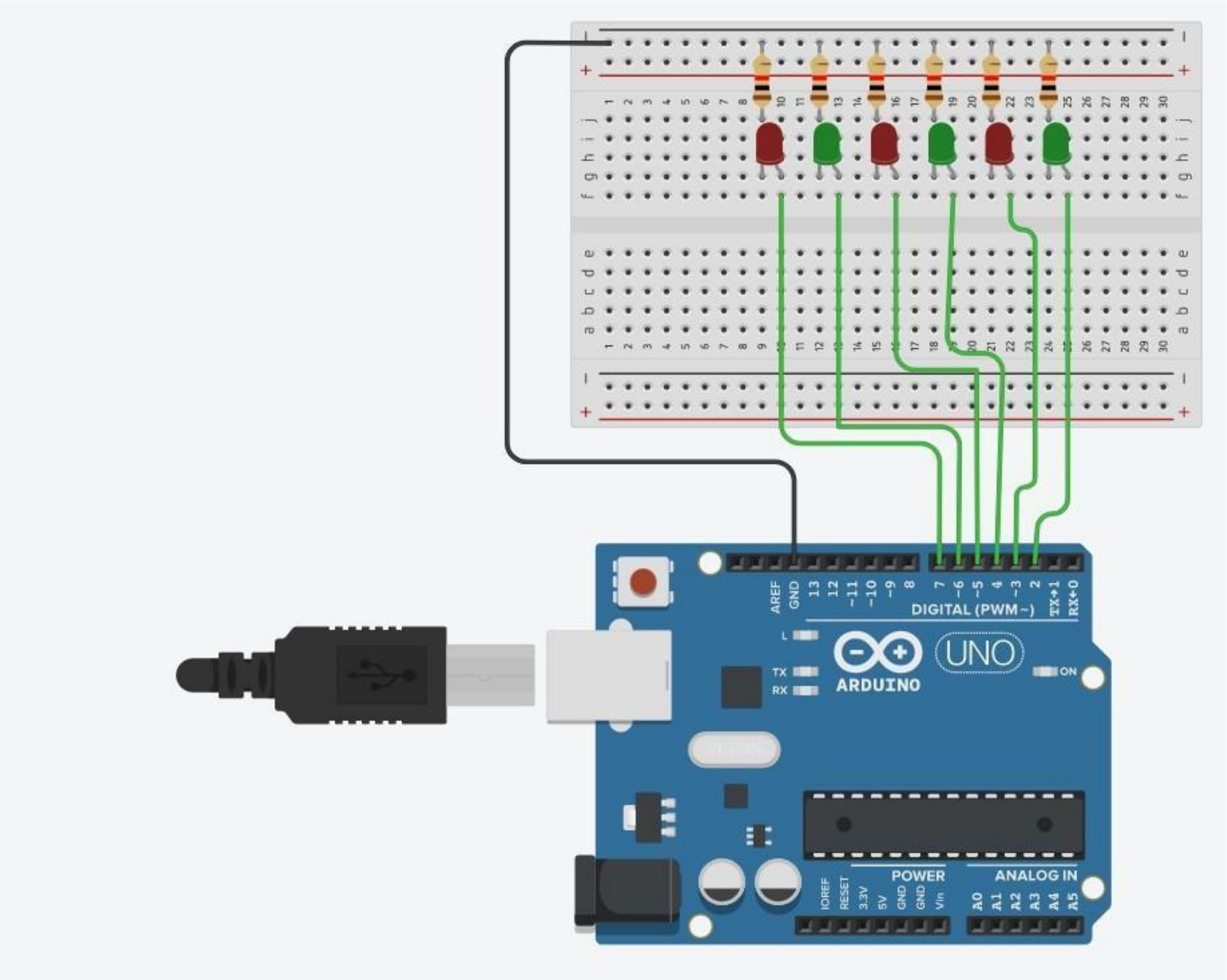
loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull- up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwriter() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

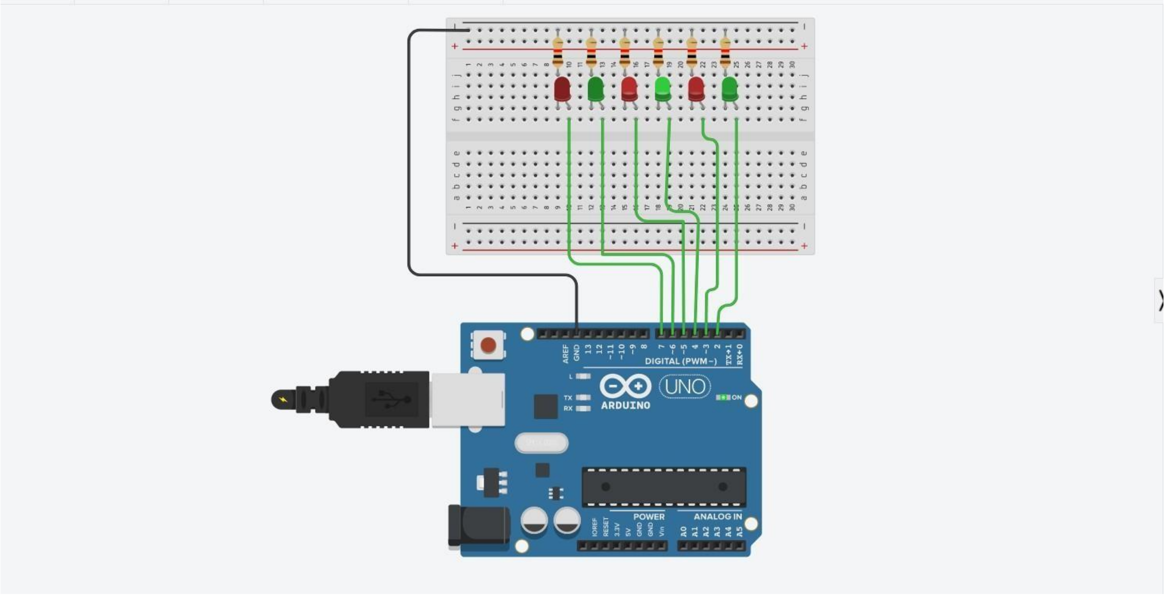
Circuit Diagram:



Code:

```
// C++ code
//
int timer=100;
void setup()
{
  for(int pin=2; pin < 8;pin++)
  {
        pinMode(pin, OUTPUT);
  }
}

void loop()
{
  for(int pin=2; pin < 8;pin++)
  {
    digitalWrite(pin, HIGH);
    delay(timer);
    digitalWrite(pin, LOW);
  }
  for(int pin=7; pin >= 2;pin--)
  {
    digitalWrite(pin, HIGH);
    delay(timer);
    digitalWrite(pin, LOW);
  }
}
```

Output:



Conclusion:

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED’s and interfacing external 6 LEDs blinking one at a time, in a back and forth formation with connections of resistors, breadboard, jumper wires and Arduino.

Practical No: 6

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

Objective:

- 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

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In

alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Jumper Wires: A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

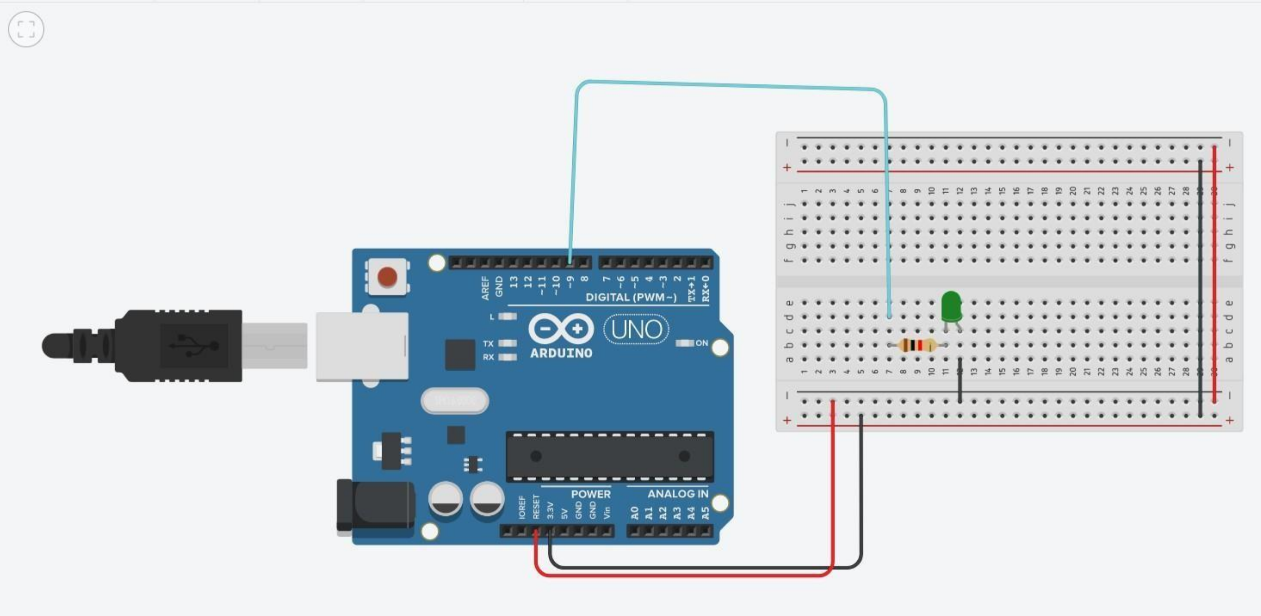
pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

analogWrite(): Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to analogWrite(), the pin will generate a steady rectangular wave of the specified duty cycle until the next call to analogWrite() (or a call to digitalRead() or digitalWrite()) on the same pin.

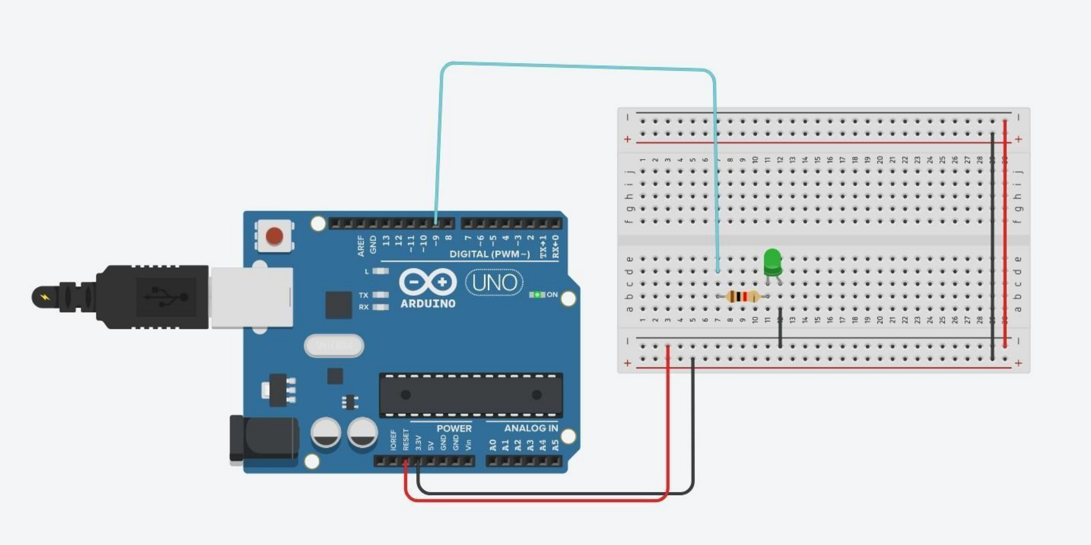
Circuit Diagram:



Code:

```
// C++ code
//
int brightness=0;
int led=9;
void setup()
{
  pinMode(led, OUTPUT);
}
void loop()
{
  for(brightness=0;brightness<=1000;brightness+=5)
  {
    analogWrite(led,brightness);
    delay(50);
  }
  for(brightness=1000;brightness>=0;brightness -=5)
  {
    analogWrite(led,brightness);
    delay(50);
  }}
```

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: 7

Aim: WAP to interface Push Button with Arduino and write a program to turn ON LED when push button is pressed.

Objective:

- 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 LED when push button is pressed

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

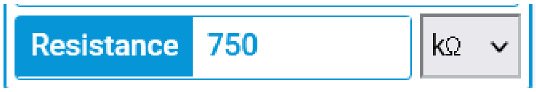
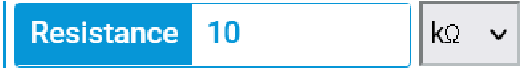
Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In

alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

[Note: Resistor with push button will be  Resistance 750 kΩ and resistor with LED will be  Resistance 10 kΩ]

Jumper Wires: A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

Push Button/ Switches: A Push Button switch is a type of switch which consists of a simple electric mechanism or air switch mechanism to turn something on or off. Depending on model they could operate with momentary or latching action function. The button itself is usually constructed of a strong durable material such as metal or plastic.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

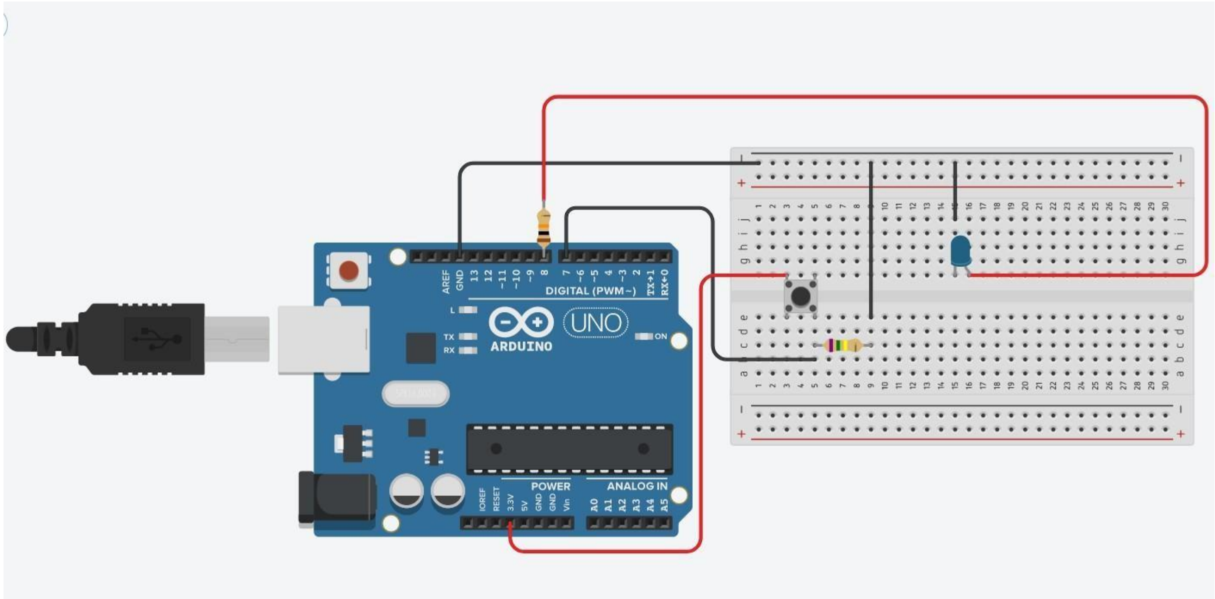
pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).

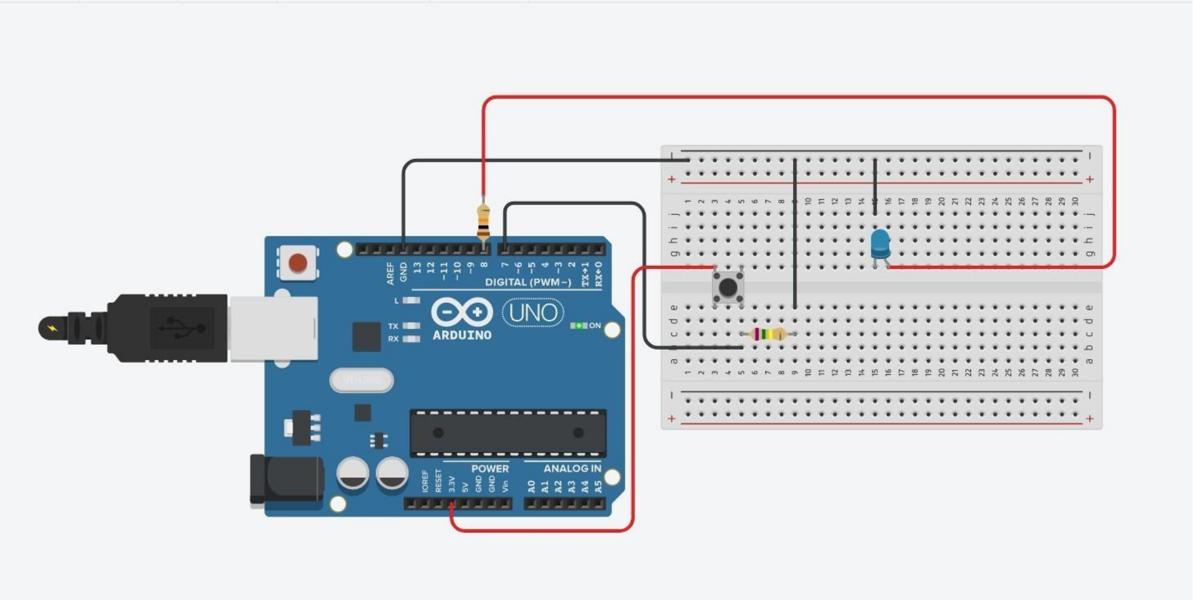
Circuit Diagram:



Code:

```
// C++ code
//
unsigned const LED=8;
unsigned const BUTTON=7;
unsigned int buttonState=0;
void setup()
{
  pinMode(BUTTON, INPUT);
  pinMode(LED, OUTPUT);
}

void loop()
{
  if(digitalRead(BUTTON)==1)
     {
      digitalWrite(LED,HIGH);
      buttonState+=1;
      if(buttonState%2==0)
      {
        digitalWrite(LED,LOW);
        buttonState=0;
      }
      delay(100);
    }
}
```

Output:



Conclusion: Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED’s and interfacing PUSH Button with Arduino to turn ON LED when push button is pressed with connections of resistors, breadboard, Jumper wires and Arduino

Practical No: 8

Aim: WAP to interface Push Button with Arduino and write a program to turn ON LEDs when push button is pressed and gets OFF after second automatically like a door bell.

Objective:

- 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 after second automatically.

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

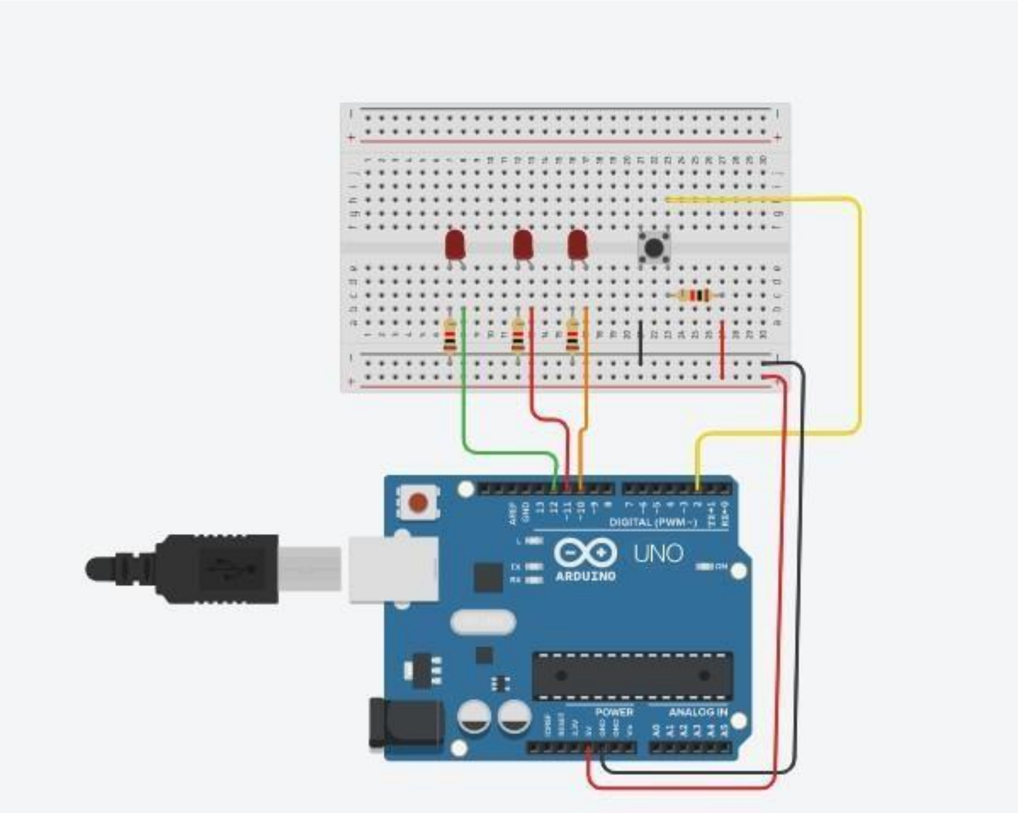
A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it.

This is the wellknown Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.
Jumper Wires: A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

Push Button/ Switches: A Push Button switch is a type of switch which consists of a simple electric mechanism or air switch mechanism to turn something on or off. Depending on model they could operate with momentary or latching action function. The button itself is usually constructed of a strong durable material such as metal or plastic.

Functions:
setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board. loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board. **pinmode():**The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.
delay(): The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function. digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.
digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).

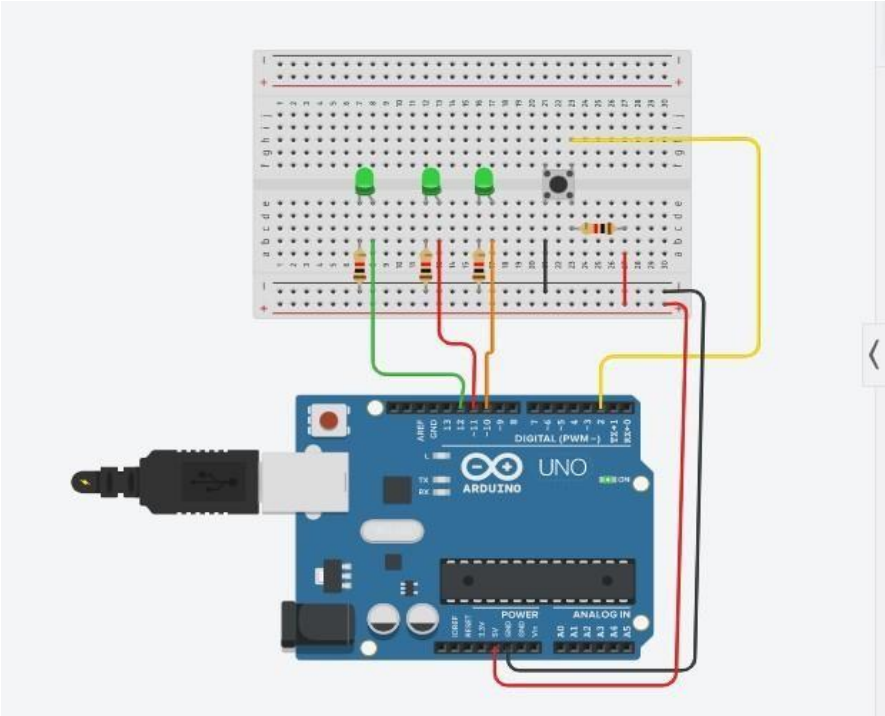
Circuit Diagram:



Code:

```
// C++ code
//
int ledPin1 = 10;
int ledPin2 = 11;
int ledPin3 = 12;
int keyPin = 2;
void setup() {
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
pinMode(keyPin, INPUT);
}
void loop()
{
int keyState = digitalRead
(keyPin);
if(keyState == 0) {
digitalWrite(ledPin1, HIGH);
delay(500);
digitalWrite(ledPin2, HIGH);
delay(500);
digitalWrite(ledPin3, HIGH);
delay(500);
}
 else {
digitalWrite(ledPin1, LOW);
delay(500);
digitalWrite(ledPin2, LOW);
delay(500);
digitalWrite(ledPin3, LOW);
delay(500);
}
}
```

Output:



Practical No: 9

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

Objective:

- 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

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In

alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Jumper Wires: A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

Push Button/ Switches: A Push Button switch is a type of switch which consists of a simple electric mechanism or air switch mechanism to turn something on or off. Depending on model they could operate with momentary or latching action function. The button itself is usually constructed of a strong durable material such as metal or plastic.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

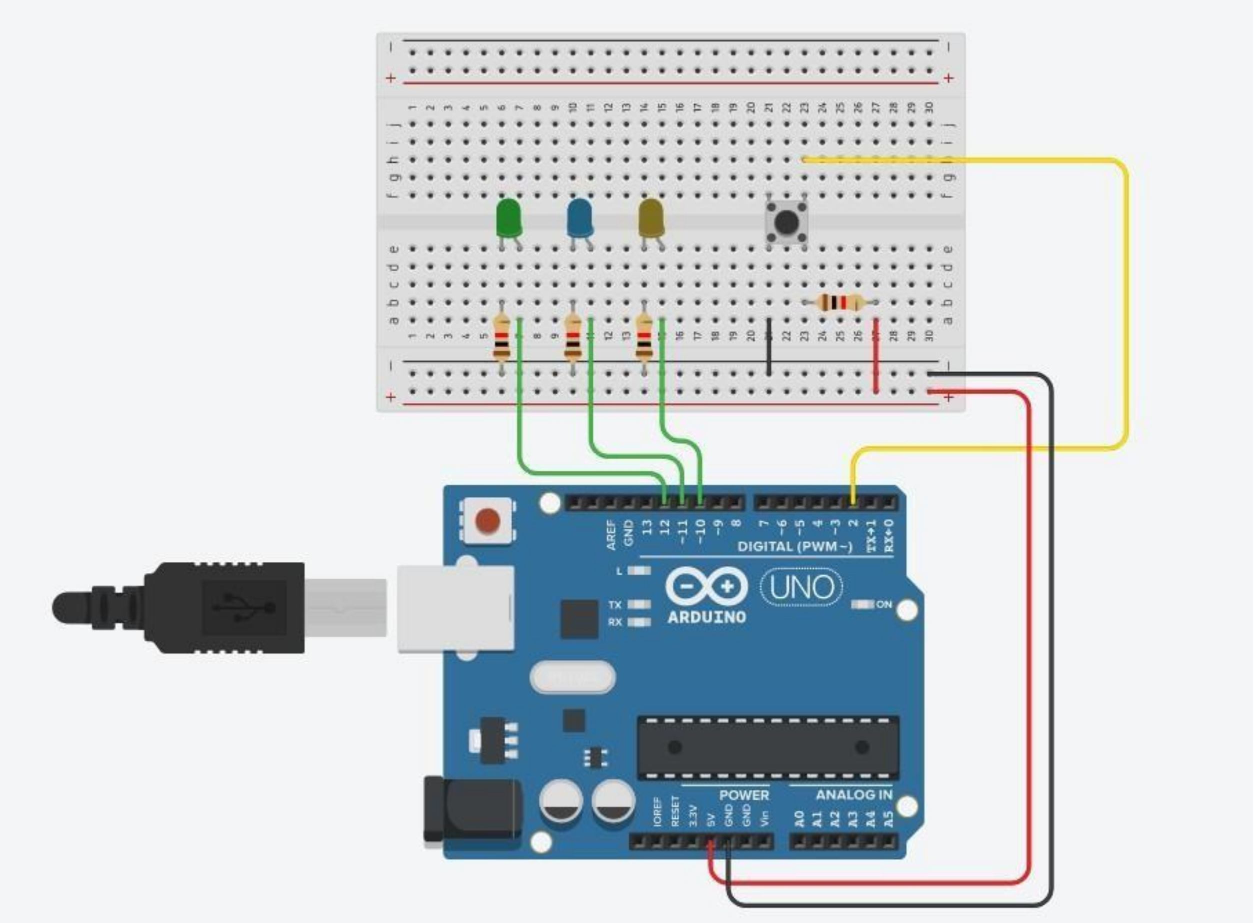
pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).

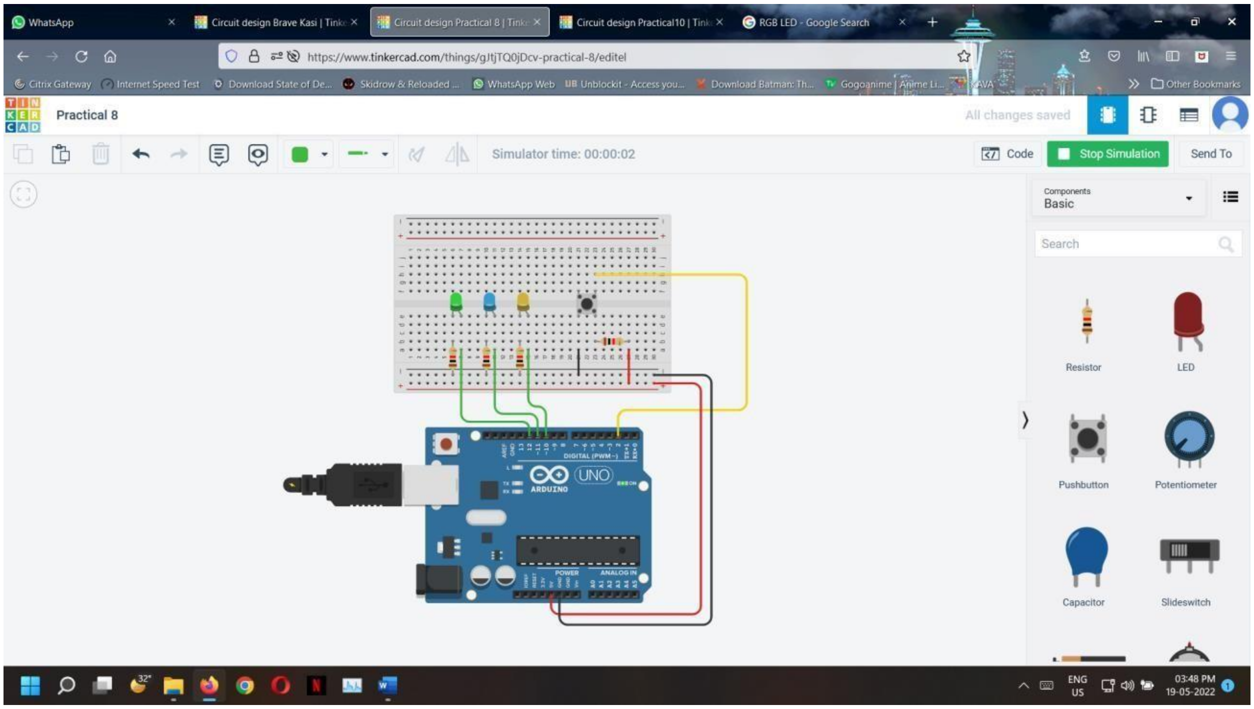
Circuit Diagram:



Code:

```
// C++ code
//

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

boolean ledOn=false;

void setup()
{
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(keyPin, INPUT);
}

void loop()
{
  int keyState=digitalRead(keyPin);
  if(keyState==0)
  {
    ledOn= !ledOn;
    digitalWrite(ledPin1, ledOn);
          delay(500);
    digitalWrite(ledPin2, ledOn);
          delay(500);
    digitalWrite(ledPin3, ledOn);
          delay(500);
  }

}
```

Output:



Conclusion:

Thus, learnt about basic components of Iot like Arduino UNO(blink Arduino onboard LED), Breadboard LED's and interfacing Push Button with Arduino to turn ON LEDS when push button is pressed & gets OFF when push button is pressed again with connections of resistors, breadboard, Jumper wires& Arduino.

Practical No: 10

Aim: WAP to interface Push Button with Arduino and write a program to turn ON LEDs when push button is pressed in 7 different patterns.

Objective:

- 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 in 7 different patterns

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

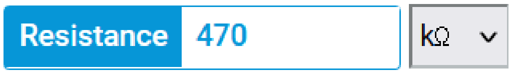
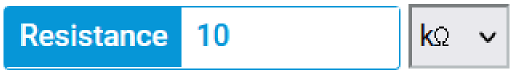
Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In

alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

[Note: resistor with LED bulbs are at  Resistance 470 kΩ and resistor with ground are  Resistance 10 kΩ]

Jumper Wires: A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

Push Button/ Switches: A Push Button switch is a type of switch which consists of a simple electric mechanism or air switch mechanism to turn something on or off. Depending on model they could operate with momentary or latching action function. The button itself is usually constructed of a strong durable material such as metal or plastic.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

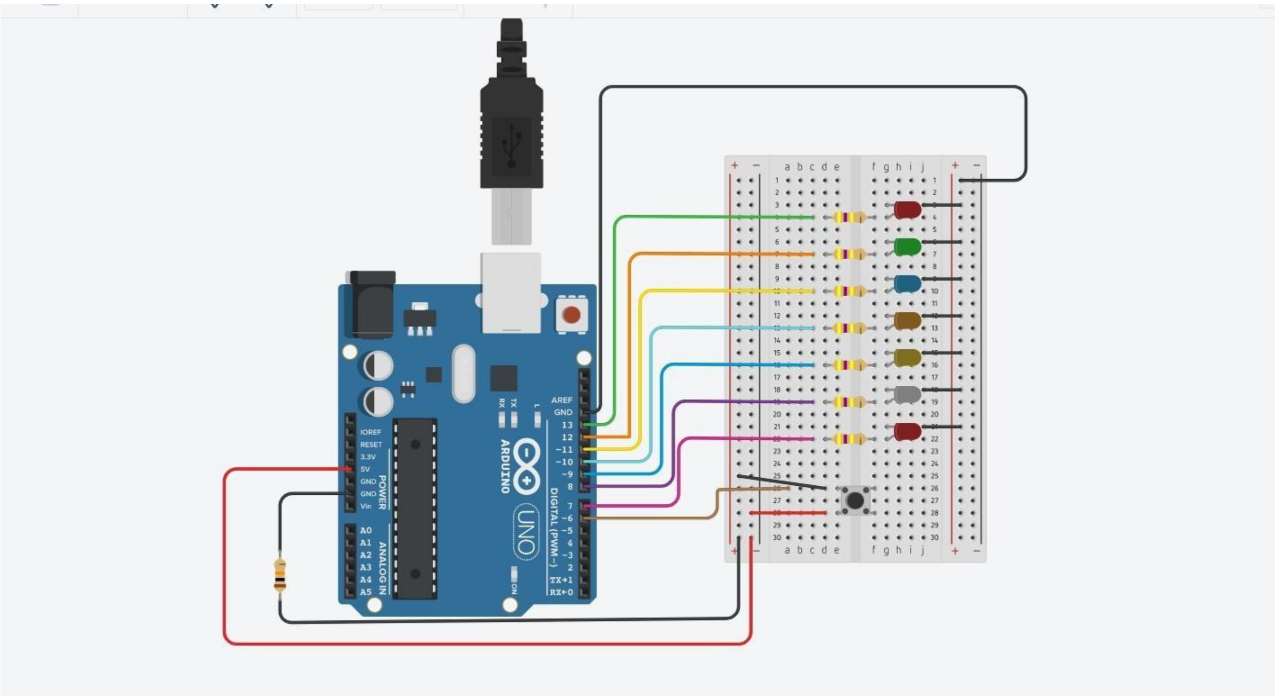
pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).

Circuit Diagram:



Code:

```
int L1=13;
int L2=12;
int L3=11;
int L4=10;
int L5=9;
int L6=8;
int L7=7;

int buttonpin=6;
int de=50;  //DELAYTIME
int p=0;
int buttonstate=0;

  void setup()
  {
    pinMode(L1, OUTPUT);
    pinMode(L2, OUTPUT);
    pinMode(L3, OUTPUT);
    pinMode(L4, OUTPUT);
    pinMode(L5, OUTPUT);
    pinMode(L6, OUTPUT);
    pinMode(L7, OUTPUT);
    pinMode(buttonpin, INPUT);
  }
  void loop()
  {

      buttonstate=digitalRead(buttonpin);
      if(buttonstate ==HIGH)
    {
      p++;
      delay(2000);
    }
    if(p==1)
    {
          digitalWrite(L1,1);
          digitalWrite(L2,0);
          digitalWrite(L3,0);
          digitalWrite(L4,0);
          digitalWrite(L5,0);
          digitalWrite(L6,0);
          digitalWrite(L7,0);           //1
          delay(de);

          digitalWrite(L1,0);
          digitalWrite(L2,1);
          digitalWrite(L3,0);
          digitalWrite(L4,0);
          digitalWrite(L5,0);
          digitalWrite(L6,0);
          digitalWrite(L7,0);  //2
          delay(de);

          digitalWrite(L1,0);
          digitalWrite(L2,0);
          digitalWrite(L3,1);
          digitalWrite(L4,0);
          digitalWrite(L5,0);
          digitalWrite(L6,0);
          digitalWrite(L7,0);
          delay(de);                    //3

          digitalWrite(L1,0);
          digitalWrite(L2,0);
          digitalWrite(L3,0);
          digitalWrite(L4,1);
          digitalWrite(L5,0);
          digitalWrite(L6,0);
          digitalWrite(L7,0);
          delay(de);                              //4

          digitalWrite(L1,0);
          digitalWrite(L2,0);
          digitalWrite(L3,0);
          digitalWrite(L4,0);
          digitalWrite(L5,1);
          digitalWrite(L6,0);
          digitalWrite(L7,0);
          delay(de);                              //5
```

```
        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,1);
        digitalWrite(L7,0);
        delay(de);                  //6

        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,1);
        delay(de);                  //7
    }

    if (p==2)
    {

        digitalWrite(L1,1);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,0);         //1
        delay(de);

        digitalWrite(L1,0);
        digitalWrite(L2,1);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,0);  //2
        delay(de);

        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,1);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,0);
        delay(de);                    //3

        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
        digitalWrite(L4,1);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,0);
        delay(de);                              //4

        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,1);
        digitalWrite(L6,0);
        digitalWrite(L7,0);
        delay(de);                              //5

        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
```

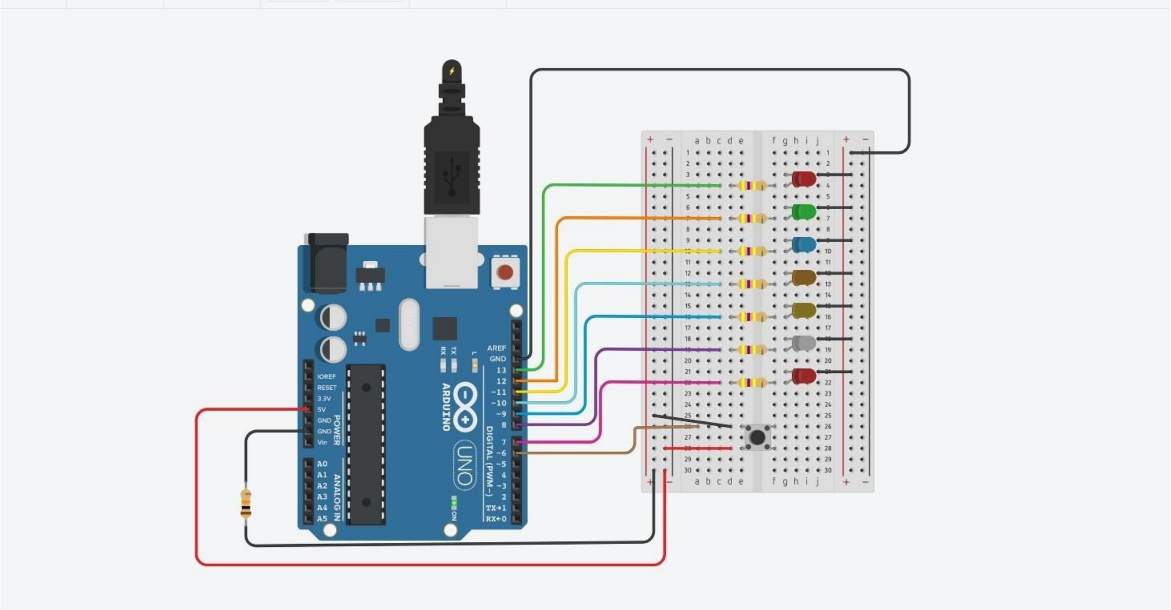
```
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,1);
        digitalWrite(L7,0);
        delay(de);                          //6

        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,1);
        delay(de);                          //7

    }

    if(p==3)
    {
        digitalWrite(L1,1);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,0);           //1
        delay(de);

        digitalWrite(L1,0);
        digitalWrite(L2,1);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,0);  //2
```

```
	delay(de);

	digitalWrite(L1,0);
	digitalWrite(L2,0);
	digitalWrite(L3,1);
	digitalWrite(L4,0);
	digitalWrite(L5,0);
	digitalWrite(L6,0);
	digitalWrite(L7,0);
	delay(de);                    //3

	digitalWrite(L1,0);
	digitalWrite(L2,0);
	digitalWrite(L3,0);
	digitalWrite(L4,1);
	digitalWrite(L5,0);
	digitalWrite(L6,0);
	digitalWrite(L7,0);
	delay(de);                         //4

	digitalWrite(L1,0);
	digitalWrite(L2,0);
	digitalWrite(L3,0);
	digitalWrite(L4,0);
	digitalWrite(L5,1);
	digitalWrite(L6,0);
	digitalWrite(L7,0);
	delay(de);                         //5

	digitalWrite(L1,0);
	digitalWrite(L2,0);
	digitalWrite(L3,0);
	digitalWrite(L4,0);
	digitalWrite(L5,0);
	digitalWrite(L6,1);
```

```
        digitalWrite(L7,0);
        delay(de);                  //6

    }
    if(p==4)
    {
        digitalWrite(L1,1);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,1);         //7,1
        delay(de);

        digitalWrite(L1,0);
        digitalWrite(L2,1);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,1);
        digitalWrite(L7,0);  //2,6
        delay(de);

        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,1);
        digitalWrite(L4,0);
        digitalWrite(L5,1);
        digitalWrite(L6,0);
        digitalWrite(L7,0);
        delay(de);                  //3,5

        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
        digitalWrite(L4,1);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,0);
        delay(de);                              //4
    }
    if (p==5)
    {

        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,0);
        digitalWrite(L4,1);
        digitalWrite(L5,0);
        digitalWrite(L6,0);
        digitalWrite(L7,0);

        digitalWrite(L1,0);
        digitalWrite(L2,0);
        digitalWrite(L3,1);
        digitalWrite(L4,0);
        digitalWrite(L5,1);
        digitalWrite(L6,0);
        digitalWrite(L7,0);
        delay(de);

        digitalWrite(L1,0);
        digitalWrite(L2,1);
        digitalWrite(L3,0);
        digitalWrite(L4,0);
        digitalWrite(L5,0);
        digitalWrite(L6,1);         //2,6
        digitalWrite(L7,0);
        delay(de);
```

```
          digitalWrite(L1,1);
          digitalWrite(L2,0);
          digitalWrite(L3,0);
          digitalWrite(L4,0);
          digitalWrite(L5,0); //1,7
          digitalWrite(L6,0);
          digitalWrite(L7,7);
          delay(de);
     }

     if(p==6)
     {
       digitalWrite(L1,1);
       delay(de);
       digitalWrite(L2,1);
       delay(de);
       digitalWrite(L3,1);
       delay(de);
       digitalWrite(L4,1);
       delay(de);
       digitalWrite(L5,1);
       delay(de);
       digitalWrite(L6,1);
       delay(de);
       digitalWrite(L7,1);
       delay(de);
       digitalWrite(L7,0);
       delay(de);
       digitalWrite(L6,0);
       delay(de);
       digitalWrite(L5,0);
       delay(de);

        digitalWrite(L4,0);
       delay(de);

        digitalWrite(L3,0);
       delay(de);
       digitalWrite(L2,0);
       delay(de);

        digitalWrite(L1,0);
       delay(de);
     }

     if(p==7)
     {
      digitalWrite(L1,0);
       digitalWrite(L2,0);
       digitalWrite(L3,0);
       digitalWrite(L4,0);
       digitalWrite(L5,0);
       digitalWrite(L6,0);
       digitalWrite(L7,0);

       p=0;

     }
    }
```

Output:



Conclusion:

Thus,learnt about basic components of loT like Arduino UNO(blink Arduino onboard LED),Breadboard LED's and interfacing Push Button with Arduino to turn ON7LEDS when push button is pressed in 7 different patterns with connections of resistors, breadboard, Jumper wires & Arduino.

Practical No: 11

Aim: WAP to interface RGB LED with Arduino and write a program to use the RGB LED with Arduino to obtain different colors.

Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To learn RGB LED
- Write a program to use the RGB LED with Arduino to obtain different colors

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Jumper Wires:

A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

RGB LED :

RGB LED means red, blue and green LEDs. RGB LED products combine these three colors to produce over 16 million hues of light. Note that not all colors are possible. Some colors are "outside" the triangle formed by the RGB LEDs. Also, pigment colors such as brown or pink are difficult, or impossible, to achieve.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

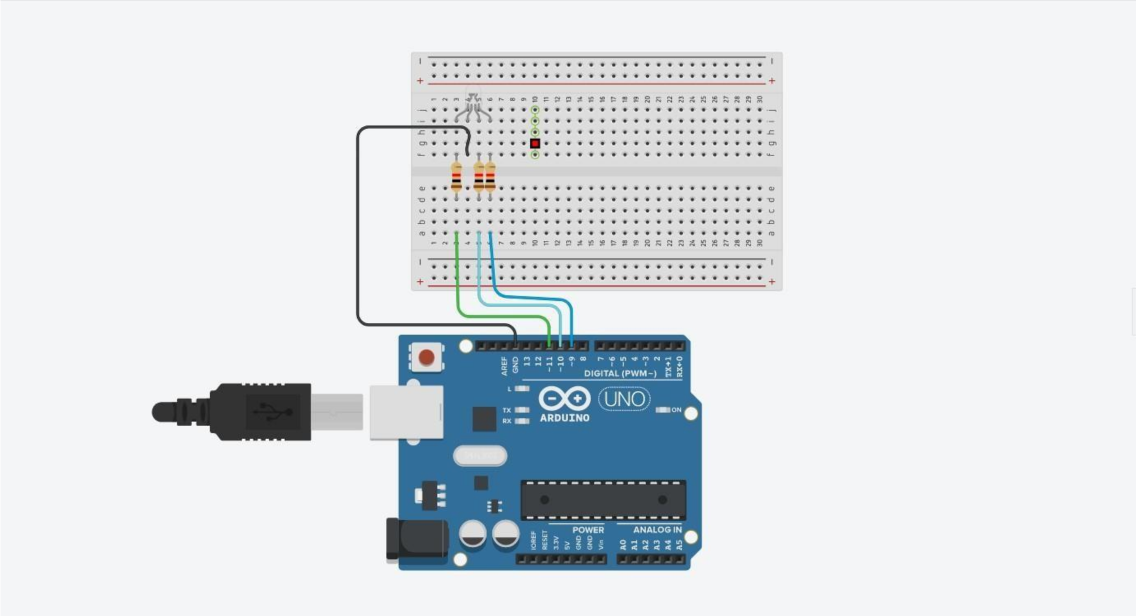
pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).

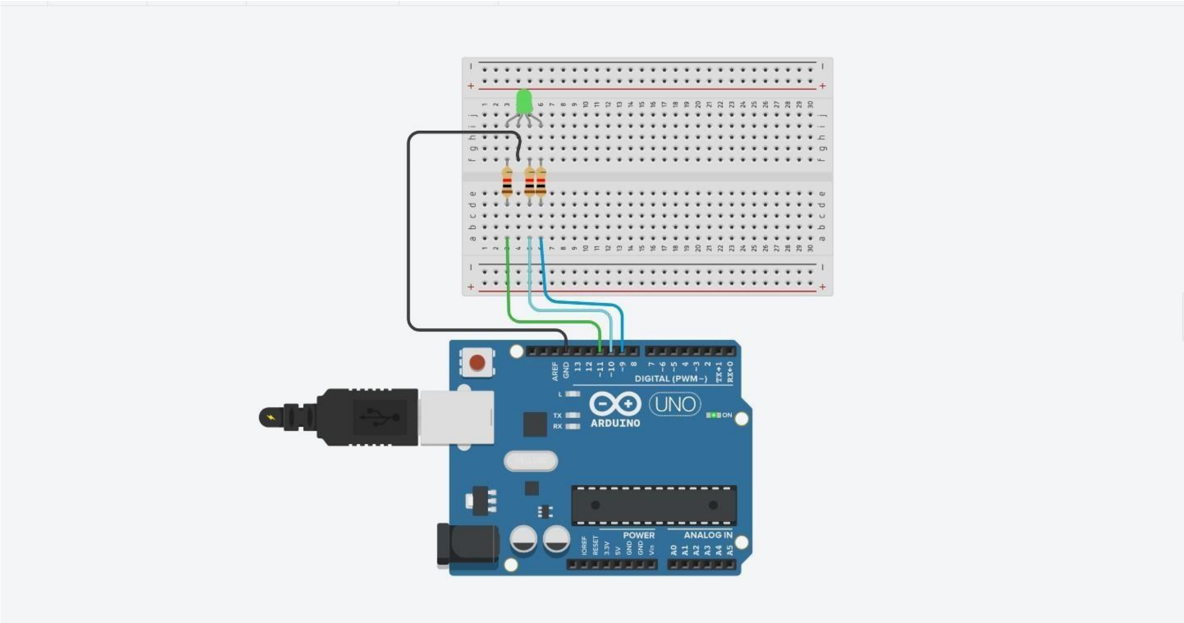
Circuit Diagram:



Code:

```
// C++ code
//

int red_light_pin=9;
int green_light_pin=10;
int blue_light_pin=11;
void setup()
{
  pinMode(red_light_pin, OUTPUT);
  pinMode(green_light_pin, OUTPUT);
  pinMode(blue_light_pin, OUTPUT);
}

void loop()
{
  digitalWrite(red_light_pin, HIGH);
  delay(1000);
  digitalWrite(red_light_pin, LOW);
  delay(1000);
  digitalWrite(green_light_pin, HIGH);
  delay(1000);
  digitalWrite(green_light_pin, LOW);
  delay(1000);
  digitalWrite(blue_light_pin, HIGH);
  delay(1000);
  digitalWrite(blue_light_pin, LOW);
  delay(1000);
}
```

Output:



Conclusion:

Thus, learnt about basic components of loT like Arduino UNO (blink Arduino onboard LED), Breadboard LED's and interfacing RGB LED with Arduino to turn
ON multiple colors as Red, Green & Blue with connections of resistors, breadboard, Jumper wires & Arduino

Practical No: 12

Aim: WAP to interface RGB LED with Arduino and write a program to use the RGB LED with Arduino to obtain different colors.

Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To learn RGB LED
- Write a program to use the RGB LED with Arduino to obtain different colors.

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Jumper Wires:

A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

RGB LED :

RGB LED means red, blue and green LEDs. RGB LED products combine these three colors to produce over 16 million hues of light. Note that not all colors are possible. Some colors are "outside" the triangle formed by the RGB LEDs. Also, pigment colors such as brown or pink are difficult, or impossible, to achieve.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

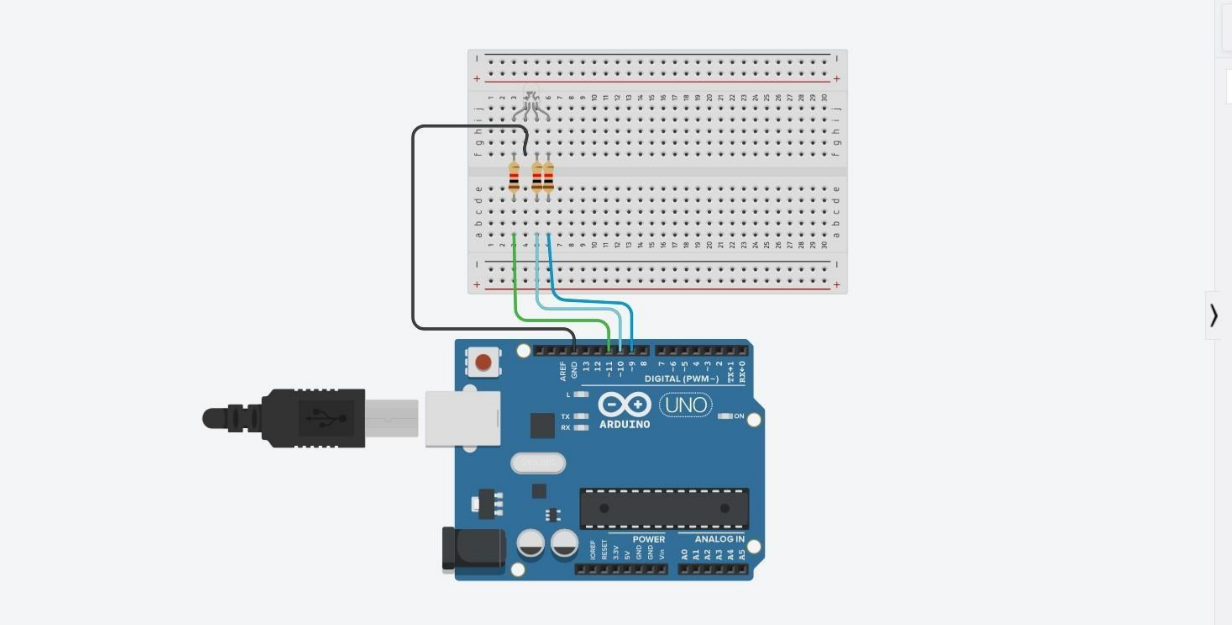
delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).

analogWrite(): Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to analogWrite(), the pin will generate a steady rectangular wave of the specified duty cycle until the next call to analogWrite() (or a call to digitalRead() or digitalWrite()) on the same pin.

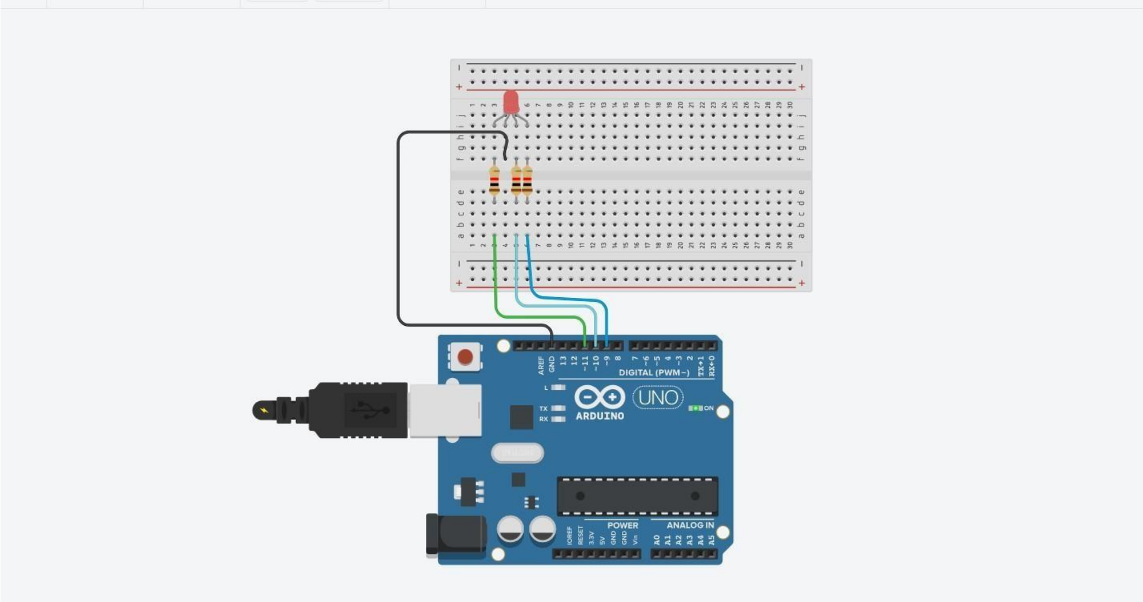
Circuit Diagram:



Code:

```
// C++ code
//

int red_light_pin=9;
int green_light_pin=10;
int blue_light_pin=11;
void setup()
{
  pinMode(red_light_pin, OUTPUT);
  pinMode(green_light_pin, OUTPUT);
  pinMode(blue_light_pin, OUTPUT);
}

void RGB_color(int red_light_value, int green_light_value, int blue_light_value)
{
    analogWrite(red_light_pin,red_light_value);
          analogWrite(green_light_pin,green_light_value);
          analogWrite(blue_light_pin,blue_light_value);
}

void loop()
{
  RGB_color(255,0,0);
  delay(1000);
  RGB_color(0,255,0);
  delay(1000);
  RGB_color(0,0,255);
  delay(1000);
  RGB_color(255,120,0);
  delay(1000);
 RGB_color(0,65,32);
 delay(1000);
 RGB_color(60,24,96);
 delay(1000);
 RGB_color(33,66,99);
 delay(1000);
 RGB_color(53,6,199);
 delay(1000);
}
```

Output:



Conclusion:

Thus, learnt about basic components of Iot like Arduino UNO(blink Arduino onboard LED).Breadboard LED's and interfacing RGB LED with Arduino to turn ON multiple8different colors with connections of resistors,breadboard,Jumper wires & Arduino.

Practical No. 13

Aim: WAP to interface ON/OFF LED with Arduino and write a program to use the Slide Button with Arduino to obtain LED ON & OFF.

Objectives:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To learn RGB LED

Write a program to use the Slide Button with Arduino to obtain LED ON & OFF.

Theory:

Breadboard, Resistor, Jumper wires, LED(generic), RGB LED and function like setup(), loop(), pinMode(), delay(), digitalWrite() etc.

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Jumper Wires:

A jumper wire is a conducting wire used to transfer electrical signals between two points in a circuit. The wires can either be used to modify circuits or to diagnose problems within a circuit.

Jumper wires typically vary in colour and size depending on what they are being used for. In breadboards, jump wires are used to establish connections between the central micro controller and other devices such as buttons and sensors.

If possible, the jumper wire should always be placed on the component side of a circuit board during assembly. The wires should also be routed in an X-Y manner, avoiding any bends. Jump wires should never be raised more than 1/8 of an inch above the surface of the circuit board.

RGB LED:

RGB LED means **red, blue and green LEDs**. RGB LED products combine these three colours to produce over 16 million hues of light. Note that not all colours are possible. Some colors are “outside” the triangle formed by the RGB LEDs. Also, pigment colours such as brown or pink are difficult, or impossible, to achieve.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

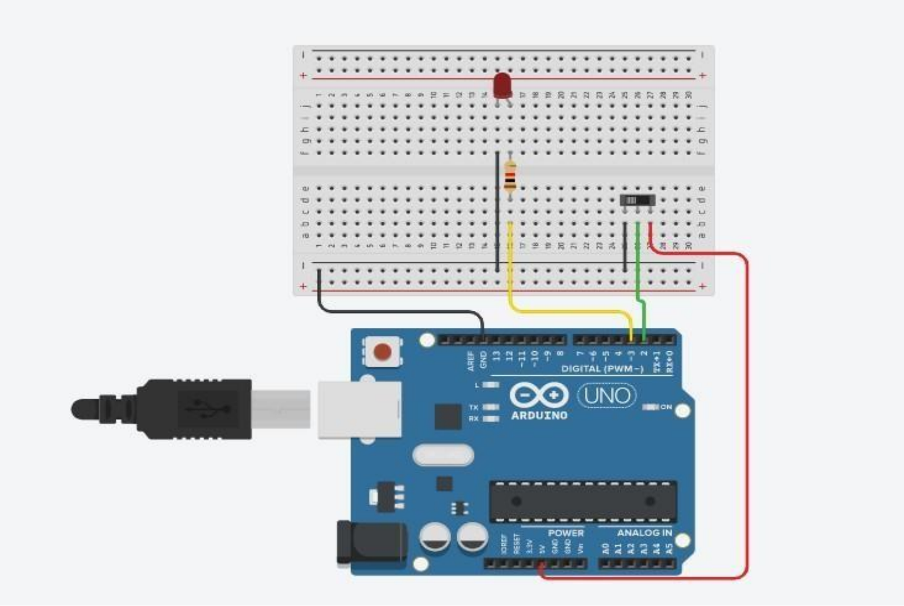
loop(): After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

pinMode(): The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay(): The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalWrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

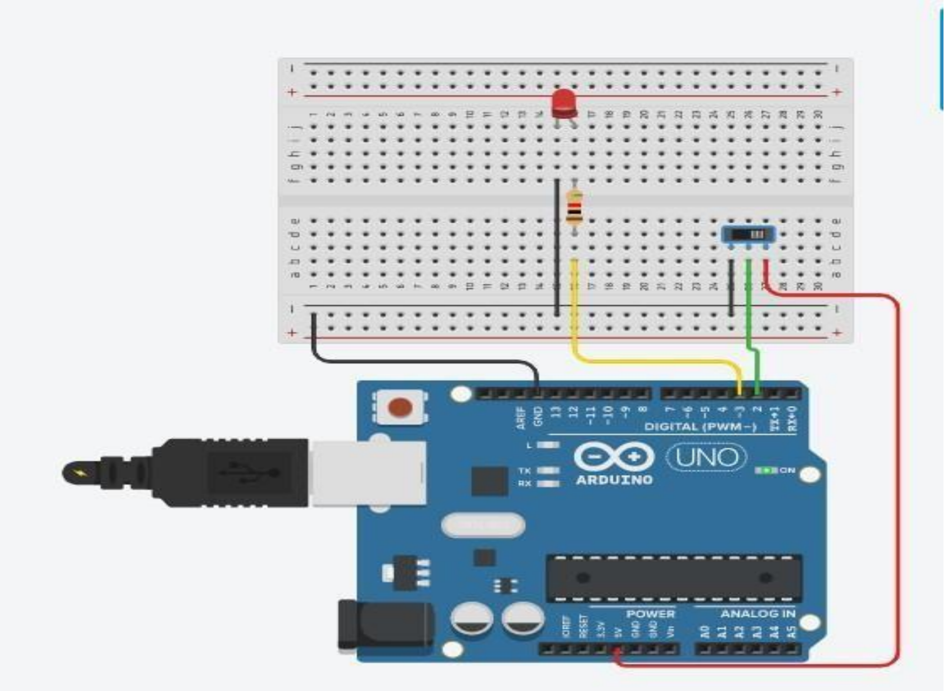
Circuit Diagram:



Code:

```
// C++ code
//
int BUTTON=2;
int LED=3;
int BUTTONstate=0;

void setup()
{
pinMode(BUTTON,INPUT);
pinMode(LED,OUTPUT);
}

void loop()
{
BUTTONstate=digitalRead(BUTTON);
if(BUTTONstate==HIGH)
{
digitalWrite(LED,HIGH);
}
else
{
digitalWrite(LED,LOW);
}
}
```

Output:



Conclusion:

Thus, learnt about basic components of IOT like Arduino UNO(blink Arduino on-board LED), Breadboard, LED’s and interfacing LED with Arduino to turn ON with slide button ON and OFF with connections of resistors, breadboard, Jumper wires & Arduino.

`Practical No: 14

Aim: WAP to interface Piezo Speaker with Arduino and write a program to use the Piezo Speaker with Arduino to obtain continuous buzzer sound.

Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To Learn Piezo Speaker
- Write a program to use the Piezo Speaker with Arduino to obtain continuous buzzer sound

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Jumper Wires:
A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

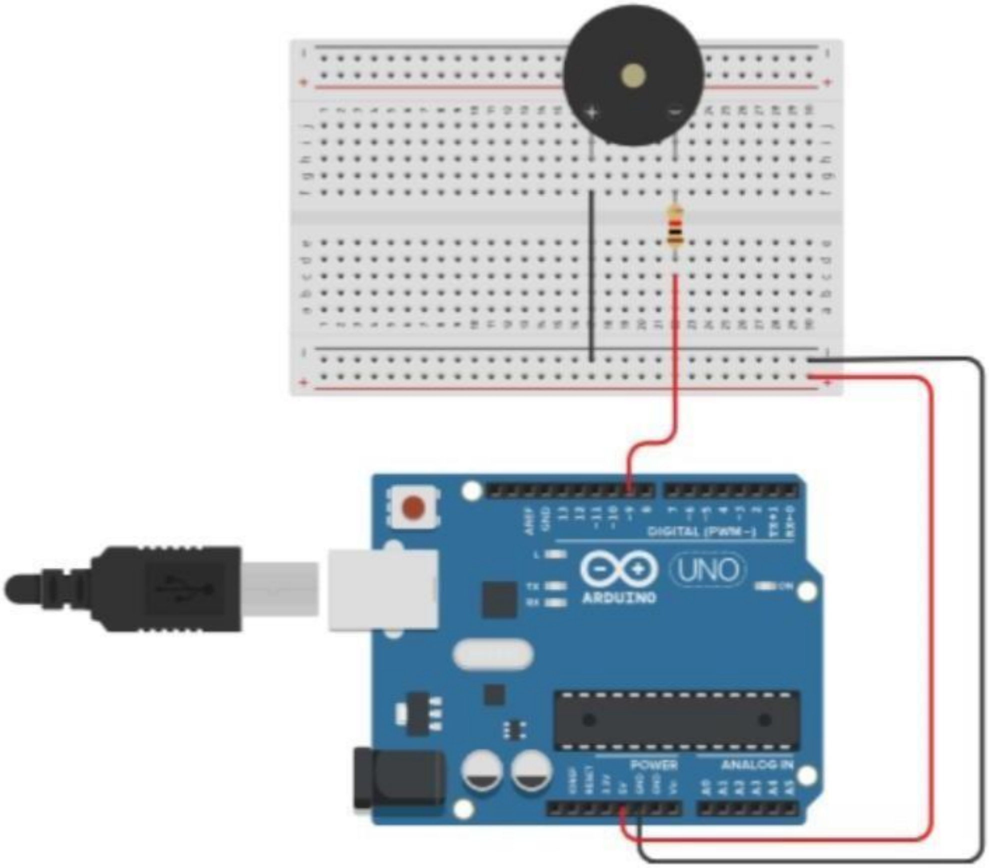
LED :
To turn on an LED, the Arduino needs to send a HIGH signal to one of it's pins. To turn off the LED, it needs to send a LOW signal to the pin. You can make the LED flash by changing the length of the HIGH and LOW states.

Piezo Speaker:
A Piezo Speaker is a loudspeaker that uses the piezoelectric effect for generating sound. The initial mechanical motion is created by applying a voltage to a piezoelectric material, and this motion is typically converted into audible sound using diaphragms and resonators.

Functions:
setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.
loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.
pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.
delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.
digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).
analogWrite(): Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to analogWrite(), the pin will generate a steady rectangular wave of the specified duty cycle until the next call to analogWrite() (or a call to digitalRead() or digitalWrite()) on the same pin.

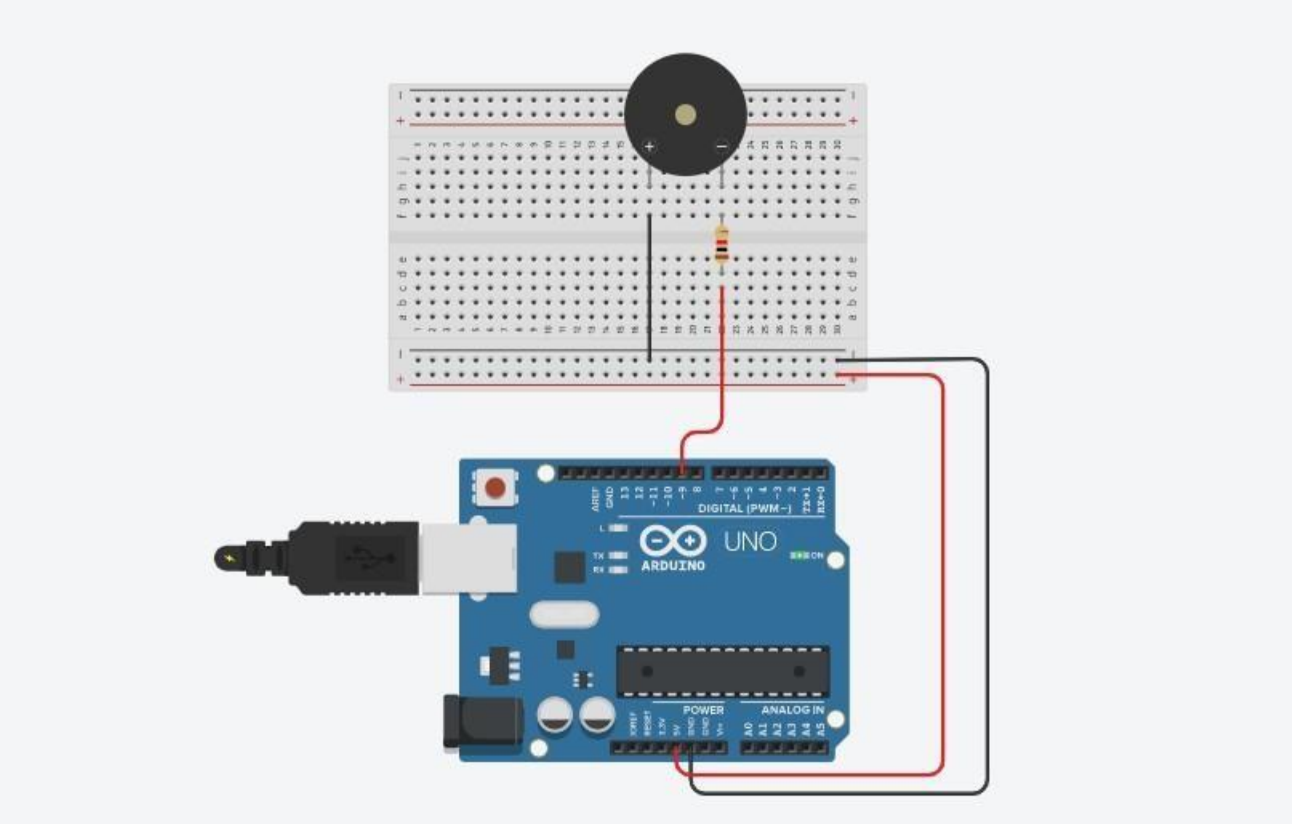
Circuit Diagram:



Code:

```
// C++ code
//
const int buzzer=9;

void setup()
{
  pinMode(buzzer, OUTPUT);
}

void loop()
{
  tone(buzzer, 2000);
  delay(1000);
  noTone(buzzer);
  delay(1000);
}
```

Output:



Conclusion:

Thus, learnt about basic components of lot like Arduino UNO (blink Arduino onboard LED), Breadboard , LED's and interfacing Piezo Speaker with Arduino for Buzz sound on Switch Button ON/OFF with connections of resistors. breadboard, Jumper wires & Arduino.

Practical No: 15

Aim: WAP to interface Piezo Speaker with Arduino and write a program to use the Piezo Speaker with Arduino to obtain on Switch Button ON/OFF.

Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To Learn Piezo Speaker
- To learn Push Button Switch
- Write a program to use the Piezo Speaker with Arduino to obtain on Switch Button ON/OFF

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In

alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Jumper Wires:

A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

LED :

To turn on an LED, the Arduino needs to send a HIGH signal to one of it's pins. To turn off the LED, it needs to send a LOW signal to the pin. You can make the LED flash by changing the length of the HIGH and LOW states.

Piezo Speaker:

A Piezo Speaker is a loudspeaker that uses the piezoelectric effect for generating sound. The initial mechanical motion is created by applying a voltage to a piezoelectric material, and this motion is typically converted into audible sound using diaphragms and resonators.

Push Button:

Image result for push button arduino wikipedia Pushbuttons or switches connect two points in a circuit when you press them. This example turns on the built-in LED on pin 13 when you press the button.

Functions:

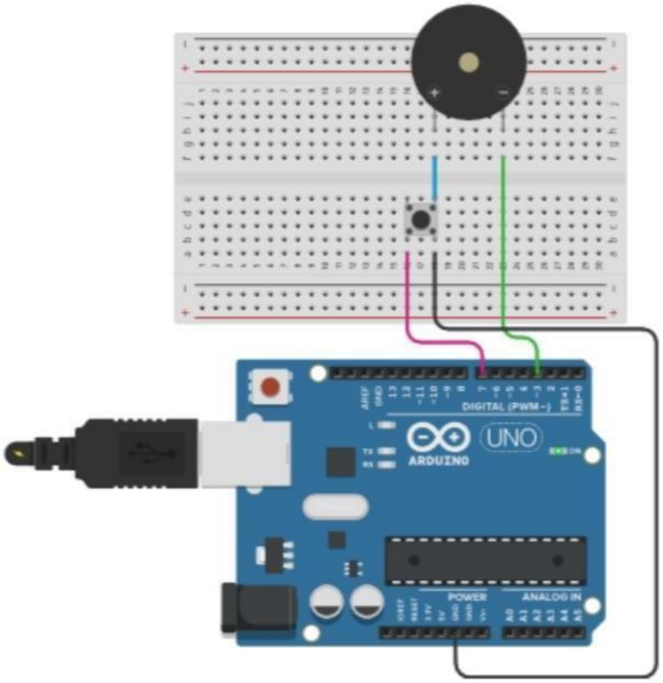
setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.
digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.
digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).
analogWrite(): Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to analogWrite(), the pin will generate a steady rectangular wave of the specified duty cycle until the next call to analogWrite() (or a call to digitalRead() or digitalWrite()) on the same pin.

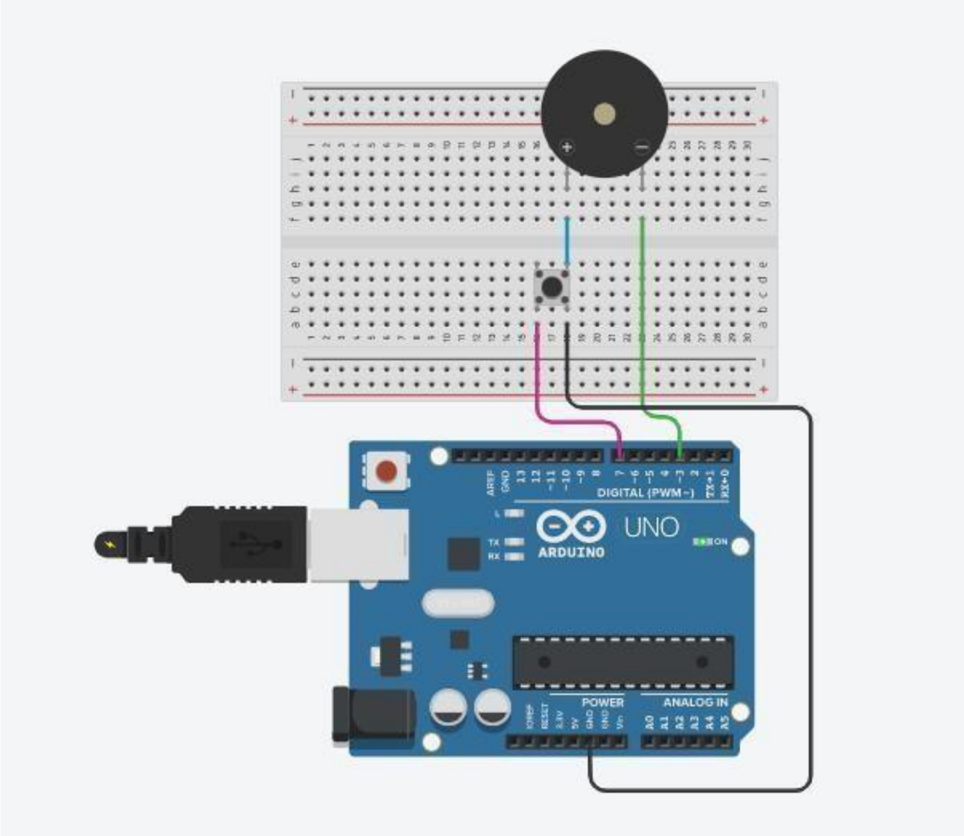
Circuit Diagram:



Code:

```
// C++ code
//

const int BUTTON_PIN=7;
const int BUZZER_PIN=3;

void setup()
{
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  pinMode(BUZZER_PIN,OUTPUT);
}

void loop()
{
  int buttonState=digitalRead(BUTTON_PIN);

  if(buttonState==LOW)
  {
    Serial.println("The button is being pressed ");
    digitalWrite(BUZZER_PIN, HIGH);
  }
  else
    if(buttonState==HIGH)
  {
    Serial.println("The button is being pressed");
    digitalWrite(BUZZER_PIN, LOW);
  }
}
```

Output:



Conclusion:

Thus, learnt about basic components of lot like Arduino UNO (blink Arduino onboard LED), Breadboard . LED's and interfacing Piezo Speaker with Arduino for Buzz sound on switch button ON/OFF with connections of resistors. breadboard, Jumper wires & Arduino.

Practical No: 16

Aim: WAP to interface Piezo Speaker with Arduino and write a program to use the Piezo. Speaker LED ON/OFF with Arduino to obtain on Switch Button ON/OFF.

Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To Learn Piezo Speaker
- To Learn LED.
- To learn Push Button Switch
- Write a program to use the Piezo Speaker LED ON/OFF with Arduino to obtain on Switch Button ON/OFF

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Jumper Wires:

A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

LED :

To turn on an LED, the Arduino needs to send a HIGH signal to one of it's pins. To turn off the LED, it needs to send a LOW signal to the pin. You can make the LED flash by changing the length of the HIGH and LOW states.

Piezo Speaker:

A Piezo Speaker is a loudspeaker that uses the piezoelectric effect for generating sound. The initial mechanical motion is created by applying a voltage to a piezoelectric material, and this motion is typically converted into audible sound using diaphragms and resonators.

Push Button:

Image result for push button arduino wikipedia Pushbuttons or switches connect two points in a circuit when you press them. This example turns on the built-in LED on pin 13 when you press the button.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

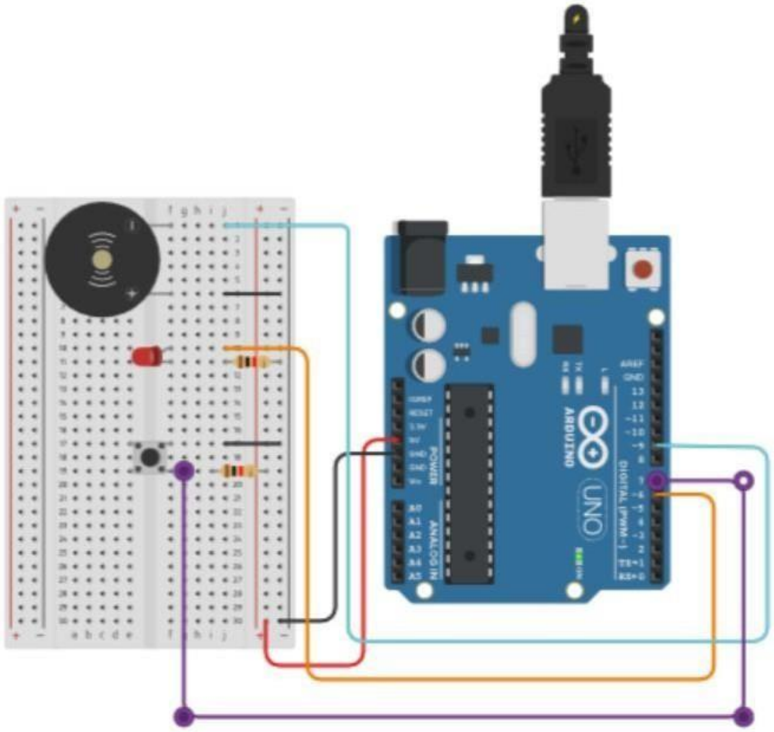
delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).

analogWrite(): Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to analogWrite(), the pin will generate a steady rectangular wave of the specified duty cycle until the next call to analogWrite() (or a call to digitalRead() or digitalWrite()) on the same pin.

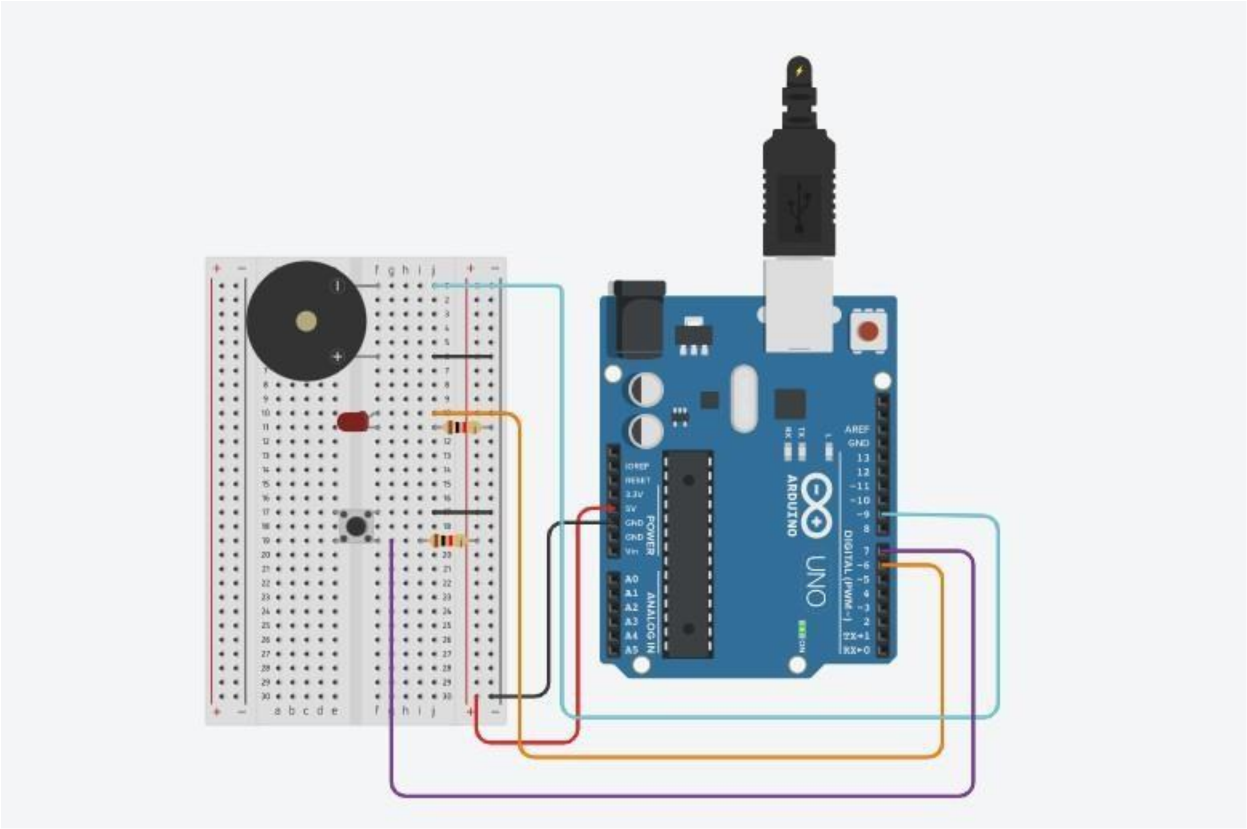
Circuit Diagram:



Code:

```
// C++ code
//

int buzzerPin=9;
int  buttonPin=7;
int ledPin=6;
const int toneFreq=600;

void setup()
{
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  int buttonState=digitalRead(buttonPin);
  if(buttonState==LOW)
  {
    digitalWrite(ledPin, HIGH);
    tone(buzzerPin,toneFreq);
  }
  else
  {
    digitalWrite(ledPin, LOW);
    noTone(buzzerPin);
  }
}
```

Output:



Conclusion:

Thus, learnt about basic components of lot like Arduino UNO (blink Arduino onboard LED), Breadboard . LED's and interfacing Piezo Speaker sound & LED ON/OFF with Arduino to press the Switch Button ON/OFF with connections of resistors, breadboard, Jumper wires & Arduino.

Practical No: 17

Aim: WAP to interface 2 Push buttons, a speaker with Arduino and WAP to turn ON LED and generate 2 different notes on two button keyboard.

Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To learn about Push buttons and speaker
- Write a program to turn ON LED and generate two different notes on two button keyboard.

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In

alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Jumper Wires:

A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

LED :

LED means red, blue and green LEDs. RGB LED products combine these three colors to produce over 16 million hues of light. Note that not all colors are possible. Some colors are “outside” the triangle formed by the RGB LEDs. Also, pigment colors such as brown or pink are difficult, or impossible, to achieve.

Push Button:

Image result for push button arduino wikipedia Pushbuttons or switches connect two points in a circuit when you press them. This example turns on the built-in LED on pin 13 when you press the button.

Speaker:

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

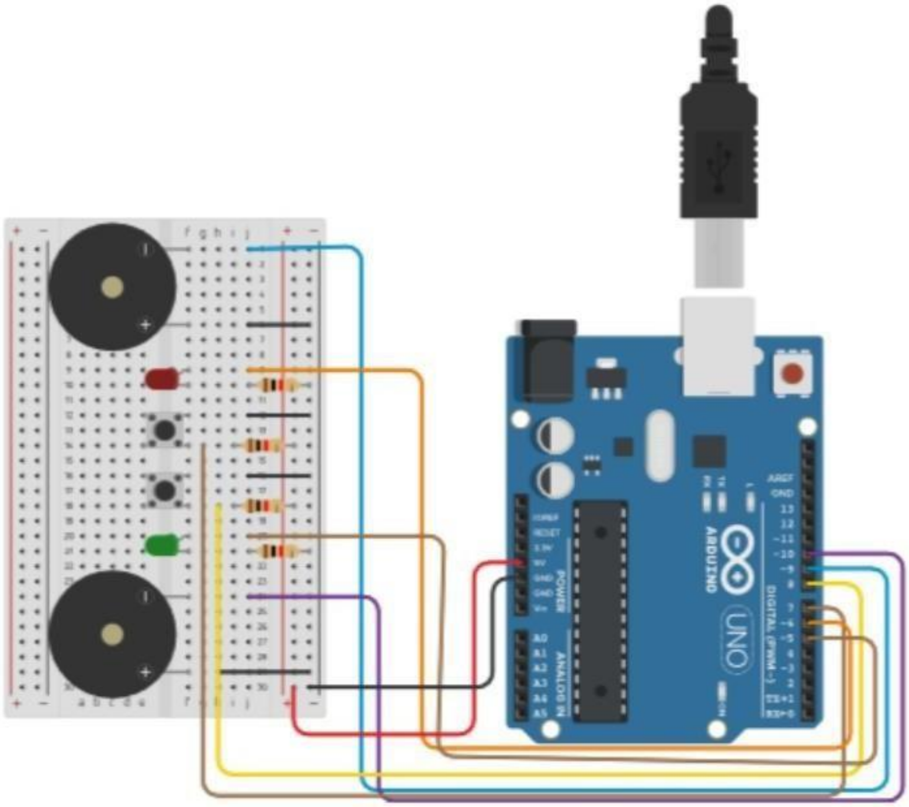
delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).

analogWrite(): Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to analogWrite(), the pin will generate a steady rectangular wave of the specified duty cycle until the next call to analogWrite() (or a call to digitalRead() or digitalWrite()) on the same pin.

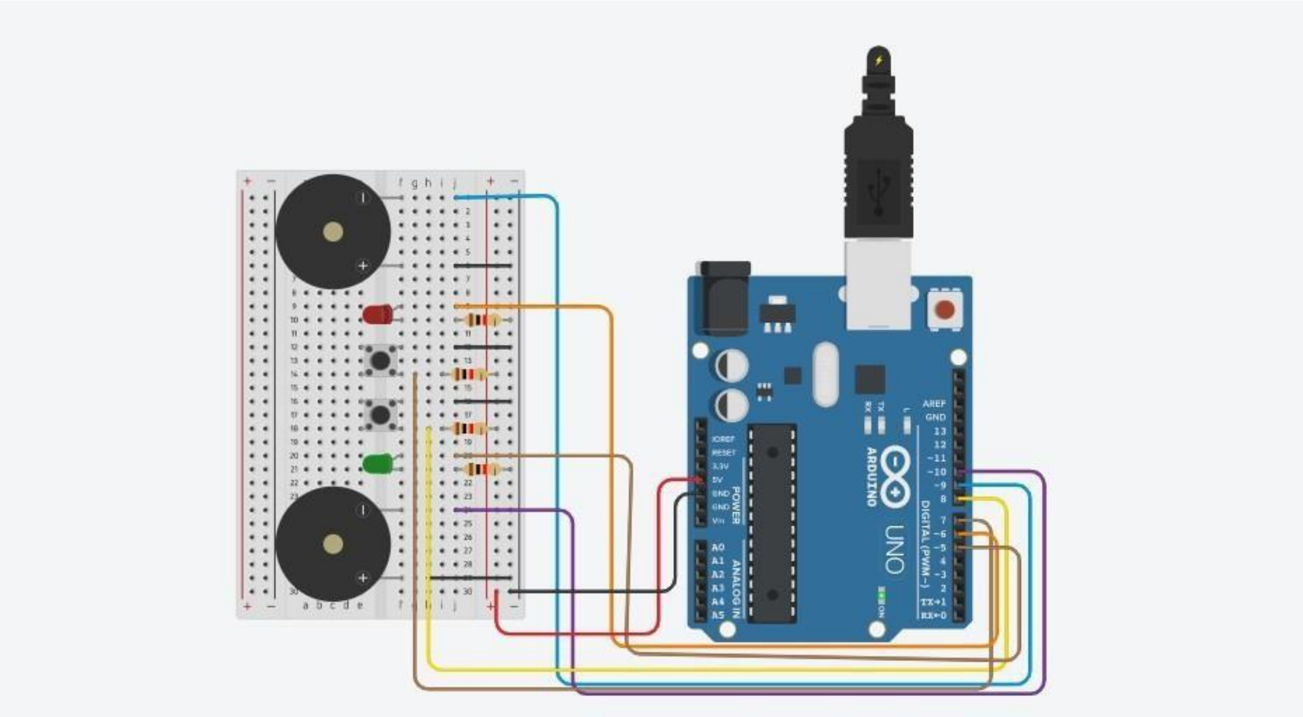
Circuit Diagram:



Code:

```
// C++ code
//

int buzzerPin1=9;
int buzzerPin2=10;
int buttonPin1=7;
int buttonPin2=8;
int ledPin1=6;
int ledPin2=5;

const int toneFreq1=523;
const int toneFreq2=349;

void setup()
{
  pinMode(buzzerPin1, OUTPUT);
  pinMode(buzzerPin2, OUTPUT);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);

}

void loop()
{
  int buttonState=digitalRead(buttonPin1);
  if(buttonState==LOW)
  {
    digitalWrite(ledPin1, HIGH);
    tone(buzzerPin1,toneFreq1);
  }
  else
  {
    digitalWrite(ledPin1, LOW);
    noTone(buzzerPin1);
  }
  {
  int buttonState=digitalRead(buttonPin2);
  if(buttonState==LOW)
  {
    digitalWrite(ledPin2, HIGH);
    tone(buzzerPin2,toneFreq2);
  }
  else
  {
    digitalWrite(ledPin2, LOW);
    noTone(buzzerPin2);
  }
 }
 }
```

Output:



Conclusion:

Thus, learnt about basic components of lot like Arduino UNO(blink Arduino onboard LED), Breadboard LED's and interfacing 2 Push buttons with Arduino to turn ON2 LEDs and Speakers with different sounds with connections of resistors, breadboard, Jumper wires & Arduino.

Practical No: 18

Aim: WAP to interface Piezo Speaker & Multicolored LED with Arduino and write a program to create fast LED stream that goes back and forward while interfacing with buzzer.

Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To learn Piezo Speaker
- To learn LEDs
- To learn Push Button Switch.
- Write a program to create fast LED stream that goes back and forward while interfacing with buzer.

Theory:

Concept of Arduino UNO:

Arduino/Genuino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Breadboard:

A breadboard allows for easy and quick creation of temporary electronic circuits or to carry out experiments with circuit design. Breadboards enable developers to easily connect components or wires thanks to the rows and columns of internally connected spring clips underneath the perforated plastic enclosure.

Resistor:

A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit. Resistors can also be used to provide a specific voltage for an active device such as a transistor. All other factors being equal, in a direct-current (DC) circuit, the current through a resistor is inversely proportional to its resistance, and directly proportional to the voltage across it. This is the well-known Ohm's Law. In alternating-current (AC) circuits, this rule also applies as long as the resistor does not contain inductance or capacitance.

Jumper Wires:

A jumper wire is an electric wire that connects remote electric circuits used for printed circuit boards. By attaching a jumper wire on the circuit, it can be short-circuited and short-cut (jump) to the electric circuit.

LED :

LED means red, blue and green LEDs. RGB LED products combine these three colors to produce over 16 million hues of light. Note that not all colors are possible. Some colors are “outside” the triangle formed by the RGB LEDs. Also, pigment colors such as brown or pink are difficult, or impossible, to achieve.

Push Button:

Image result for push button arduino wikipedia Pushbuttons or switches connect two points in a circuit when you press them. This example turns on the built-in LED on pin 13 when you press the button.

Piezo Speaker:

A Piezo Speaker is a loudspeaker that uses the piezoelectric effect for generating sound. The initial mechanical motion is created by applying a voltage to a piezoelectric material, and this motion is typically converted into audible sound using diaphragms and resonators.

Functions:

setup(): The setup() function is called when a sketch starts. Use it to initialize variables, pin modes, start using libraries, etc. The setup function will only run once, after each powerup or reset of the Arduino board.

loop():After creating a setup() function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

pinmode():The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

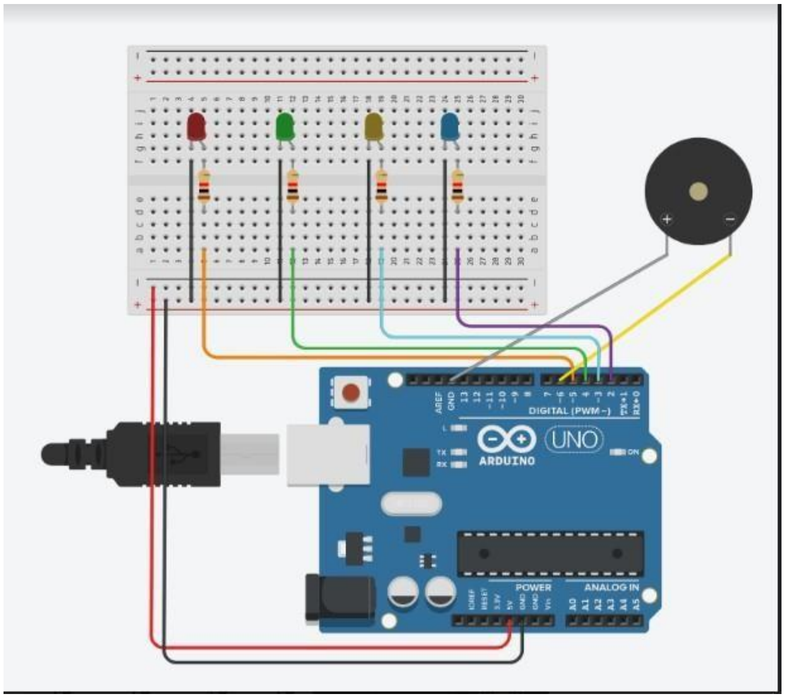
delay():The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell wether the voltage at this pin is high (~ 5V) or low (~ 0V) or, in other words, if the pin is at logic state 1 or 0 (or HIGH/LOW).

analogWrite(): Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to analogWrite(), the pin will generate a steady rectangular wave of the specified duty cycle until the next call to analogWrite() (or a call to digitalRead() or digitalWrite()) on the same pin.

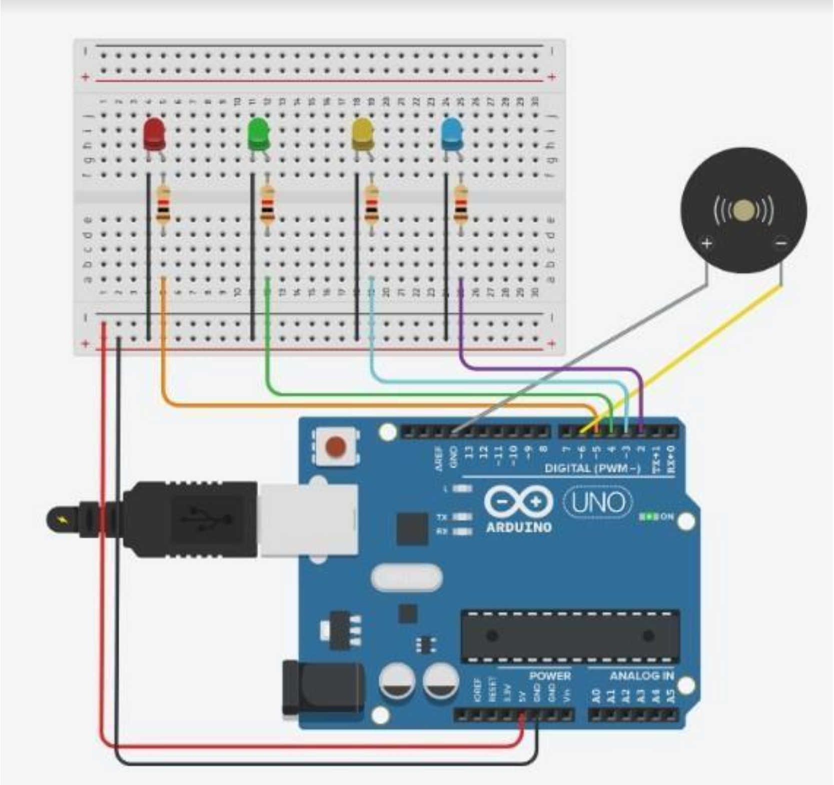
Circuit Diagram:



Code:

```
// C++ code
//

int blueLED=2;
int yellowLED=3;
int greenLED=4;
int  redLED=5;
int buzzer=6;
//const int toneFreq=261;

void setup()
{
  pinMode(blueLED, OUTPUT);
  pinMode(yellowLED, OUTPUT);
  pinMode(greenLED, OUTPUT);
  pinMode(redLED, OUTPUT);
  pinMode(buzzer, OUTPUT);
  digitalWrite(buzzer,HIGH);
}

void loop()
{
  digitalWrite(blueLED, HIGH);
  delay(50);
  digitalWrite(blueLED, LOW);
  digitalWrite(yellowLED, HIGH);
  delay(50);
  digitalWrite(yellowLED, LOW);
  digitalWrite(greenLED, HIGH);
  delay(50);
  digitalWrite(greenLED, LOW);
  digitalWrite(redLED, HIGH);
  delay(50);
  digitalWrite(redLED, LOW);
  digitalWrite(greenLED, HIGH);
  delay(50);
  digitalWrite(greenLED, LOW);
  digitalWrite(yellowLED, HIGH);
  delay(50);
  digitalWrite(yellowLED, LOW);
}
```

Output:



Conclusion:

Thus, learnt about basic components of lot like Arduino UNO(blink Arduino onboard LED),Breadboard ,LED's and interfacing fast LED stream that goes back and forward while interfacing with buzzer with connections of resistors, breadboard, Jumper wires & Arduino.