```
/*
    The Code and Logic used behind this project is
independently designed by
                 *******   NISHANTKUMAR V PATEL    ********
 */

#include <Servo.h>                  // Different Libraries
Included.
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

int current;       // Variable Declarations.
int previous;
String inputString;
long inputInt;
String binput;
const float flowrate=25;

void ServoMotor_Rotation_FUNCTION ();         // Function
Prototypes Declarations.
void Discrete_ServoMotor_Rotation_FUNCTION ();
void IR_Sensor_FUNCTION ();
void Device_Information_FUNCTION ();
void User_Manual_FUNCTION ();

const int ROW_NUM = 4;
const int COLUMN_NUM = 4;
char keys[ROW_NUM][COLUMN_NUM] =          // different key
declarations
{
  {'1','2','3', 'A'},
  {'4','5','6', 'B'},           // it is 4x3 matrix keypad
Array.
  {'7','8','9', 'C'},
  {'*','0','#', 'D'}
};
byte row_pins[ROW_NUM] = {9, 8, 7, 6};          // to setup
the pin numbers.
```

```
byte col_pins[COLUMN_NUM] = {5, 4, 3, 2};

Keypad keypad = Keypad(makeKeymap(keys), row_pins, col_pins,
ROW_NUM, COLUMN_NUM);      // create the Keypad Object.
LiquidCrystal_I2C lcd(0x27,16,2);       // create the LCD
Display Object with I2C chip address with 16 characters and 2
rows or lines.
Servo s1;


void setup()
{
  lcd.init();  // to initialize the LCD.
  lcd.backlight();
  pinMode(13,INPUT);    // input pin for reading IR Sensor
signals.
  s1.attach(10);       // pin for sending the signals to
servo-motor.
  inputString.reserve(9);
}



void loop()    // --Main Function--           // to run
Infinitely....
{

  {
    IR_Sensor_Mode_FUNCTION ();
  }

  char key = keypad.getKey();         // pressed key accepted
as input for keypad.
  if (key)        // if any key is pressed then condition
becomes true and enter inside the below conditional
statements.
  {
    if (key >= '0' && key <= '9')
    {
      inputString += key;                   // storing the
multiple key input in a single string variable.
      lcd.setCursor(1,0);
      lcd.print("Your Input is");
      lcd.setCursor(1,1);
      lcd.print(inputString);
      lcd.setCursor(12,1);
      lcd.print("(ml)");
      delay(100);
      inputInt = inputString.toInt();           // converting
from String to Integer data type.
    }

    else if (key == '#')           // to execute the multiple
key input.
    {
      ServoMotor_Rotation_FUNCTION ();
    }

    else if (key == '*')            // to clear the multiple
key input.
    {
      inputString= "";
      inputInt= 0;
      binput="";
    }

    else if (key == 'A')        // if A is pressed then servo-
motor starts to rotate to Open the Dispenser.
    {
      lcd.clear();
      lcd.setCursor(5,0);
      lcd.print("Opened");
      for (int i=0; i<=102; i+=1)           // to initialize
the loop with conditions.
      {
        s1.write(i);        // to rotate the servo motor.
        delay(10);
      }
    }

    else if (key == 'B')          // if B is pressed then servo-motor starts to rotate discretely to Open the Dispnser partially.
    {
      binput+=key;      // store the multiple keys in a single variable- binput.
      Discrete_ServoMotor_Rotation_FUNCTION ();
    }

    else if (key == 'C')        // if C is pressed then servo-motor starts to rotate back to Close the Dispenser.
    {
      lcd.clear();            // to clear the LCD display
      lcd.setCursor(5,0);
      lcd.print("Closed");        // to print the string or character on LCD display.
      for (int i=102; i>=0; i-=1)
      {
        s1.write(i);
        delay(10);
      }
      delay(1800);
      lcd.clear();
    }

    else if (key=='D')
    {
      Device_Information_FUNCTION ();
      delay(1500);                      // delay(milliseconds)
      User_Manual_FUNCTION ();            // Every Function Prototypes are called back in main infinitely runing void loop().
    }
  }

}
```

```
  /* Function Prototyping of Different Functions for Modes.
  or Function Prototypes; which is call back in main code
structure or void loop() */

void ServoMotor_Rotation_FUNCTION ()

{
  flowrate;        // Flowrate of Actual Liquid Dispenser.
Flowrate is in (ml/second) unit.
  float time=inputInt/flowrate;     //Fluid mechanics
calculations to get the time in seconds.
  float seconds= time*1000;

  if (inputInt> 0 && inputInt!=0)
  {
    lcd.clear();
    lcd.setCursor(3,0);
    lcd.print("Started!");
    for (int i=0; i<=102; i+=1)     // for loop with initial
and ending degree angles for servo-motor.
    {
      s1.write(i);
      delay(10);
    }
    lcd.clear();
    lcd.setCursor(3,0);
    lcd.print("Running...");

    {
      delay(seconds-400);       //give the delay of amount of
seconds which is derived from the fluid mechanics
calculations.
    }

    for (int i=102; i>=0; i-=1)       //it tells to servo-
motor to rotate up to how much of angles and with speed
increments
```

```
    {
      s1.write(i);
      delay(10);
    }
    lcd.clear();
    lcd.setCursor(3,0);
    lcd.print("Completed!");
    delay(2000);
    lcd.clear();
  }
}


void Discrete_ServoMotor_Rotation_FUNCTION ()

{
  lcd.clear();
  lcd.setCursor(0,0);           //setting up the location of
the cursor in lcd display.
  lcd.print("Partial Openings");
  if (binput == "B")
  {
    lcd.setCursor(5,1);
    lcd.print("@ 1X");
    for (int i=0; i<=20; i+=1)        // if B key is pressed
one time, this loop will execute.
    {
      s1.write(i);
      delay(10);
    }
  }
  else if (binput == "BB")
  {
     lcd.setCursor(5,1);
     lcd.print("@ 2X");
     for (int i=20; i<=40; i+=1)      // if B key is pressed
consecutively two times, this loop will run.
     {
        s1.write(i);
        delay(10);
     }
  }
  else if (binput == "BBB")
  {
     lcd.setCursor(5,1);
     lcd.print("@ 3X");
     for (int i=40; i<=60; i+=1)
     {
        s1.write(i);
        delay(10);
     }
  }
  else if (binput == "BBBB")
  {
     lcd.setCursor(5,1);
     lcd.print("@ 4X");
     for (int i=60; i<=80; i+=1)
     {
        s1.write(i);
        delay(10);
     }
  }
  else if (binput == "BBBBB")       // if B key is pressed
consecutively five times, below loop will run.
  {
    lcd.setCursor(5,1);
    lcd.print("@ 5X");
    for (int i=80; i<=102; i+=1)
    {
      s1.write(i);
      delay(10);
    }
  }
}


void IR_Sensor_Mode_FUNCTION ()
 {
  current= digitalRead(13);             // 0 means, if object
is detected. 1 means, if object is not detected.
  if (current==0)            // if something is detected
(human hands for washing) then servo-motor rotate to Open the
Dispenser.
  {
    if (previous==1)
    {
      for (int i=0; i<=102; i+=1)
      {
        s1.write(i);
        delay(7);
      }
      delay(1000);
    }
  }

  else if(current==1)           // if something is not
detected then servo motor rotate back (reverse) to close the
dispenser.
  {
    if (previous==0)
    {
      for (int i=102; i>=0; i-=1)           // for
(Initialization; Condition; Updation)
      {
          s1.write(i);
          delay(7);
      }
    }
  }
  previous=current;         // assign the value of current
into previous variable for next iteration.
  delay(50);
}


void Device_Information_FUNCTION ()         // this function
```

```
is made up of entirely upon LCD codes.

{
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Hello World");
  delay(1500);
  lcd.clear();

  lcd.setCursor(4,0);
  lcd.print("I AM"); delay(1000);
  lcd.clear();
  lcd.setCursor(5,0); lcd.print("SMART");
  lcd.setCursor(0,1); lcd.print("LIQUID DISPENSER");
  delay(3500);
  lcd.clear();

  lcd.setCursor(1,0); lcd.print("Designed By");
  delay(500); lcd.setCursor(0,1);
  lcd.print("NVCK PATEL 1818");
  delay(3500);
  lcd.clear();

  lcd.setCursor(2,0); lcd.print("Save Water!");
  delay(2000);
  lcd.clear();
}

void User_Manual_FUNCTION ()        //  this function also is
made up of entirely upon LCD codes.

{
  lcd.clear(); lcd.setCursor(3,0);
  lcd.print("Welcome to"); lcd.setCursor(2,1);
  lcd.print("User Manual"); delay(3000);
  lcd.clear();

  lcd.setCursor(4,0); lcd.print("Mode-1");
  lcd.setCursor(5,1); lcd.print("Manual");
```

```
delay(3000);
lcd.clear();
lcd.setCursor(0,0); lcd.print("Press A to Open");
lcd.setCursor(0,1); lcd.print("Press C to Close");
delay(3500);
lcd.clear();

lcd.setCursor(4,0); lcd.print("Mode-2");
lcd.setCursor(0,1); lcd.print("Partial Openings");
delay(3500);
lcd.clear();
lcd.setCursor(0,0); lcd.print("Press *");
lcd.setCursor(0,1); lcd.print("to start Mode-2");
delay(3500);
lcd.clear(); lcd.setCursor(0,0);
lcd.print("Press B to set"); lcd.setCursor(0,1);
lcd.print("openings upto 5X"); delay(3500);
lcd.clear();

lcd.setCursor(4,0); lcd.print("Mode-3");
lcd.setCursor(3,1); lcd.print("Auto-Stop");
delay(3500);
lcd.clear();
lcd.setCursor(0,0); lcd.print("Press 0 to 9 keys");
lcd.setCursor(0,1); lcd.print("as Input in ml");
delay(3500);
lcd.clear();
lcd.setCursor(0,0); lcd.print("Press *");
lcd.setCursor(2,1); lcd.print("to clear Input");
delay(3500);
lcd.clear();
lcd.setCursor(0,0); lcd.print("Press #");
lcd.setCursor(2,1); lcd.print("to start Input");
delay(3500);
lcd.clear();

lcd.setCursor(4,0); lcd.print("Mode-4");
lcd.setCursor(0,1); lcd.print("Sensor/ Default");
delay(3500);
```

```
  lcd.clear();
}
```