

# IBM PROJECT – REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

## ASSIGNMENT 1

CODE:

```
#include <LiquidCrystal.h>
```

```
// initialize the library with the numbers of the interface pins
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
//For ultrasound sensor
```

```
int distanceThreshold = 0;
```

```
int cm = 0;
```

```
int inches = 0;
```

```
//for Relay Control
```

```
int releNO = 13;
```

```
int inputPir = 8;
```

```
int val = 0;
```

```
int resuldoSensorLDR;
```

```
int sensorLDR = A0;
```

```
//For Gas sensor
```

```
int const PINO_SGAS = A1;
```

```
long readUltrasonicDistance(int triggerPin, int echoPin)
```

```

{
  pinMode(triggerPin, OUTPUT); // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

```

```

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);

  pinMode(releNO, OUTPUT);
  pinMode(inputPir, INPUT);
  pinMode(sensorLDR, INPUT);
  Serial.begin(9600);
}

```

```

void loop() {
  // set threshold distance to activate LEDs
  distanceThreshold = 350;
  // measure the ping time in cm

```

```
cm = 0.01723 * readUltrasonicDistance(7, 6);
```

```
// convert to inches by dividing by 2.54
```

```
inches = (cm / 2.54);
```

```
    lcd.setCursor(0,0); // Sets the location at which subsequent text written to the  
LCD will be displayed
```

```
    lcd.print("D:"); // Prints string "Distance" on the LCD
```

```
    lcd.print(cm); // Prints the distance value from the sensor
```

```
    lcd.print("cm");
```

```
    delay(10);
```

```
    val = digitalRead(inputPir);
```

```
    resuldoSensorLDR = analogRead(sensorLDR);
```

```
    if(resuldoSensorLDR<600)
```

```
{
```

```
    if(val == HIGH)
```

```
{
```

```
    digitalWrite(releNO, HIGH);
```

```
    lcd.setCursor(0,1);
```

```
    lcd.print("L: On ");
```

```
    delay(5000);
```

```
}
```

```
else{
```

```
    digitalWrite(releNO, LOW);lcd.setCursor(0,1);
```

```
    lcd.print("L: Off");
```

```
    delay(300);
```

```
}
```

```
}
```

```
else{ digitalWrite (releNO, LOW);  
Serial.println(resuldoSensorLDR);  
delay(500);  
}
```

```
int color = analogRead(PINO_SGAS);
```

```
lcd.setCursor(8,0);  
//lcd.print("");  
if(color <= 85){  
    lcd.print("G:Low ");  
} else if(color <= 120){  
    lcd.print("G:Med ");  
} else if(color <= 200){  
    lcd.print("G:High");  
} else if(color <= 300){  
    lcd.print("G:Ext ");  
}
```

```
delay(250);  
}
```

TINKERCAD LINK:

<https://www.tinkercad.com/things/jEHx7f50XjP-smooth-turing-kasi/editel?sharecode=Cri9RyyG4luVKOQNxX94uLg3Ecmk8--OY0vmxlzMpzg>

## CIRCUIT DESIGN:

