Coding and Solution

Team ID	PNT2022TMID47935
Project Name	Real-time river water quality monitoring and control system

Utilization Of Algorithms

1. We are collecting the data from the sensor nodes.
2. We have to setup the IBM cloud connection configuration in Node-RED platform
3. Then it can connect the IBM Watson IoT with Node-RED platform
4. Then the data are transferred to IBM Watson IoT platform
5. We have to design and develop the app for our needed works.
6. And connect the app with Node-RED
7. So, it can easily show the real time water's pH and Turbidity values in our mobile app
8. If we want to close the particular dam, we needed motor controller.
9. So, we made a motor controller in our own mobile app.
10. The controller's results are shown in Node-RED
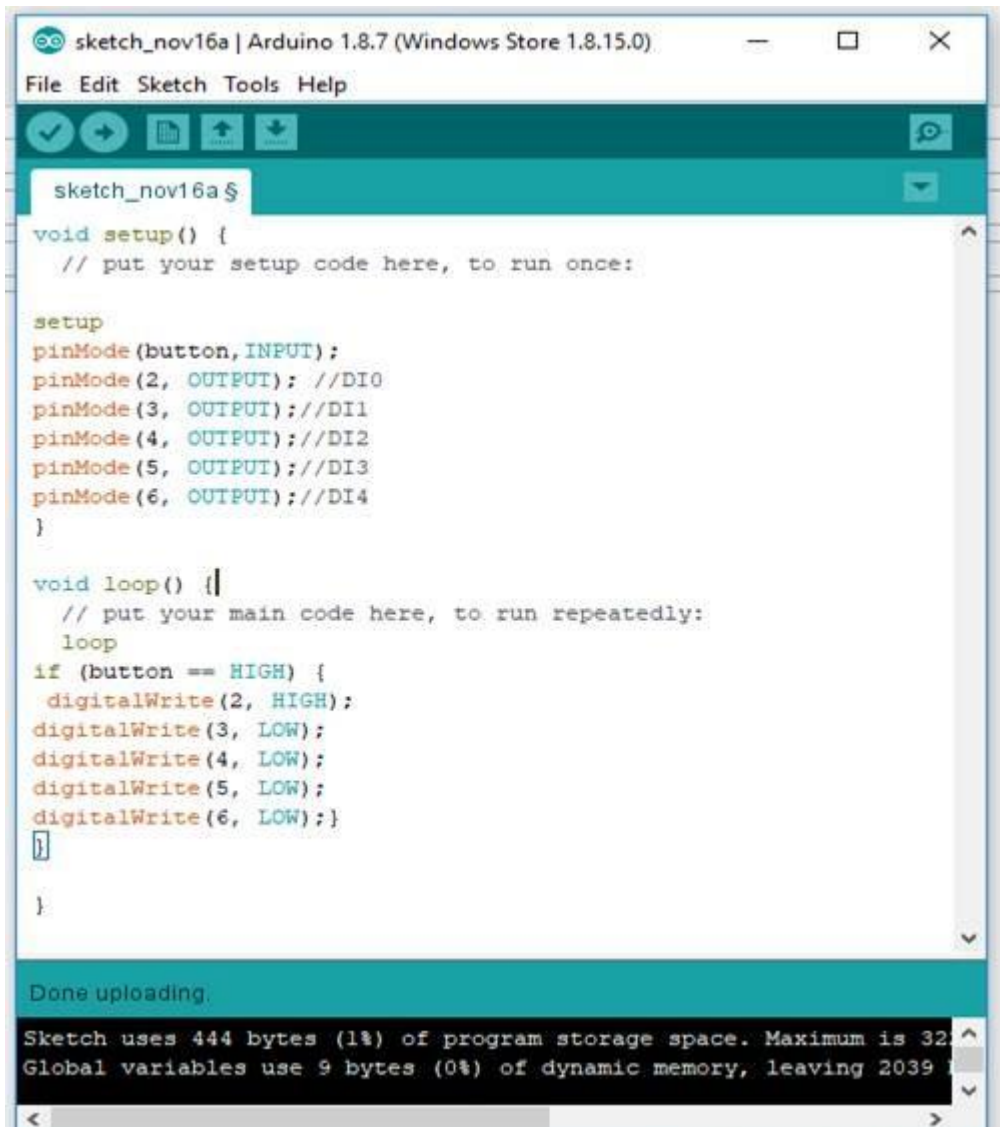
Dynamic Program

```
void setup()
{
  Serial.begin(115200);
  pinMode(led, OUTPUT);
  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  wifiConnect();
  mqttConnect();
}

void loop() {
  bool isNearby = dist < 100;
  digitalWrite(led, isNearby);

  publishData();
  delay(500);

  if (!client.loop()) {
    mqttConnect();
```

```
      }
    }
```

```
void setup() {
  // put your setup code here, to run once:

  setup
  pinMode(button, INPUT);
  pinMode(2, OUTPUT); //DI0
  pinMode(3, OUTPUT); //DI1
  pinMode(4, OUTPUT); //DI2
  pinMode(5, OUTPUT); //DI3
  pinMode(6, OUTPUT); //DI4
  }

void loop() {
  // put your main code here, to run repeatedly:
  loop
  if (button == HIGH) {
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);}
  }

  }
```

Done uploading.

```
Sketch uses 444 bytes (1%) of program storage space. Maximum is 32
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039
```

Optimisation

```
void mqttConnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting MQTT client to "); Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
```

```arduino
  }
}

void initManagedDevice() {
  if (client.subscribe(topic)) {
    // Serial.println(client.subscribe(topic));
    Serial.println("IBM subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}
void publishData()
{
  digitalWrite(trigpin,LOW);
  digitalWrite(trigpin,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin,LOW);
  duration=pulseIn(echopin,HIGH);
  dist=duration*speed/2;
  if(dist<100){
    String payload = "{\"Alert Distance is\":";
    payload += dist;
    payload += "}";

    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);
      if(client.publish(publishTopic, (char*) payload.c_str())) {
      Serial.println("Warning crosses 110cm -- it automaticaly of the loop");
      digitalWrite(led,HIGH);
    }

  }
    if(dist>101 && dist<111){
    String payload = "{\"Normal Distance\":";
    payload += dist;
    payload += "}";

    Serial.print("\n");
    Serial.print("Sending payload: ");
    Serial.println(payload);

    }
```

}