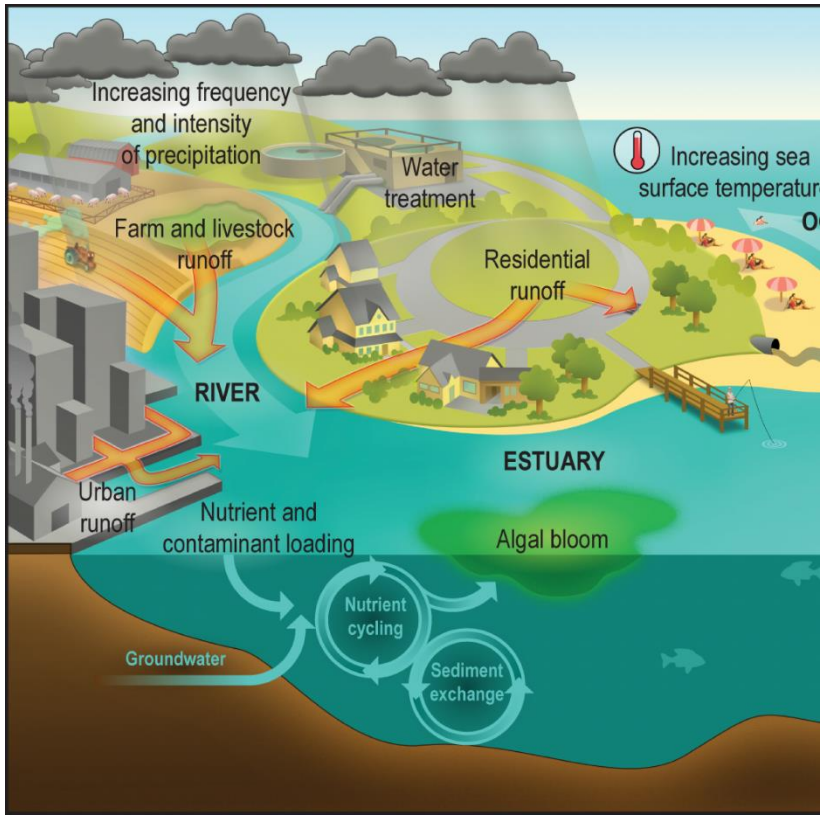
IOT BASED MOBILE (NAVIGATING IN WATER) WATER POLLUTION MONITORING SYSTEM

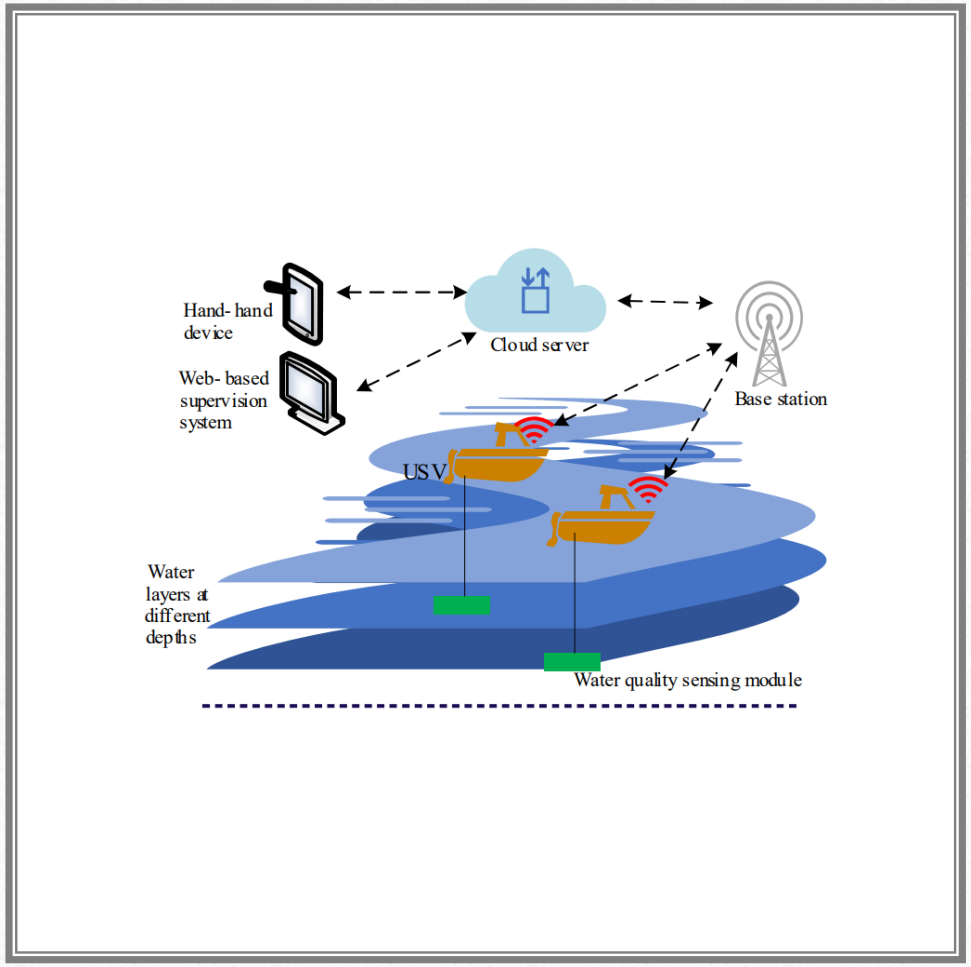
VINEESCAR.V



INTRODUCTION

- Water quality is one of the main factors to control health and the state of diseases in people and animals.

- Traditionally, detection of water quality was manually performed where water samples were obtained and sent for examination to the laboratories which is time taking process, cost and human resources.



- Such techniques do not provide data in real-time. The proposed water quality monitoring system is consisting of a microcontroller and basic sensors, is compact and is very useful for pH, turbidity, water level detection, temperature and humidity of the atmosphere, continuous and real-time data sending via wireless technology to the monitoring station.

OBJECTIVE

Building an unmanned boat with manual navigation and sensors to measure the parameters to determine the quality of water and logging it for real-time data acquisition regarding the water system.

LITERATURE REVIEW

EVALUATION

FIELD RECONSTRUCTION

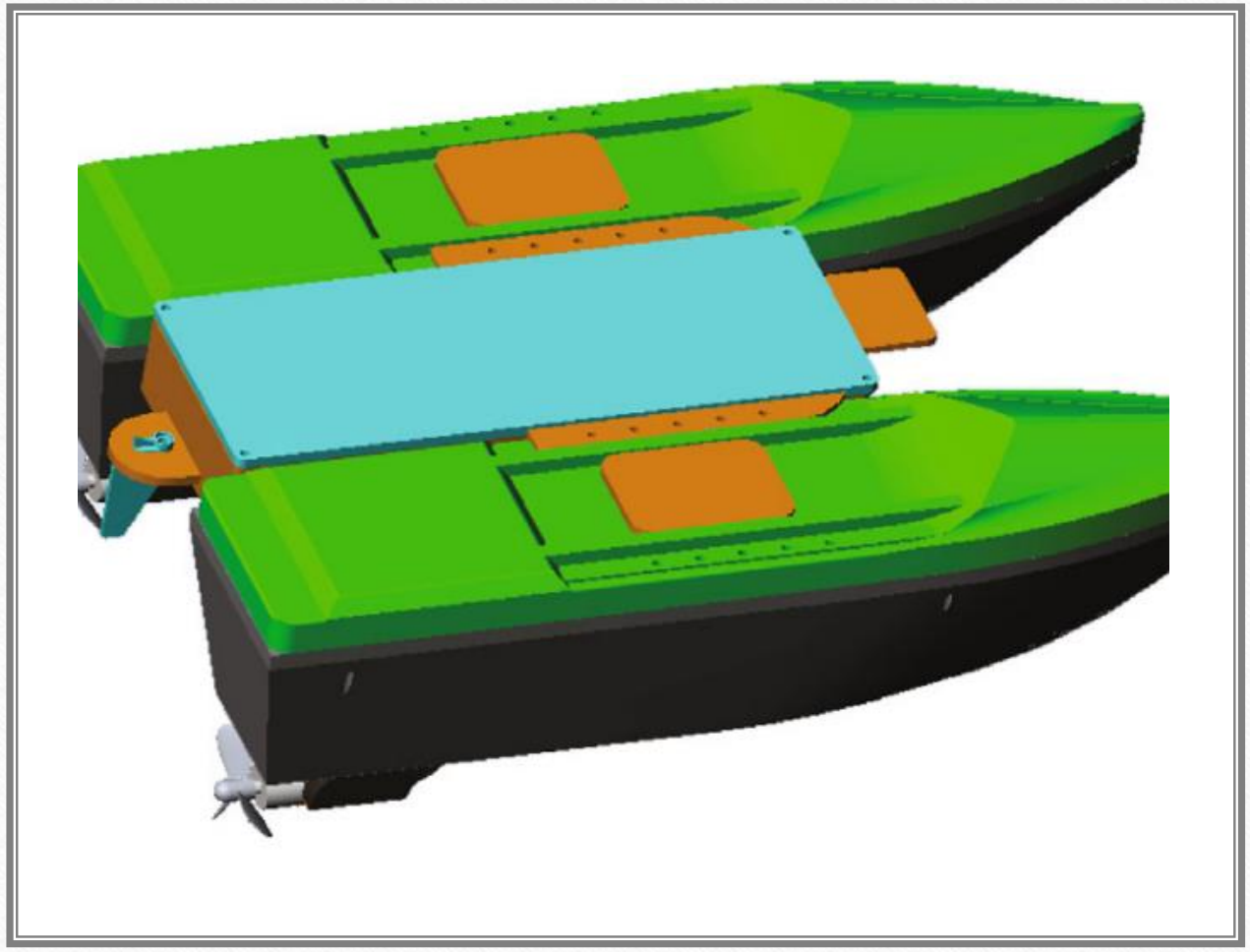
DEVELOPMENT

SENSOR DEPLOYMENT

PATH PLANNING

ENVIRONMENT MODELING

DATA INTERPRETATION



RESEARCH GAP

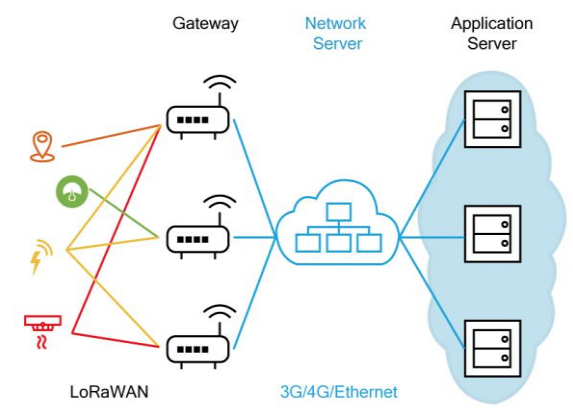
1)Boat design,

In many papers they designed their boat using plastic and regiform.

Here we are going to design our boat using fiber. And we changed our shape of unmanned boat for more stability.

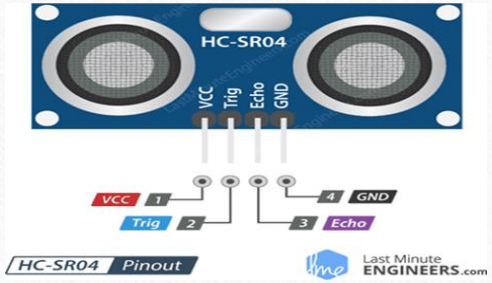
2)Network,

Most of the researches are based on small area so they used ZigBee and WIFI module for their connectivity. Here we are using lora network with long range and low power consumption.



- 3)Sensors,

We are going to use sonar sensor to detect the clear path of the boat movement.



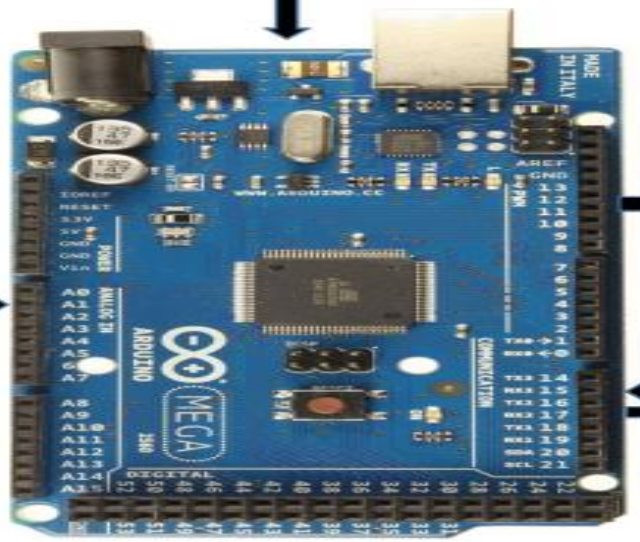
4)Energy,

We planned to add a solar panel for long life of the boat.

METHODOLOGY

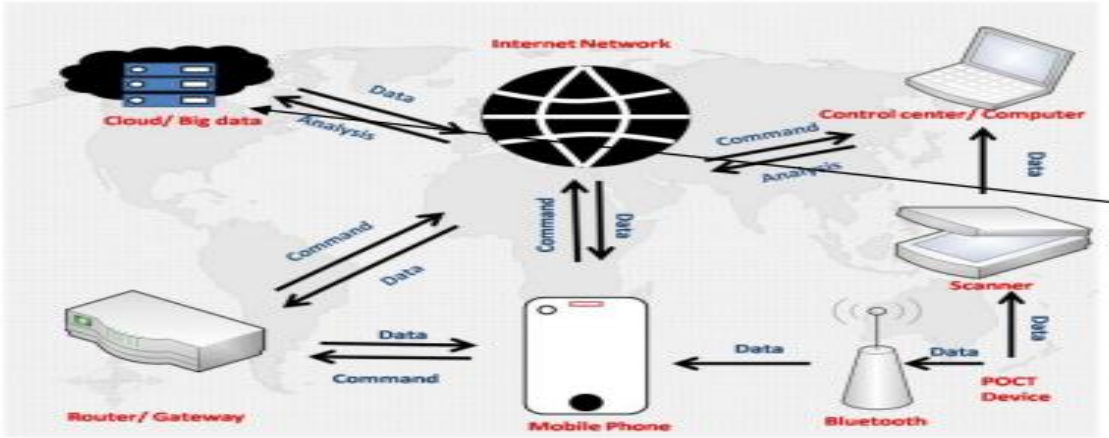
POWER SUPPLY MODULE (BATTERY + SOLAR PANAL)

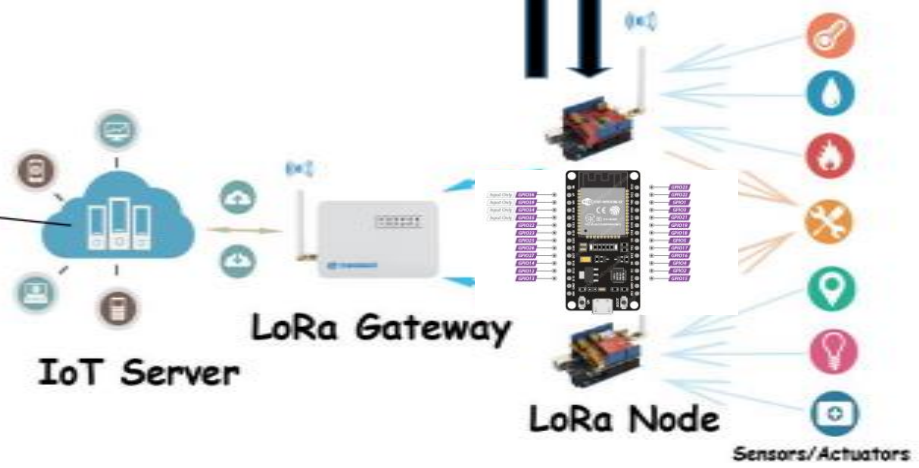
1)WATER QUALITY SAMPLING MODULE	TEMPERATURE
	PH
	CONDUCTIVITY
	DISSOLVED OXYGEN
	TURBIDITY
2)GPS POSITIONING MODULE	
3)SONNAR SENSOR	



MOTION MODULE	MOTOR CONTROL
PUMP CONTROL MODULE	

LORA MODULE



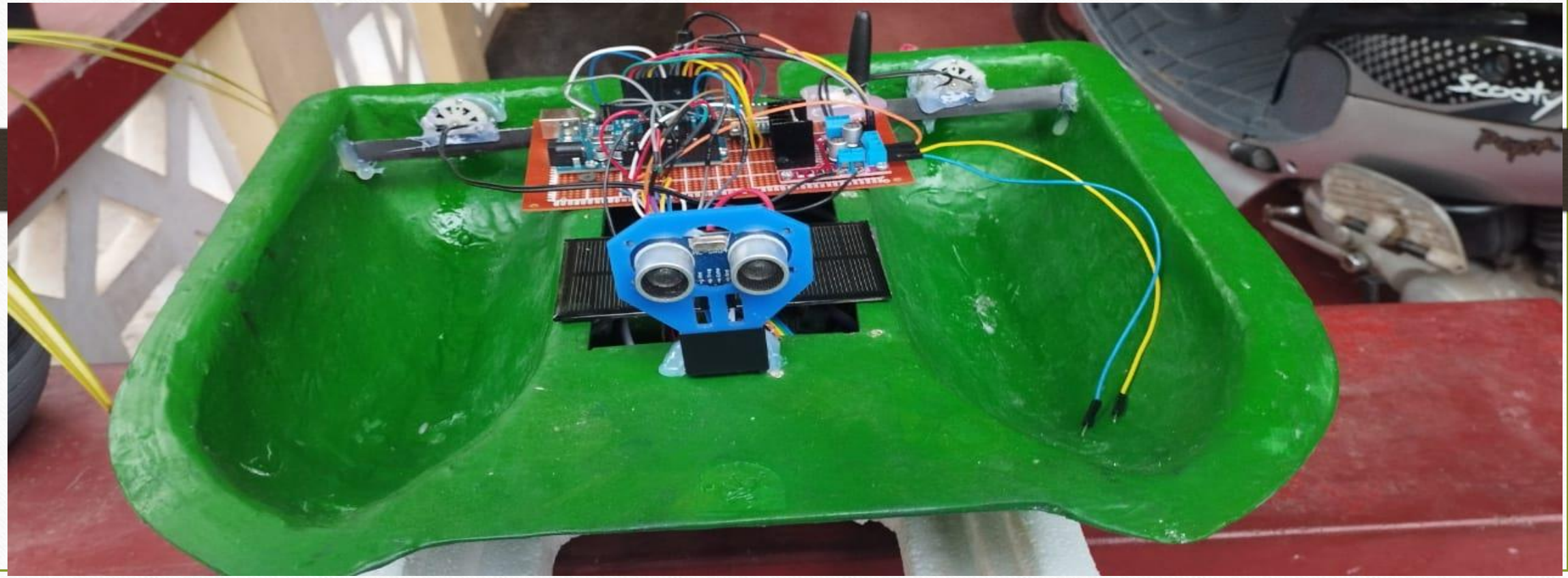


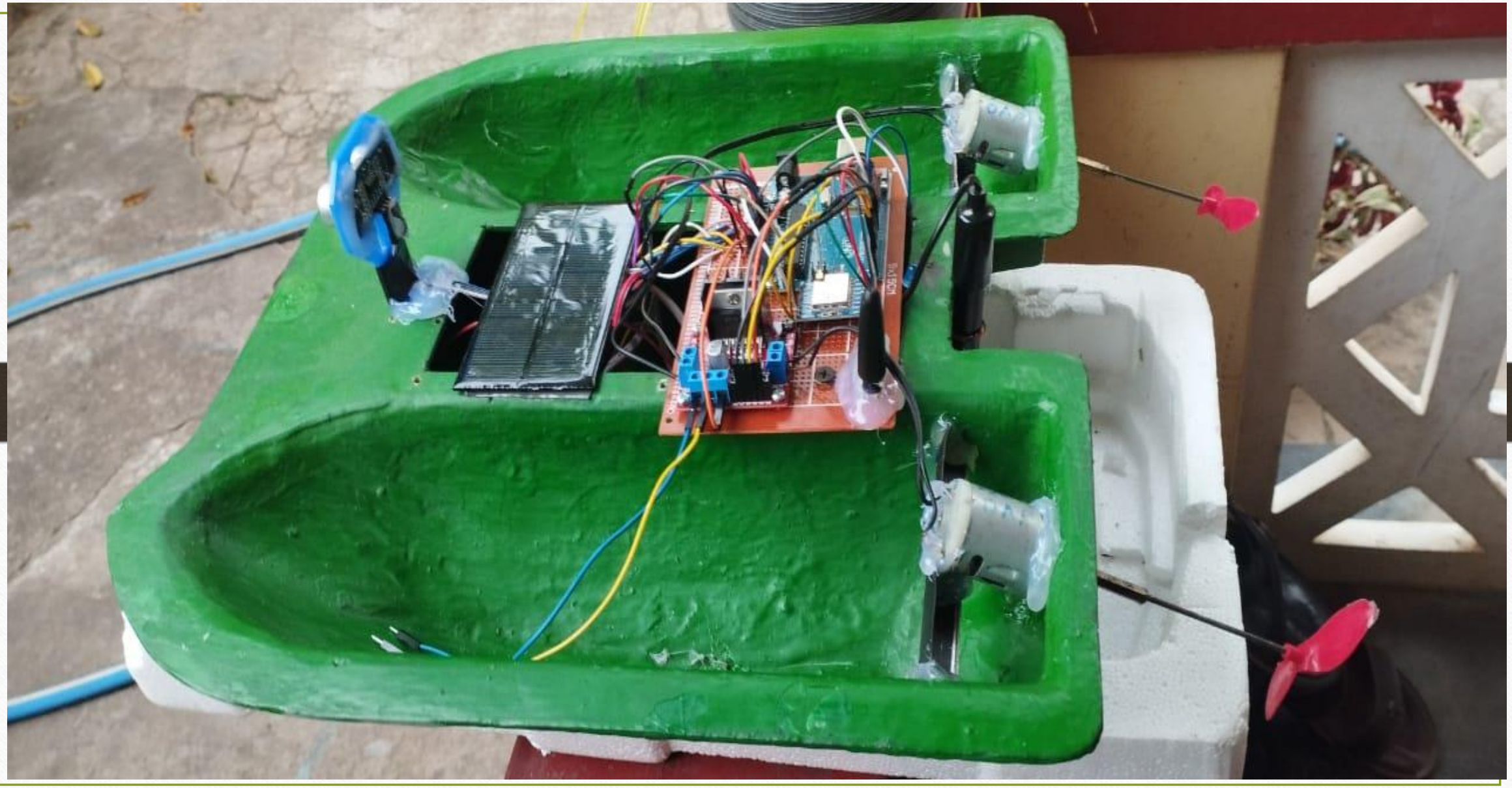
Motor controller(L298N) is shown below,

- In this project I used Arduino uno and ESP32 modules.
- Arduino uno is with the boat which is responsible for collecting the sensor data and communicate with esp32 through the lora module.
- ESP32 is responsible to connect with the blynk application to control the motor.
- Here I used lora SX1278 which one module is connected in esp32 and another one in Arduino uno .
- So only the esp32 need internet connection. We can control the boat about 10km range from esp32.
- I am providing the code which is coded in esp32 and Arduino uno.
- At lora module at a time one way communication will happen synchronize the communication is the most difficult part.

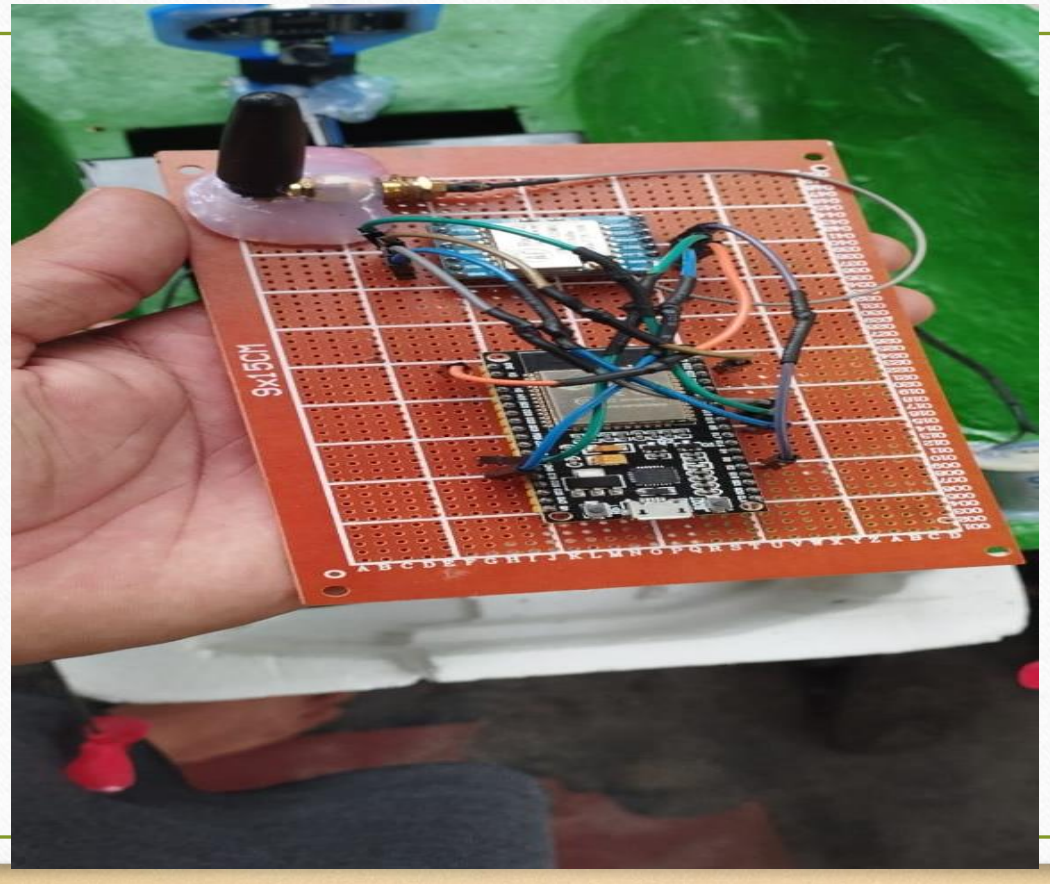
BLYNK IOT INTERFACE

Boat with Arduino uno which is connected with lora





ESP32 which is connected with lora amd blynk IOT



ESP32 code

```
#define BLYNK_PRINT Serial

#define BLYNK_TEMPLATE_ID "TMPL6cWTFiXrJ"
#define BLYNK_TEMPLATE_NAME "LoRaBoat"
#define BLYNK_AUTH_TOKEN "QTQedVDCVt8xu5L6ilrX7Jj4b4F26Mg8"


#include <BlynkSimpleEsp32.h>
#include <WiFi.h>
#include <WiFiClient.h>

char auth[] = BLYNK_AUTH_TOKEN;

char ssid[] = "Vinu 4G";
char pass[] = "AQYD3LYE3RY";


#include <SPI.h>
#include <LoRa.h>

#define SCK 18
#define MISO 23
#define MOSI 19
#define SS 5
#define RST 14
#define DIO0 2

int counter = 0;
int sender=0;
int moveF , moveB , moveL , moveR,collect ; // Variables for movement
```

```
int move;

int tds,turbidity,phvalue,temperature,distance;

BLYNK_WRITE(V0)                 // Front Movement
{
  moveF = param.asInt(); // assigning incoming value from pin V1 to a variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  Serial.print("MoveF value is: ");
  Serial.println(moveF);
}

BLYNK_WRITE(V1)                 // left Movement
{
  moveL = param.asInt(); // assigning incoming value from pin V1 to a variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  Serial.print("MoveL value is: ");
  Serial.println(moveL);
}

BLYNK_WRITE(V2)                 // right Movement
{
  moveR = param.asInt(); // assigning incoming value from pin V1 to a variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
```

```
  Serial.print("MoveR value is: ");
  Serial.println(moveR);
}

BLYNK_WRITE(V3)                 // back Movement
{
  moveB = param.asInt(); // assigning incoming value from pin V1 to a variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  Serial.print("MoveB value is: ");
  Serial.println(moveB);
}
BLYNK_WRITE(V4)                 // Front Movement
{
  collect = param.asInt(); // assigning incoming value from pin V1 to a variable
  // You can also use:
  // String i = param.asStr();
  // double d = param.asDouble();
  Serial.print("collect value is: ");
  Serial.println(collect);
}


void setup() {
  Serial.begin(9600);
  while (!Serial);

  Blynk.begin(auth, ssid, pass);

```

```
  Serial.println("LoRa Duplex with ESP32");

  SPI.begin(SCK, MISO, MOSI, SS);
  LoRa.setPins(SS, RST, DIO0);

  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa init failed. Check your connections.");
    while (true);
    Serial.println("LoRa Duplex init success");
  }
}

void loop() {
  if (collect !=100)
  {
  Serial.print("MoveF value is: ");
  Serial.println(moveF);

  Serial.print("MoveL value is: ");
  Serial.println(moveL);

  Serial.print("MoveR value is: ");
  Serial.println(moveR);

  Serial.print("MoveB value is: ");
  Serial.println(moveB);

  Serial.print("collect value is: ");
  Serial.println(collect);

```

```
121     Blynk.virtualWrite(V11, 300);
122     Serial.println("wrighting testing");
123    if(moveF>30 & moveR<30 & moveL<30 & moveB<30){
124      move = 10;}
125    else if(moveF<30 & moveR>30 & moveL<30 & moveB<30){
126      move = 20;}
127    else if(moveF<30 & moveR<30 & moveL>30 & moveB<30){
128      move=30;}
129    else if(moveF<30 & moveR<30 & moveL<30 & moveB>30){
130      move = 40;}
131
132    else{
133      move=0;
134    }
135
136    Serial.println("Sending packets ");
137   // Serial.println(counter);
138
139    LoRa.beginPacket();
140    LoRa.write(collect);
141    LoRa.write(move);
142    LoRa.print("Hello from esp32 ,");
143    LoRa.print(counter);
144    LoRa.endPacket();
145    delay(2000);
146    //counter++;
147    }
148
149  else if(collect==100)
150  {
```

```
151   int packetSize = LoRa.parsePacket();
152   if (packetSize) {
153     Serial.print("Receiving : ");
154     tds=LoRa.read();
155     turbidity=LoRa.read();
156     phvalue=LoRa.read();
157     temperature=LoRa.read();
158     distance=LoRa.read();
159     while (LoRa.available()) {
160       Serial.print((char)LoRa.read());
161     }
162     Serial.println();
163    // Serial.println(sender);
164     Blynk.virtualWrite(V6, tds);
165     Blynk.virtualWrite(V5, phvalue);
166     Blynk.virtualWrite(V7, turbidity);
167     Blynk.virtualWrite(V8, distance);
168     Blynk.virtualWrite(V9, temperature);
169     Serial.println("wrighting sensor datas");
170     }
171
172 }
173
174 }
175
```

Arduno uno code

```
#include <SPI.h>
#include <LoRa.h>

const int ledPin = 7;

#define SS_PIN 10
#define RST_PIN 9
#define DI0_PIN 2

//sensor start
#include <Wire.h>

#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 8
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

#define TURBIDITY_PIN A0//turbidity sensor pin
#define SENSOR_PIN A1  //conductivity sensor
#define ONE_WIRE_BUS 8 //  temperature sensor

float calibration_value = 21.34 - 0.7;  //ph
int phval = A5;
unsigned long int avgval;
int buffer_arr[10],tempk;


#define trigPin A3
#define echoPin A4
```

```
long duration, distance; // Variables for saving the values from ultrasonic sensor
//sensor end

const int temp = 8;  //temperature sensor

// Motor A connections
const int in3 = 3;
const int in4 = 4;

const int enA = 7;
const int in1 = 6;
const int in2 = 5;

int counter = 0;
int collectU=0;
int moveM=0;

void setup() {

  Serial.begin(9600);
  sensors.begin(); //for temperature sensor
  while (!Serial);

  pinMode(ledPin, OUTPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  // Set the motor control pins as outputs
  pinMode(enA, OUTPUT);
```

```
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);


  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  pinMode(temp, INPUT);

  pinMode(phval, INPUT);


  Serial.println("LoRa Duplex with Arduino Uno");

  LoRa.setPins(SS_PIN, RST_PIN, DI0_PIN);

  if (!LoRa.begin(433E6)) {
    Serial.println("LoRa init failed. Check your connections.");
    while (true);
  }
  Serial.println("LoRa Duplex init success");
}

void loop() {

digitalWrite(ledPin, HIGH);

if (collectU != 100)
{

```

```
 91    if (moveM == 10) {
 92      digitalWrite(in1, HIGH);
 93      digitalWrite(in2, LOW);
 94      digitalWrite(in3, HIGH);
 95      digitalWrite(in4, LOW);
 96      Serial.println("Direction: Forward");
 97    } else if(moveM==20) {
 98      digitalWrite(in1, HIGH);
 99      digitalWrite(in2, LOW);
100      digitalWrite(in3, LOW);
101      digitalWrite(in4, LOW);
102      Serial.println("Direction: right");
103    } else if(moveM==30) {
104      digitalWrite(in1, LOW);
105      digitalWrite(in2, LOW);
106      digitalWrite(in3, HIGH);
107      digitalWrite(in4, LOW);
108      Serial.println("Direction: left");
109    }else if(moveM==40) {
110      digitalWrite(in1, LOW);
111      digitalWrite(in2, HIGH);
112      digitalWrite(in3, LOW);
113      digitalWrite(in4, HIGH);
114      Serial.println("Direction: backward");
115    }else  {
116      digitalWrite(in1, LOW);
117      digitalWrite(in2, LOW);
118      digitalWrite(in3, LOW);
119      digitalWrite(in4, LOW);
120      Serial.println("Direction: no movement");
```

```
    }


  int packetSize = LoRa.parsePacket();
  if (packetSize) {
  Serial.print("Received packet: ");
  collectU = LoRa.read();
  moveM = LoRa.read();
  while (LoRa.available()) {
    Serial.print((char)LoRa.read());
  }
  Serial.println();

  Serial.print("collectu no is :");
  Serial.println(collectU);

  Serial.print("motor control no is ");
  Serial.println(moveM);
  }

}

else if(collectU==100)
{
    digitalWrite(ledPin, LOW);

    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
```

```
151     Serial.println("Direction: no movement");
152
153     delay(2000);
154
155         for (int i = 0; i < 10; i++){
156             int sensorValue1 = analogRead(SENSOR_PIN);        //conductivity start
157             float tdsValue = map(sensorValue1, 0, 1023, 0, 1000);  // Replace with your conversion function
158             Serial.print("Conductivity: ");
159             Serial.print(tdsValue);
160             Serial.println(" ppm ");                          //conductivity end
161
162             int turbidityValue = analogRead(TURBIDITY_PIN);          //Turbidity start
163             float turbidity = map(turbidityValue, 600, 0, 0, 100);
164             Serial.print("Turbidity: ");
165             Serial.print(turbidity);
166             Serial.println("%");                                     //turbidity end
167
168             for(int i=0;i<10;i++)                                  //ph start
169             {
170             buffer_arr[i]=analogRead(A0);
171             delay(30);
172             }
173             for(int i=0;i<9;i++)
174             {
175             for(int j=i+1;j<10;j++)
176             {
177             if(buffer_arr[i]>buffer_arr[j])
178             {
179             tempk=buffer_arr[i];
180             buffer_arr[i]=buffer_arr[j];
```

```
181                 buffer_arr[j]=tempk;
182                 }
183                 }
184                 }
185                 avgval=0;
186                 for(int i=2;i<8;i++)
187                 avgval+=buffer_arr[i];
188                 float volt=(float)avgval*5.0/1024/6;
189                 float ph_act = -5.70 * volt + calibration_value;
190                 Serial.print("pH Val: ");
191                 Serial.println(ph_act);                                    //ph end
192
193                 Serial.print("Requesting temperatures...");                 //temperatutre start
194                 sensors.requestTemperatures(); // Send the command to get temperatures
195                 Serial.println("DONE");
196                 float tempC = sensors.getTempCByIndex(0);
197                 if(tempC != DEVICE_DISCONNECTED_C)
198                 {
199                   Serial.print("Temperature for the device is: ");
200                   Serial.println(tempC);
201                 }
202                 else
203                 {
204                   Serial.println("Error: Could not read temperature data");
205                 }                                                                  //temperature end
206
207                 digitalWrite(trigPin, LOW);           //ultrasonic sensor
208                 delayMicroseconds(2);
209                 digitalWrite(trigPin, HIGH);     // send waves for 10 us
210                 delayMicroseconds(10);
```

```
211                 duration = pulseIn(echoPin, HIGH); // receive reflected waves
212                 distance = duration / 58.2;        // convert to distance
213                 Serial.print("distance: ");
214                 Serial.print(distance);
215                 Serial.println("cm");
216
217                 Serial.print("Sending packet: ");
218                 Serial.println(counter);
219
220                 LoRa.beginPacket();
221                 LoRa.write(tdsValue);
222                 LoRa.write(turbidity);
223                 LoRa.write(ph_act);
224                 LoRa.write(tempC);
225                 LoRa.write(distance);
226                 LoRa.print("Hello from Arduino Uno ,");
227                 LoRa.print(counter);
228                 LoRa.endPacket();
229
230                 delay(2000);
231
232
233 }
234
235
236 collectU = 0 ;
237 }
238
239 }
```

REFERENCES

[1] Mohamed Aslam, Sreerag K, Stebin T Jose, Mr. G. Chandrashekar(2022). IoT Based Water Pollution Monitoring RC Boat. (IJARSCT) Dhanalakshmi Srinivasan Engineering College (Autonomous),Perambalur ISSN (Online) 2581-9429.

[2] S. Deng, J. Yang, Y. Wu, J. Lu, 'Research on small mobile water quality online monitoring device', *2013 10th IEEE International Conference on Control and Automation (ICCA)*, 2013, 351–356.

[3] T. Li, M. Xia, J. Chen, Y. Zhao, C. De Silva, 'Automated Water Quality Survey and Evaluation Using an IoT Platform with Mobile Sensor Nodes', *Sensors*, . 17, . 8, 2017.

[4] S. Pasika S. T. Gandla, 'Smart water quality monitoring system with cost-effective using IoT', *Heliyon*, . 6, .7, . e04096, 2020.

[5Jayti Bhatt, Jignesh Patoliya, Iot Based Water Quality Monitoring System, IRFIC, 21feb,2016.

[6Nikhil Kedia, Water Quality Monitoring for Rural Areas- A Sensor Cloud Based Economical Project, in 1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India, 4-5 September 2015. 978-1-4673-6809-4/15/$31.00 ©2015 IEEE

[7] Malljunath BS, Ohm JR, Vasuden Van VV. Color and texture descriptors. IEEE Trans. On CSVT, 2001, 11 (6): 703-715.

[8] A.N.Prasad, K. Mamun, F. Islam, H. Haqva, 'Smart Water Quality Monitoring System', 12 2015..

[9] T. Lambrou, C. Anastasiou, C. Panayiotou, 'A Nephelometric Turbidity System for Monitoring Residential Drinking Water Quality', 09 2009, . 29, . 43–55.

[10] M. O. Faruq, I. H. Emu, M. N. Haque, M. Dey, N. K. Das, M. Dey, 'Design and implementation of cost effective water quality evaluation system', *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2017, . 860–863.

THANK YOU