**idea.ino**

```
#include<Arduino.h>
#include <SoftwareSerial.h>
#include<NewPing.h>

SoftwareSerial BTserial(7, 6);  // RX | TX

NewPing ultra(9,10); //t && e

String MODE = "AUTOMATIC";
String MOTOR_STATUS = "OFF";
int timmer = 0;
int val;


int sending_timmer =0;
bool error_status = false ;

bool start = true;

void setup()
{


  BTserial.begin(9600);
  BTserial.setTimeout (200);
  Serial.begin(9600);

  pinMode( 13 , OUTPUT); //MOTOR

  pinMode( 2 , OUTPUT );  // ERROR
  pinMode( 3 , OUTPUT ); //MOTOR GOOD STATUS

  pinMode( 4 , INPUT ); // ERROR REST BUTTON

  pinMode( 11 , INPUT ); // MANUAL MOTOR ON
  pinMode( 12 , INPUT ); // MANUAL MOTOR OFF


 // OUTPUT DISPLAY
  pinMode( A0 , OUTPUT );
  pinMode( A1 , OUTPUT );
  pinMode( A2 , OUTPUT );
  pinMode (A3 , OUTPUT );




  // MANUAL AND AUTOMATIC INDICATOR
  pinMode(A4 , OUTPUT);
  pinMode(A5 , OUTPUT);

  // MANUAL AND AUTOMATIC ALTER
  // pinMode( 5 , INPUT );

  pinMode(8 , OUTPUT); // MOTOR


```

```
58
59
60
61
62  }
63
64  void loop()
65  {
66
67
68    if(start)
69    {
70      digitalWrite(3 , HIGH);
71      digitalWrite(2 , LOW);
72      start = false;
73
74      digitalWrite(A0 , LOW);
75      digitalWrite(A1 , LOW);
76      digitalWrite(A2 , LOW);
77      digitalWrite(A3 , LOW);
78
79      digitalWrite(8 , HIGH);
80
81      digitalWrite( 13 , LOW);
82
83    }
84
85
86
87   // DATA FROM ULTRASONIC
88   int dur = ultra.ping_median();
89   int distance = ultra.convert_in(dur);
90   Serial.print(distance);
91
92   Serial.print("  ");
93   Serial.print(timmer);
94   Serial.print("  ");
95   Serial.print(MODE);
96   Serial.print("  ");
97   Serial.println(sending_timmer);
98
99
100
101  // GETTING DATA FROM BLUETOOTH
102  if (BTserial.available()>2)
103   {
104     String str = BTserial.readString();
105     Serial.print(str);
106
107     if( str == "SETMANUAL\r\n" )
108     {
109       MODE = "MANUAL";
110     }
111
112     if( str == "SETAUTOMATIC\r\n"  )
113     {
114       MODE = "AUTOMATIC";
115     }
116
117     if( str ==  "RESETPUSH\r\n" )
```

```
    {
      error_status = false;
      digitalWrite( 2 , LOW); // ERROR OFF
      digitalWrite( 3 , HIGH); // MOTOR GOOD ON
    }

    // MANUAL CONTROL IN MOBILE
    if( str == "SETMOTORON\r\n" && !error_status &&  MODE == "MANUAL" )
    {
      digitalWrite(13 , HIGH); //MOTOR ON
      digitalWrite(8 , LOW); // MOTOR
      MOTOR_STATUS = "ON";
    }

    if( str == "SETMOTOROFF\r\n" && !error_status &&  MODE == "MANUAL" )
    {
      digitalWrite( 13 , LOW); // MOTOR OFF
      digitalWrite(8 , HIGH); //MOTOR
      MOTOR_STATUS = "OFF";
    }

  }
  // CONDITIONS IF AUTOMATIC
  if(MODE == "AUTOMATIC" && !error_status )
  {
   if( distance >= 5 )
   {
     digitalWrite(13 , HIGH); //MOTOR ON
     digitalWrite(8 , LOW); // MOTOR
     MOTOR_STATUS = "ON";
   }

   if( distance <=1 )
   {
     digitalWrite( 13 , LOW); // MOTOR OFF
     digitalWrite(8 , HIGH); // MOTOR
     MOTOR_STATUS = "OFF";
   }

  }

  //MANUAL PHYSICAL BUTTONS


   if( MODE== "MANUAL" && !error_status)
   {
     int temp_manual_on = digitalRead(11);
     if(temp_manual_on == 1)
     {
       digitalWrite(13 , HIGH); //MOTOR ON
       digitalWrite(8 , LOW);  // MOTOR
       MOTOR_STATUS = "ON";
     }
     int temp_manual_off = digitalRead(12);
     if(temp_manual_off ==1)
     {
       digitalWrite( 13 , LOW); // MOTOR OFF
       digitalWrite(8 , HIGH); // MOTOR
       MOTOR_STATUS = "OFF";
     }
```

```
  }



// MOTOR WORKING CHECKER

if( MOTOR_STATUS == "ON" && !error_status )
{
 if(timmer ==0)
 {
   val = distance;
 }
 timmer++;

  if(timmer > 50)
  {
   if(val == distance)
   {

     digitalWrite( 13 , LOW); // MOTOR OFF
     digitalWrite(8 , HIGH); //MOTOR
     MOTOR_STATUS = "OFF";
     digitalWrite( 2 , HIGH); // ERROR LIGHT ON
     digitalWrite( 3 , LOW);  // MOTOR GOOD OFF
     error_status = true;

     Serial.println("bgfxcvbn");

   }
   timmer =0;

  }


}


int temp_button_status = digitalRead(4);
if(  temp_button_status == 1 )
{
 error_status = false;
 digitalWrite( 2 , LOW); // ERROR OFF
 digitalWrite( 3 , HIGH); // MOTOR GOOD ON
}


if(distance == 1)
{
 digitalWrite(A0 , HIGH);
 digitalWrite(A1 , HIGH);
 digitalWrite(A2 , HIGH);
 digitalWrite(A3 , HIGH);
}

if(distance == 2)
{
   digitalWrite(A0 , HIGH);
   digitalWrite(A1 , HIGH);
   digitalWrite(A2 , HIGH);
   digitalWrite(A3 , LOW);
```

```
  }
  if(distance == 2)
  {
     digitalWrite(A0 , HIGH);
     digitalWrite(A1 , HIGH);
     digitalWrite(A2 , LOW);
     digitalWrite(A3 , LOW);
  }

  if(distance == 4)
  {
     digitalWrite(A0 , HIGH);
     digitalWrite(A1 , LOW);
     digitalWrite(A2 , LOW);
     digitalWrite(A3 , LOW);
  }

  if(distance == 5)
  {
     digitalWrite(A0 , LOW);
     digitalWrite(A1 , LOW);
     digitalWrite(A2 , LOW);
     digitalWrite(A3 , LOW);
  }

  sending_timmer++;

  if( sending_timmer >= 40 )
  {
   if( distance >=5  ||  distance <= 0)
   {
     BTserial.println("FAULT DATA");
   }
   float temp_water_per = ((5-distance))*25;
   BTserial.println(String(temp_water_per));

   Serial.print("SENT   ");
   Serial.println(temp_water_per);

   sending_timmer =0;
  }
  if( MODE == "MANUAL" )
  {
   digitalWrite(A4 , HIGH);
   digitalWrite(A5 , LOW);
  }
  if( MODE == "AUTOMATIC" )
  {
   digitalWrite(A4 , LOW);
   digitalWrite(A5 , HIGH);
  }

}

```