

[INTERNET OF THINGS]

Practical-3

-:AIM:-

Arduino architecture and basic programming.

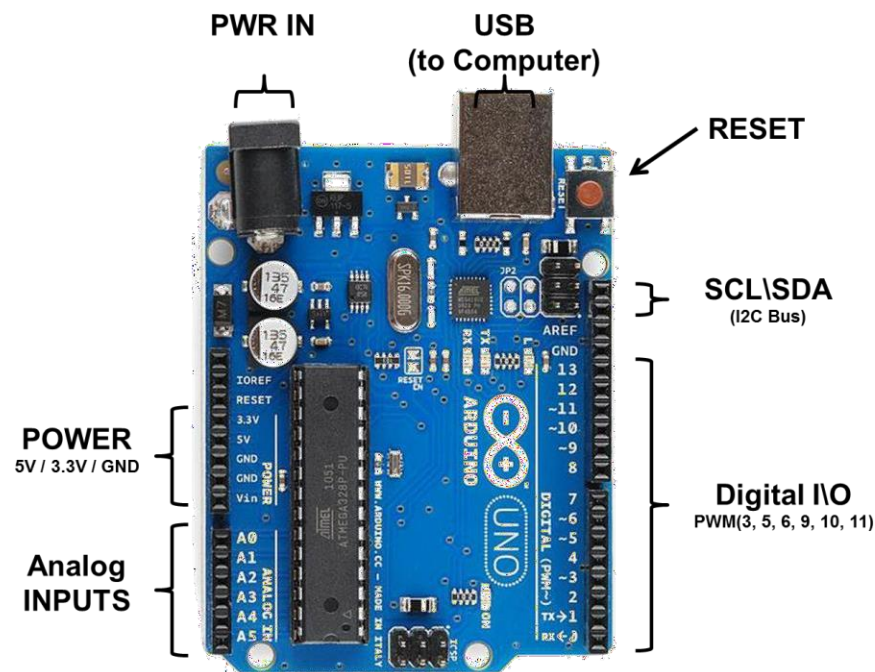
Submitted By: Dharmay Sureja

Enrollment No:17012011056

AIM:- Arduino architecture and basic programming.

Theory:

Arduino is an open-source physical computing platform designed to make experimenting with electronics and programming more fun and intuitive. Arduino has its own unique, simplified programming language and a lots of premade examples and tutorials exists. With Arduino you can easily explore lots of small-scale sensors and actuators like motors, temperature sensors, etc.



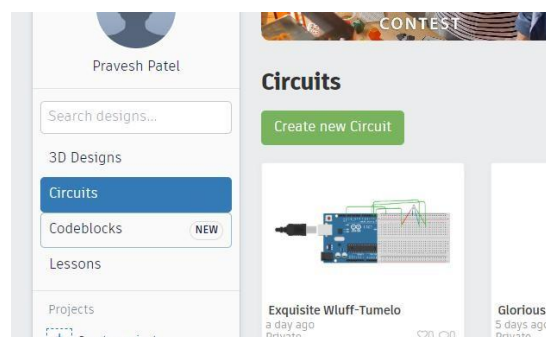
Practical-3

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

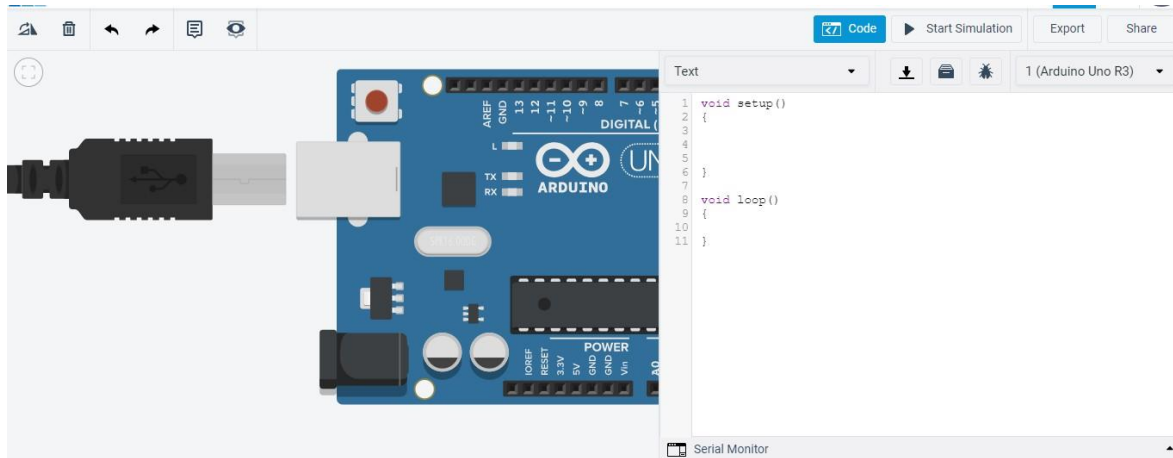
Arduino Programming On tinkercad

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

1. Open www.tinkercad.com
2. Login using Google Account
3. After login click on Circuit design > create new Circuit



4. Search Arduino board and drag & drop in workspace
5. Write code.



Arduino function:-

- **pinMode() :**
 - Configures the specified pin to behave either as an input or an output. See the Digital Pins page for details on the functionality of the pins.
 - Syntax - pinMode(pin, mode)
 - Parameters
 - pin: the Arduino pin number to set the mode of.
 - mode: INPUT, OUTPUT, or INPUT_PULLUP. See the Digital Pins page for a more complete description of the functionality.
 - Returns – Nothing
- **digitalRead() :**
 - Reads the value from a specified digital pin, either HIGH or LOW.
 - Syntax - digitalRead(pin)
 - Parameters
 - pin: the Arduino pin number you want to read
 - Returns - HIGH or LOW

- **digitalWrite() :**

- Write a HIGH or a LOW value to a digital pin.
- Syntax - digitalWrite(pin, value)
- Parameters
 - pin: the Arduino pin number.
 - value: HIGH or LOW.
- Returns – Nothing

- **delay() :**

- Pauses the program for the amount of time (in milliseconds) specified as parameter. (There are 1000 milliseconds in a second.)
- Syntax - delay(ms)
- Parameters
 - ms: the number of milliseconds to pause. Allowed data types: unsigned long.
- Returns – Nothing

- **analogRead() :**

- Reads the value from the specified analog pin.
- Syntax - analogRead(pin)
- Parameters
 - pin: the name of the analog input pin to read from (A0 to A5 on most boards, A0 to A6 on MKR boards, A0 to A7 on the Mini and Nano, A0 to A15 on the Mega).
- Returns - The analog reading on the pin. Although it is limited to the resolution of the analog to digital converter (0-1023 for 10 bits or 0-4095 for 12 bits). Data type: int.

- **analogWrite() :**

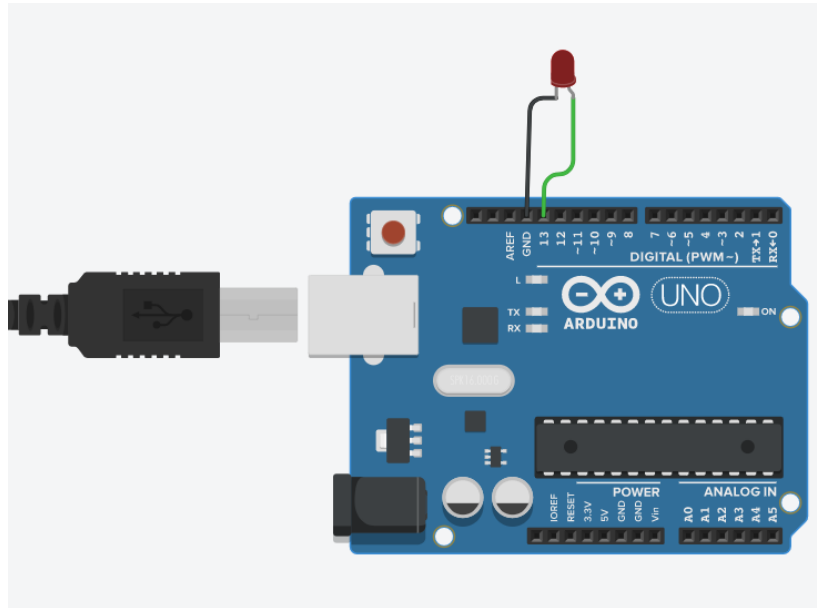
- Writes an analog value (PWM wave) to a pin.
- Syntax - analogWrite(pin, value)
- Parameters
 - pin: the Arduino pin to write to. Allowed data types: int.
 - value: the duty cycle: between 0 (always off) and 255 (always on).
 - Allowed data types: int.
- Returns – Nothing

Experiment

1. Working with LED a. LED ON

Components used : Arduino Uno R3, LED

Circuit:



Code:

```
void setup()
{
  pinMode(13, OUTPUT);
}

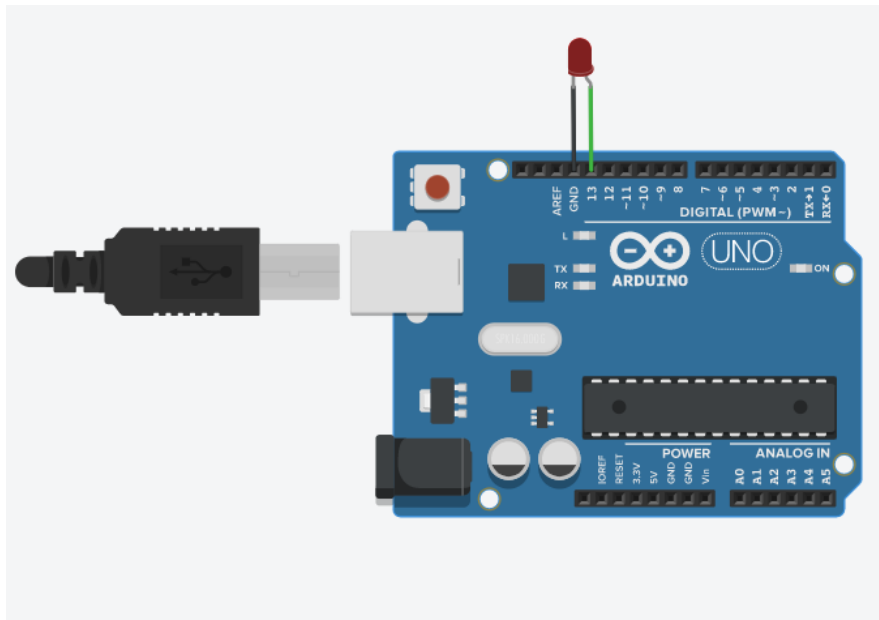
void loop()
{
  digitalWrite(13, HIGH);
}
```

b. LED Blinking

Components used : Arduino Uno R3, LED

Circuit:

Practical-3



Code:

```
void setup()
{
  pinMode(13, OUTPUT);
}

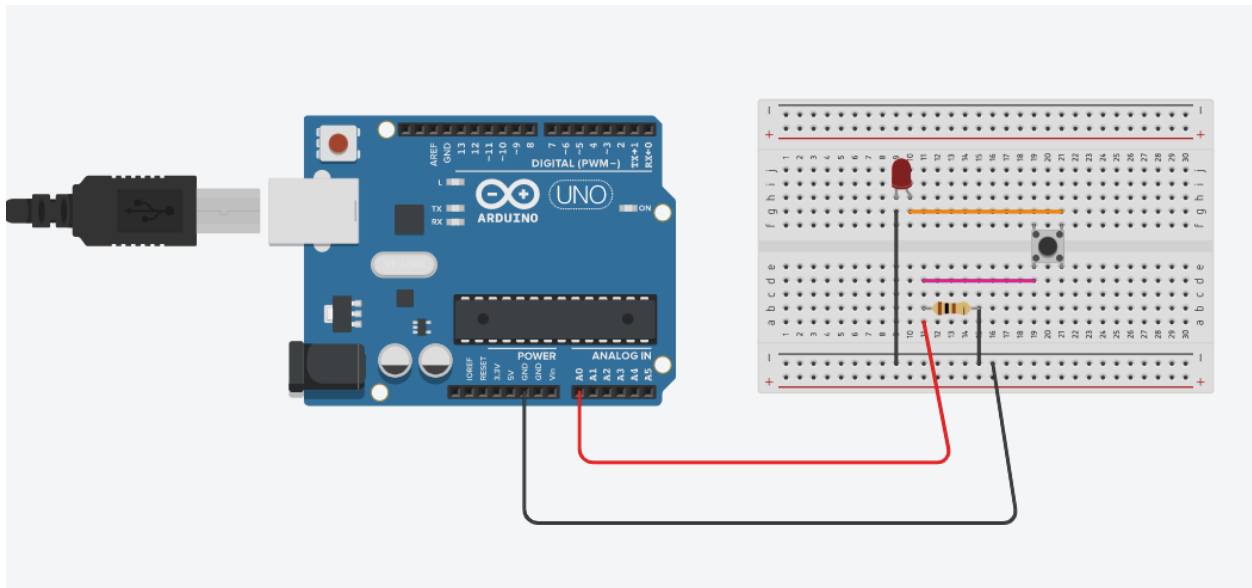
void loop()
{
  digitalWrite(13, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(13, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

c. LED ON/OFF using push button

Components used : Arduino Uno R3, LED, Push button, Resistor

Practical-3

Circuit:



Code:

```
void setup()
{
  pinMode(A0, OUTPUT);
}

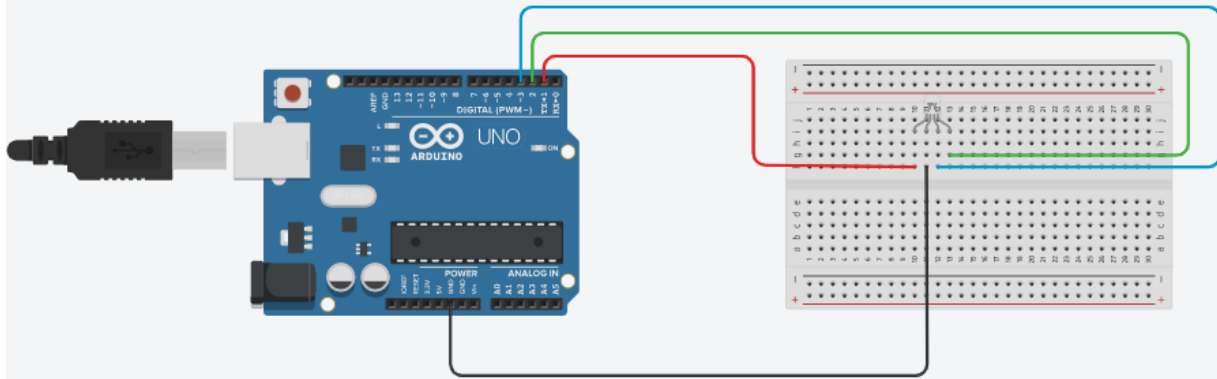
void loop()
{
  analogWrite(A0, 255);
}
```

d. Working with RGB LED

Components used : Arduino Uno R3, RGB - LED

Circuit:

Practical-3



Code:

```
void setup()
{
  pinMode(1, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
}

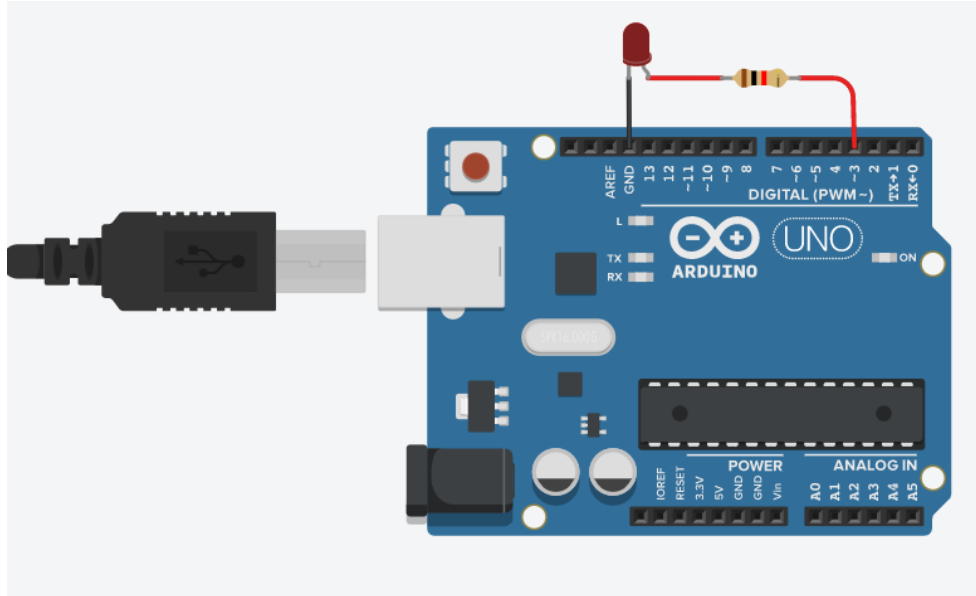
void loop()
{
  digitalWrite(1, HIGH);
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(1, LOW);
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(1, LOW);
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

Practical-3

2. Increase and decrease the brightness of LED

Components used : Arduino Uno R3, LED, Resistor

Circuit:



Code:

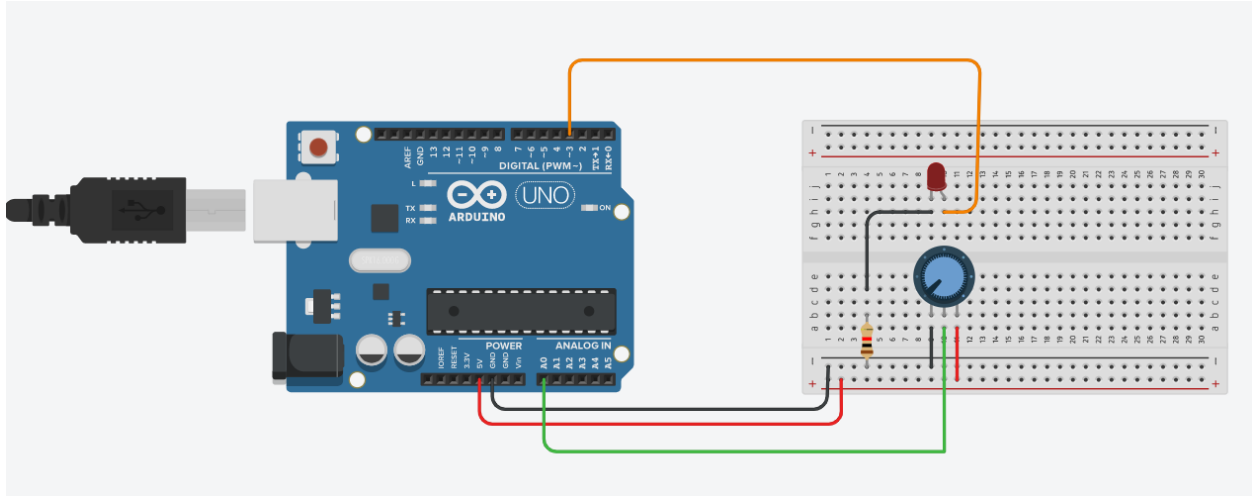
```
void setup()
{
  pinMode(3, OUTPUT);
}

void loop()
{
  for (int i = 0; i <= 255; ) {
    analogWrite(3, i);
    delay(100);
    i+=15;
  }
}
```

3. Increase and decrease the brightness of LED using potentiometer

Components used : Arduino Uno R3, LED, Resistor, Potentiometer

Circuit:



Code:

```
void setup()
{
  pinMode(A0, INPUT);
  pinMode(3, OUTPUT);
}

void loop()
{
  int p_val=analogRead(A0);
  int v=map(p_val,0,1023,0,255);
  analogWrite(3,v);
}
```