```
#include <ThingerESP8266.h>


#define input 0       //D3  Flowmeter

#define trigPin 2     //D4

#define echoPin 4     //D2 Ultrasonic

#define forward 14    //D5

#define reverse 12    //D6 Controller


#define USERNAME "Hb027"

#define DEVICE_ID "NodeMCU"

#define DEVICE_CREDENTIAL "Something"


#define SSID "Haan"

#define SSID_PASSWORD "123445678"


ThingerESP8266 thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);



int X;

int Y;

float TIME = 0;

float FREQUENCY = 0;

float WATER = 0;

float TOTAL = 0;

float LS = 0;                  //Flow Meter


     long duration;

     int distance;

     int percent=0;                 //Ultra_sonic_sensor
int Opening=0;

int Openingprev=0;

int rev=0;


void setup() {

         Serial.begin(115200);


        pinMode(LED_BUILTIN, OUTPUT);         //Inbuild Led test

        pinMode(input,INPUT);                 //Flowmeter

        pinMode(trigPin, OUTPUT);             // Sets the trigPin as an Output

        pinMode(echoPin, INPUT);              // Sets the echoPin as an Input

        pinMode(forward,OUTPUT);              //Forward of gate valve

        pinMode(reverse,OUTPUT);              //Reverse of gate valve


  thing.add_wifi(SSID, SSID_PASSWORD);


  thing["led"] << digitalPin(LED_BUILTIN);


  thing["opening"] << inputValue(Opening);




  thing["water"] >> outputValue(WATER);

  thing["water total"] >> outputValue(TOTAL);

  thing["percent"] >> outputValue(percent);


}


void loop() {
    X = pulseIn(input, HIGH);

    Y = pulseIn(input, LOW);

    TIME = X + Y;

    FREQUENCY = 1000000/TIME;

    WATER = FREQUENCY/7.5;

    LS = WATER/60;

     if(FREQUENCY >= 0)

      {

       if(isinf(FREQUENCY)&&(WATER>20.00))

       {

         Serial.println("VOL. :0.00");

         Serial.print("TOTAL:");

         Serial.print( TOTAL);

         Serial.println(" L");

       }else

       {

         TOTAL = TOTAL + LS;

         Serial.print("VOL.: ");

         Serial.print(WATER);

         Serial.println(" L/M");

         Serial.print("TOTAL:");

         Serial.print( TOTAL);

         Serial.println(" L");

        }

      }
digitalWrite(trigPin, LOW);

delayMicroseconds(2);                   // Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);
```

```
        digitalWrite(trigPin, LOW);          // Reads the echoPin, returns the sound wave travel time in
microseconds

        duration = pulseIn(echoPin, HIGH);     // Calculating the distance

        distance= duration*0.034/2;          // Prints the distance on the Serial Monitor

        Serial.print("Distance: ");

        Serial.println(distance);

        if(distance>100)

        { percent=percent;

          }

         else

         { percent=distance;

           }

         Serial.print("Percent: ");

         Serial.println(percent);              //Prints tank level percentage


thing.handle();


   Serial.print("Current:");

   Serial.println(Opening);

   Serial.print("Previous:");

   Serial.println(Openingprev);

  if(Openingprev==Opening)

  {   digitalWrite(forward, LOW);

     digitalWrite(reverse, LOW);

  }else if(Openingprev!=Opening)

  {

     switch (Opening)

       {

        case 25:
```

```
{   digitalWrite(reverse, HIGH);
    delay(rev);
    digitalWrite(reverse, LOW);
    delay(200);
    digitalWrite(forward, HIGH);
    delay(1500);
    digitalWrite(forward, LOW);
    rev=1500;
}break;
case 50:
{   digitalWrite(reverse, HIGH);
    delay(rev);
    digitalWrite(reverse, LOW);
    delay(200);
    digitalWrite(forward, HIGH);
    delay(2500);
    digitalWrite(forward, LOW);
    rev=2500;
}break;
case 75:
{    digitalWrite(reverse, HIGH);
     delay(rev);
     digitalWrite(reverse, LOW);
     delay(200);
     digitalWrite(forward, HIGH);
     delay(3500);
     digitalWrite(forward, LOW);
     rev=3500;
 }break;
```

```
         default:{
            digitalWrite(forward, LOW);
            digitalWrite(reverse, LOW);}
        }
  }
  Openingprev=Opening;
}
```