

Practical No: 19

Aim: To interface a 7 Segment Display with Arduino & WAP to turn ON Display for displaying values from 0 to 9 & A to F.

Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To Learn about 7 Segment Display
- WAP to interface a 7 Segment with Arduino & turning display for displaying values from 0 to 9 & A to F.

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.

7 segment Display:

The 7-segment display, also written as “seven segment display”, consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown. Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit (both Decimal and Hex) to be displayed.

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.

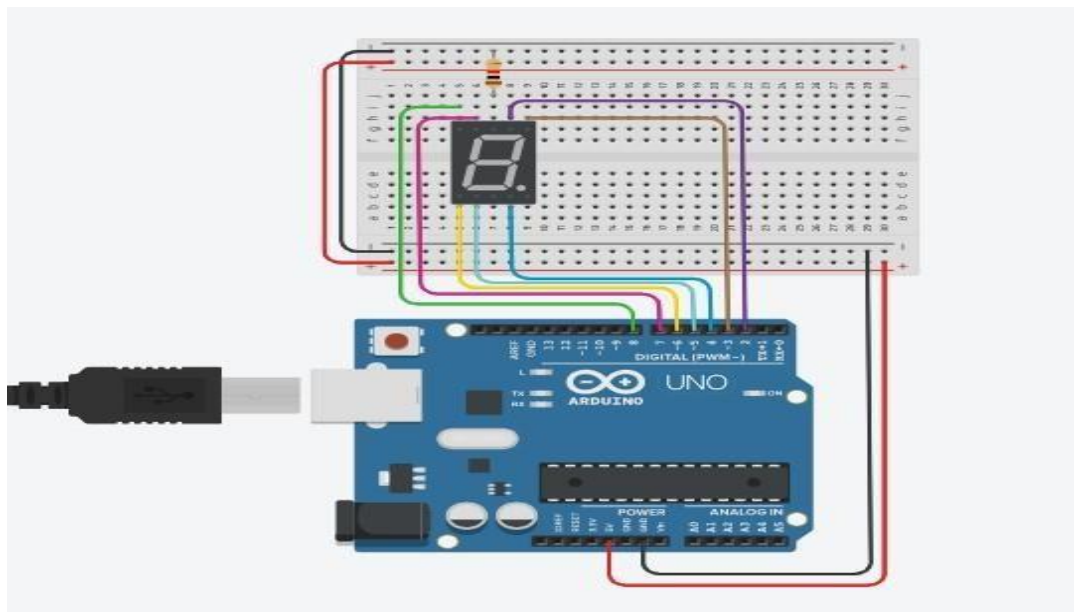
pinmode(): The pinMode() function is used to configure a specific pin to behave either as an input or an output. It is possible to enable the internal pull-up resistors with the mode INPUT_PULLUP. Additionally, the INPUT mode explicitly disables the internal pull-ups.

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 digitalWrite() function is used to read the logic state at a pin. It is capable to tell whether 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 digitalWrite() or digitalWrite()) on the same pin.

CircuitDiagram:**Code:**

```
// C++ code
//make an array to save sev seg pin configuration of numbers
int num_array[10][7]={1, 1, 1, 1, 1, 1, 0}, //0
                      {0, 1, 1, 0, 0, 0, 0}, //1
                      {1, 1, 0, 1, 1, 0, 1}, //2
                      {1, 1, 1, 1, 0, 0, 1}, //3
                      {0, 1, 1, 0, 0, 1, 1}, //4
                      {1, 0, 1, 1, 0, 1, 1},    //5
                      {1, 0, 1, 1, 1, 1, 1},    //6
                      {1, 1, 1, 0, 0, 0, 0},    //7
                      {1, 1, 1, 1, 1, 1, 1},    //8
                      {1, 1, 1, 0, 0, 1, 1}}; //9
```

```

int letter_array[6][7]={1, 1, 1, 0, 1, 1, 2}, //A
                        {0, 0, 1, 1, 1, 1, 1}, //b
                        {0, 0, 0, 1, 1, 0, 1}, //c
                        {0, 1, 1, 1, 1, 0, 1}, //d
                        {1, 0, 0, 1, 1, 1, 1}, //E
                        {1, 0, 0, 0, 1, 1, 1}}; //f

int segAtoGpinouts[7]={2, 3, 4, 5, 6, 7, 8};

//Function Header
void WriteToSSD(int *);

void setup()
{
    //set pin nodes
    for (int i=0;i<7;i++)
    {
        pinMode(segAtoGpinouts[i],OUTPUT);
    }
}

void loop()
{
    //counter loop
    for( int letter_counter =0; letter_counter <6; letter_counter++)
    {
        delay(1000);

        WriteToSSD(letter_array[letter_counter]);
    }

    delay(3000);

    for(int num_counter =0;num_counter<=9;num_counter++)
    {
        delay(1000);
    }
}

```

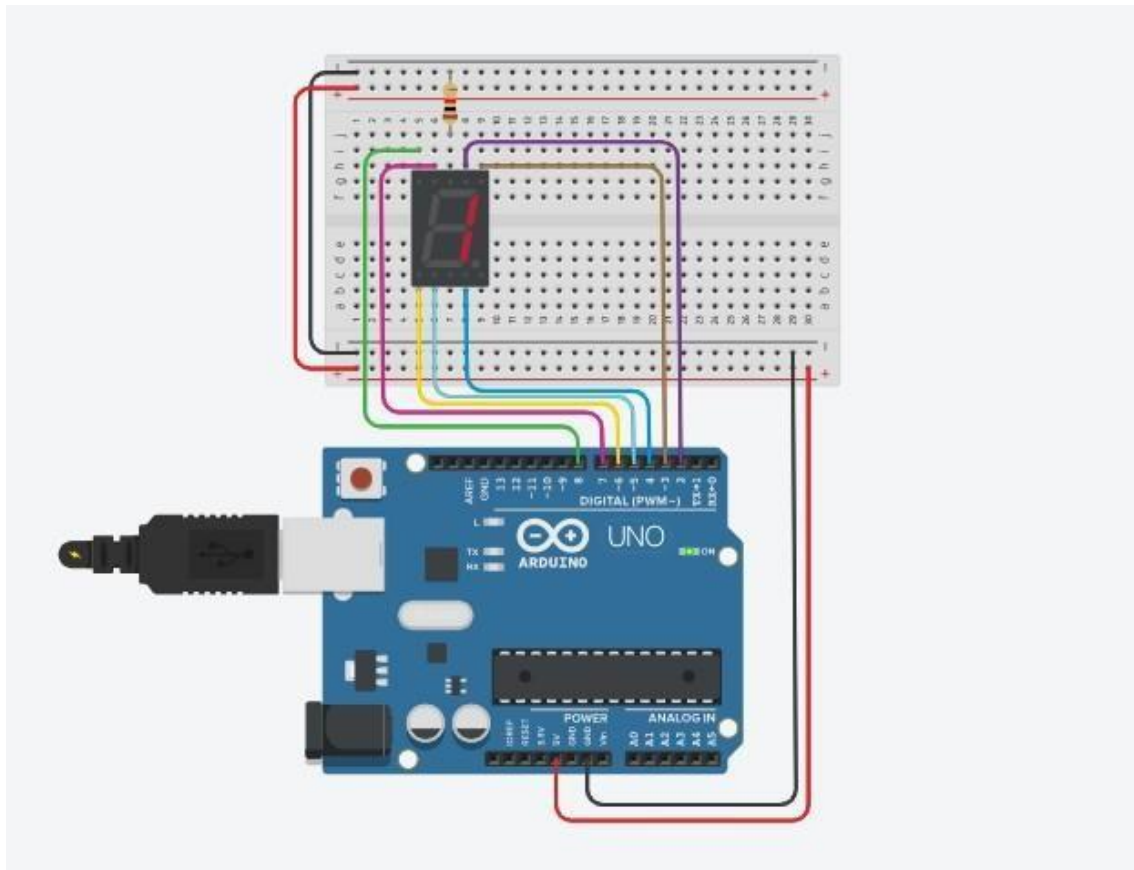
```

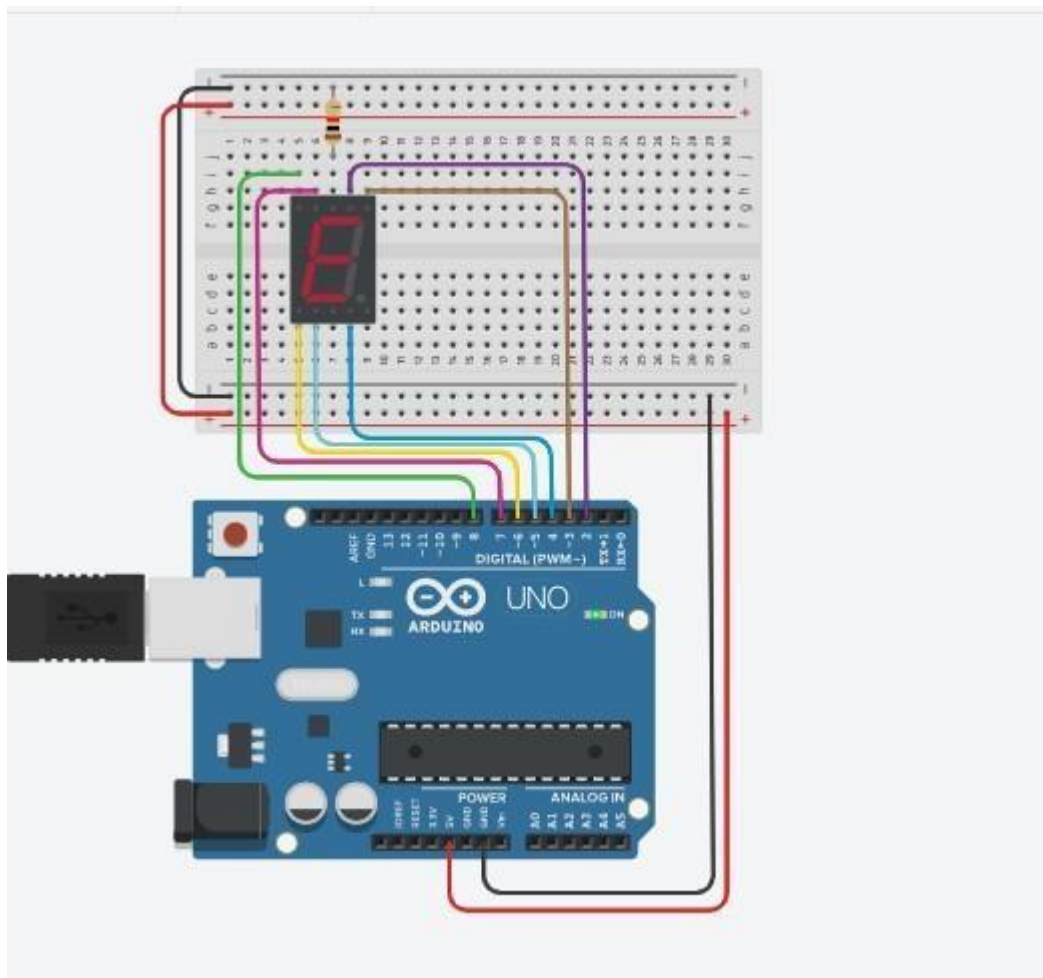
WriteToSSD(num_array[num_counter]);
} delay(3000);
}

void WriteToSSD(int * segmentArray)
{
  for(int i=0;i<7;i++)
    digitalWrite(segAtoGpinouts[i],segmentArray[i]);
}

```

Output:



**Conclusion:**

Thus, learnt about basic components of lot like Arduino UNO(blink Arduino onboard LED),Breadboard 7segment display and interfacing to turn ON 7 segment Display for displaying values from 0 to 9 & A to F with connections of resistors, breadboard, Jumper wires & Arduino.

Practical No: 20

Aim: To interface 16-2 Character LCD Module with Arduino & WAP to display running with LCD character on parallel interface LCD controller chip.

Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To Learn about 16x2 LCD.
- WAP to Display running with Character LCDS based on parallel interface LCD controller chip.

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.

16x2 LCD:

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data.

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.

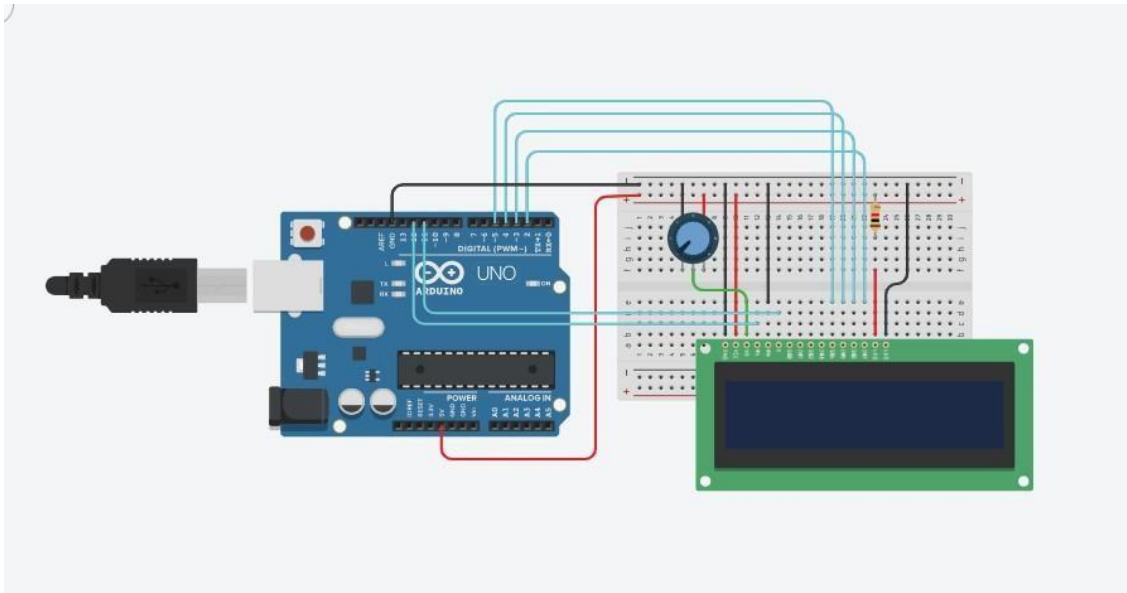
digitalwrite() : Write a HIGH or a LOW value to a digital pin. If the pin has been configured as an OUTPUT with pinMode() , its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH , 0V (ground) for LOW.

digitalRead(): The digitalRead() function is used to read the logic state at a pin. It is capable to tell whether 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:

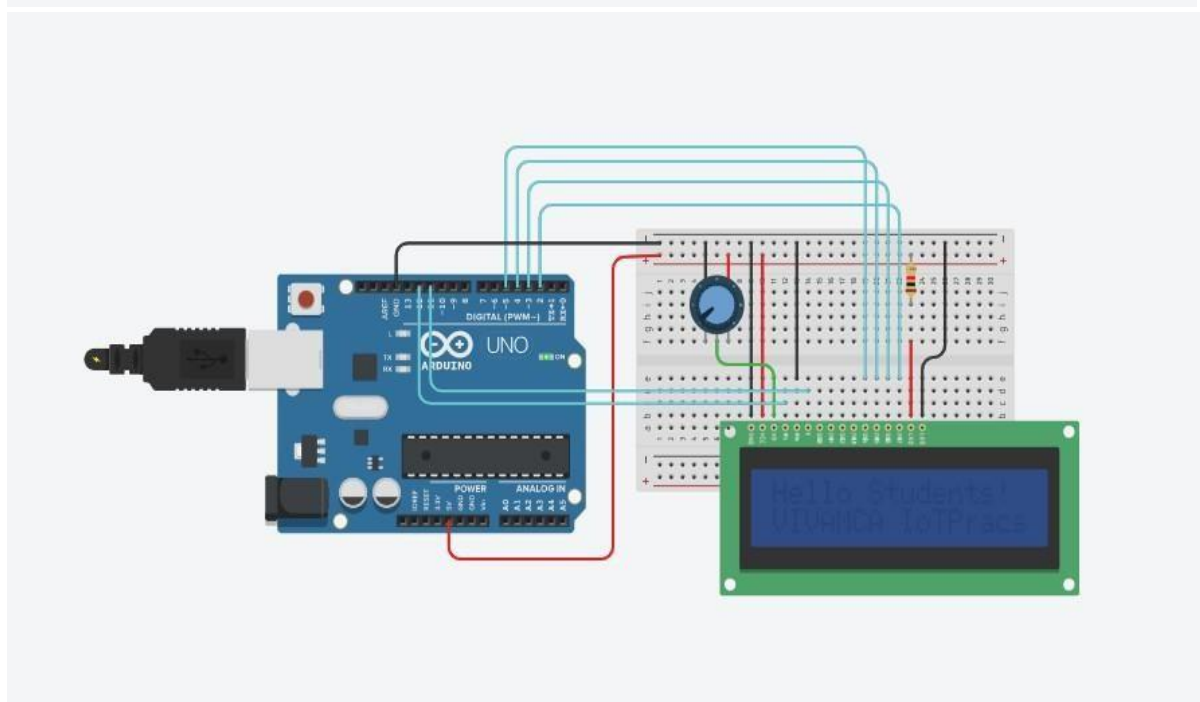
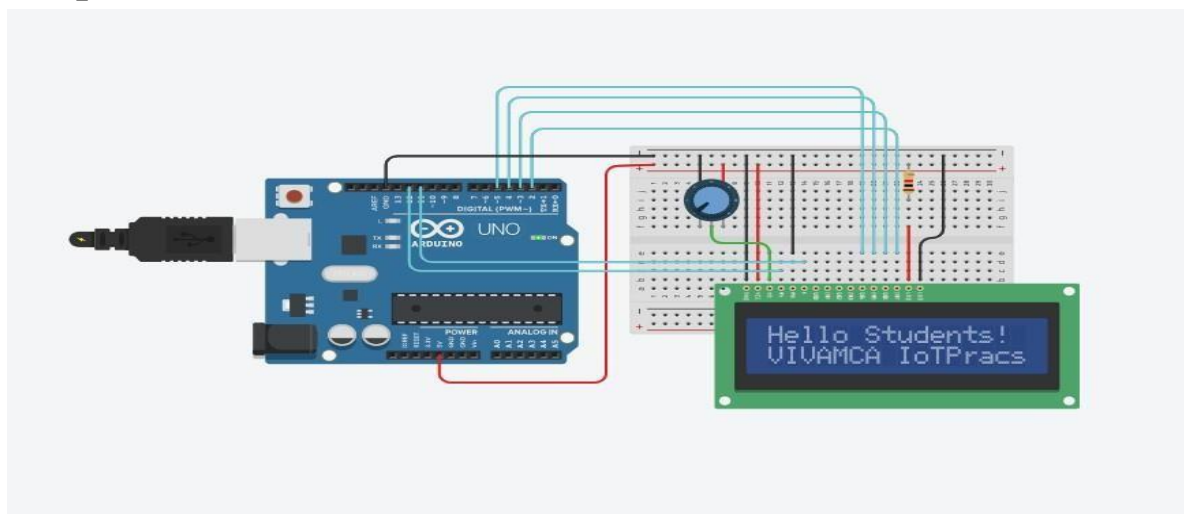


Code: // C++ code

```
//
#include<LiquidCrystal.h>
LiquidCrystal lcd(12,11,5,4,3,2);

void setup()
{
  lcd.begin(16,2);
  lcd.clear();
}

void loop()
{
  lcd.print("Hello Students!");
  lcd.setCursor(0,1);
  lcd.print("VIVAMCA IoTPracs");
}
```

Output:

Changing the potentiometer

Conclusion:

Thus learnt about basic components of IoT like Arduino UNO(blink Arduino onboard LED), Breadboard 7segment display and interfacing 16x2 LCD display for displaying running with LCD to custom character generator with LCD controller chip with connections of resistors,breadboard,Jumper wires&Arduino.

Practical No: 21

Aim: To interface 16-2 Character LCD Module with Arduino & WAP to display running with LCD character on parallel interface LCD controller chip.

Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To Learn about 16x2 LCD.
- WAP to Display running with Character LCDS based on parallel interface LCD controller chip.

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.

16x2 LCD:

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data.

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.

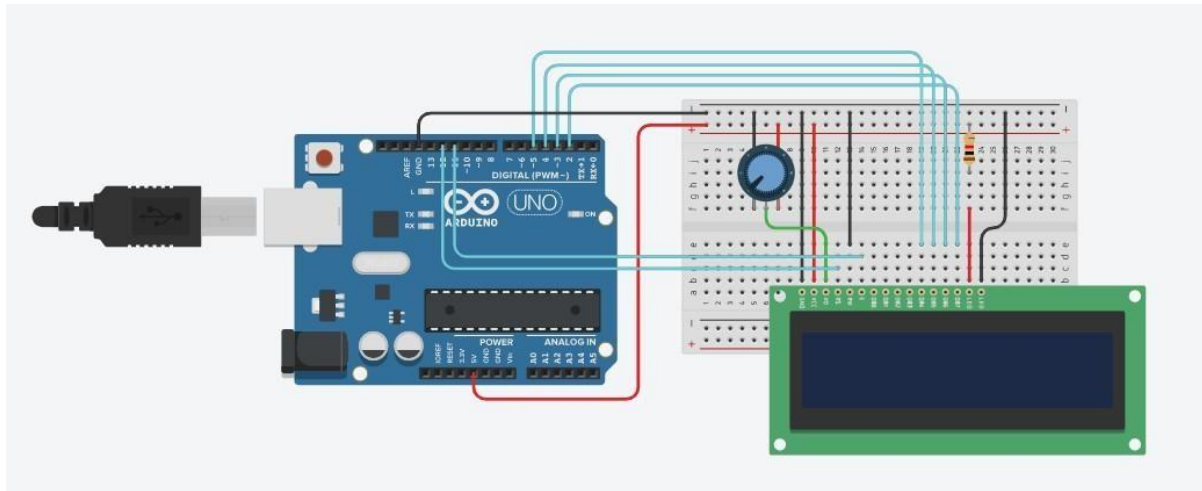
delay(): The way the delay() function works is pretty simple. It accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on to the next line of code when it encounters this function.

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 whether 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
//
// include the library code:
#include <LiquidCrystal.h>

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

// make some custom characters:
byte Heart[8] = {
  0b00000,
  0b01010,
  0b11111,
  0b11111,
  0b01110,
  0b00100,
  0b00000,
  0b00000
};
```

```
byte Bell[8] = {  
0b00100,  
0b01110,  
0b01110,  
0b01110,  
0b11111,  
0b00000,  
0b00100,  
0b00000  
};
```

```
byte Alien[8] = {  
0b11111,  
0b10101,  
0b11111,  
0b11111,  
0b01110,  
0b01010,  
0b11011,  
0b00000  
};
```

```
byte Check[8] = {  
0b00000,  
0b00001,  
0b00011,  
0b10110,  
0b11100,  
0b01000,  
0b00000,  
0b00000  
};
```

```
byte Speaker[8] = {  
0b00001,  
0b00011,  
0b01111,  
0b01111,  
0b01111,  
0b00011,  
0b00001,  
0b00000  
};
```

```
};
```

```
byte Sound[8] = {  
0b00001,  
0b00011,  
0b00101,  
0b01001,  
0b01001,  
0b01011,  
0b11011,  
0b11000  
};
```

```
byte Skull[8] = {  
0b00000,  
0b01110,  
0b10101,  
0b11011,  
0b01110,  
0b01110,  
0b00000,  
0b00000  
};
```

```
byte Lock[8] = {  
0b01110,  
0b10001,  
0b10001,  
0b11111,  
0b11011,  
0b11011,  
0b11111,  
0b00000  
};
```

```
void setup()  
{  
    // initialize LCD and set up the number of columns and rows:  
    lcd.begin(16, 2);  
  
    // create a new character  
    lcd.createChar(0, Heart);
```

```
// create a new character
lcd.createChar(1, Bell);
// create a new character
lcd.createChar(2, Alien);
// create a new character
lcd.createChar(3, Check);
// create a new character
lcd.createChar(4, Speaker);
// create a new character
lcd.createChar(5, Sound);
// create a new character
lcd.createChar(6, Skull);
// create a new character
lcd.createChar(7, Lock);

// Clears the LCD screen
lcd.clear();

// Print a message to the lcd.
lcd.print("Custom Character");
}

// Print All the custom characters
void loop()
{
    lcd.setCursor(0, 1);
    lcd.write(byte(0));

    lcd.setCursor(2, 1);
    lcd.write(byte(1));

    lcd.setCursor(4, 1);
    lcd.write(byte(2));

    lcd.setCursor(6, 1);
    lcd.write(byte(3));

    lcd.setCursor(8, 1);
    lcd.write(byte(4));

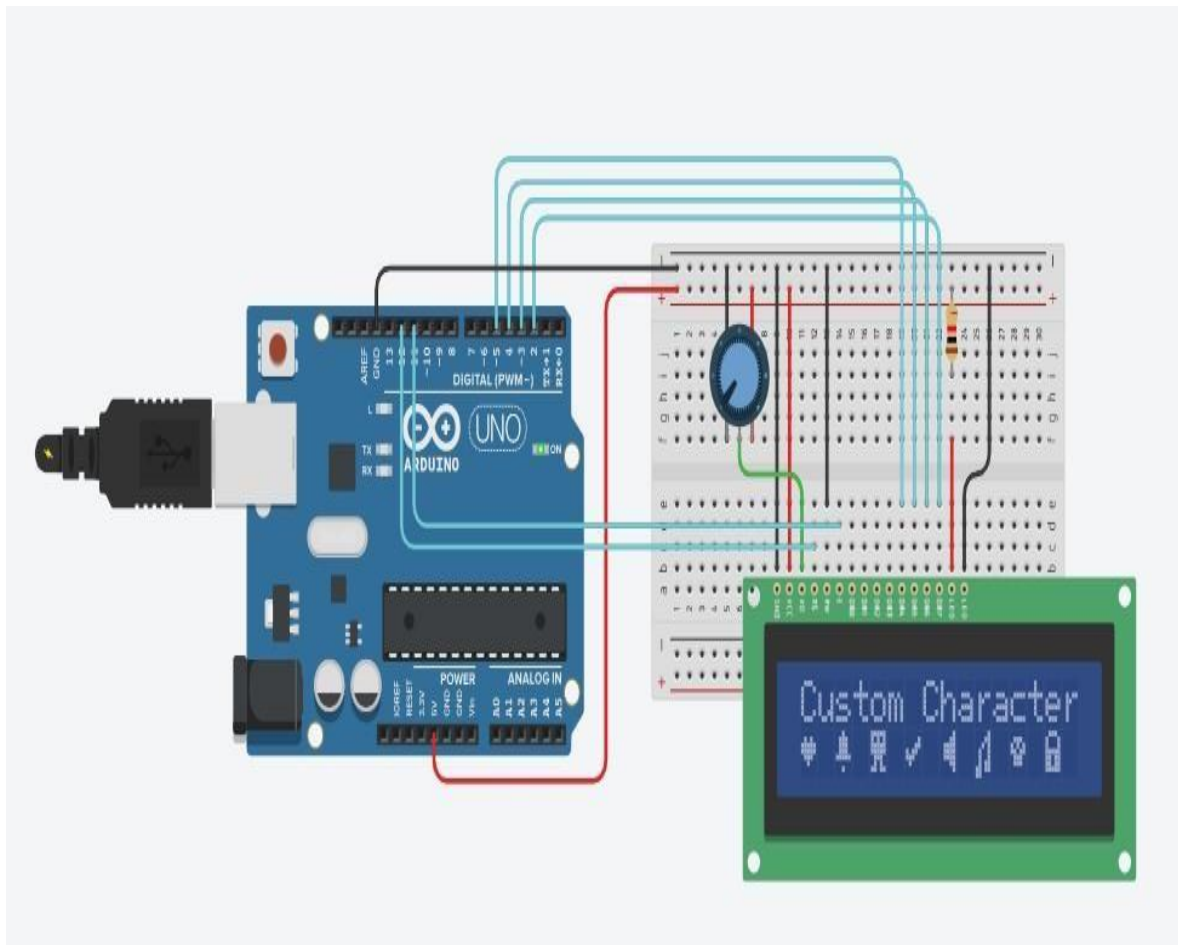
    lcd.setCursor(10, 1);
    lcd.write(byte(5));

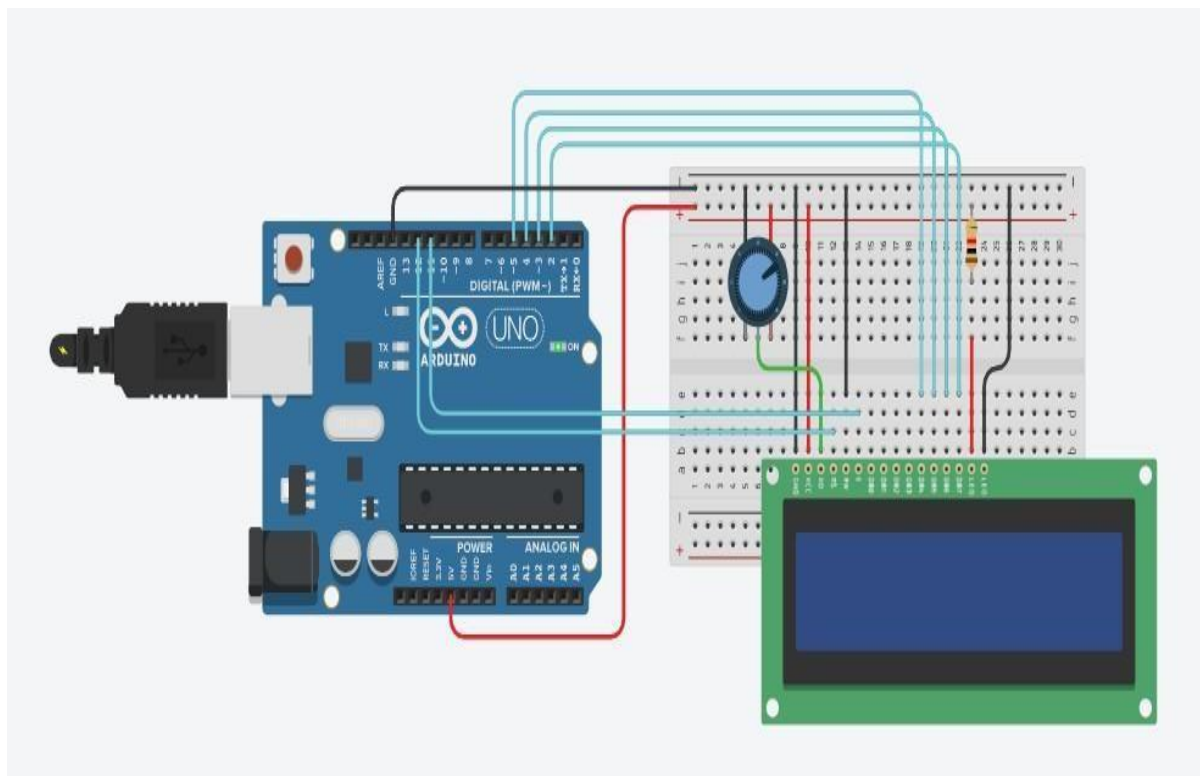
    lcd.setCursor(12, 1);
```



```
lcd.write(byte(6));  
  
lcd.setCursor(14, 1);  
lcd.write(byte(7));  
}
```

Output:





Changing the potentiometer

Conclusion:

Thus learnt about basic components of IoT like Arduino UNO(blink Arduino onboard LED), Breadboard 7segment display and interfacing 16x2 LCD display for displaying running with LCD to custom character generator with LCD controller chip with connections of resistors,breadboard,Jumper wires&Arduino.