```
#include <OneWire.h>

#include <DallasTemperature.h>

#include <YunClient.h>

#include <ThingerYun.h>

#include "RTClib.h" //include Adafruit RTC library

RTC_DS3231 rtc; //Make a RTC DS3231 object


#define USERNAME "rasadin"

#define DEVICE_ID "rasadin_project"

#define DEVICE_CREDENTIAL "V7@UZ@Y516lv"

ThingerYun thing(USERNAME, DEVICE_ID,
DEVICE_CREDENTIAL);


const int analogInPin_level = A2;

const int analogInPinOxy = A3;

float sensor_rasadin_oxy_Value = 0;

float LevelValue = 0;

const int ledPin =  13;

const int pin_temp = 4;

const int analogInPin = A0;

int sensorValue = 0;

unsigned long int avgValue;

float b;
```

```
int buf[10],temp;

OneWire oneWireDS(pin_temp);

DallasTemperature DS_temp(&oneWireDS);


//Set the names of days

char daysOfTheWeek[7][12] = {"Sunday", "Monday",
"Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday"};

String stringOne, stringTwo, stringThree;

void setup () {


DS_temp.begin();

pinMode(ledPin, OUTPUT);

Bridge.begin();

thing["LED"] << digitalPin(ledPin);

thing["DATE Y_M_D"] >> outputValue(DATE_RASADIN());

thing["PH"] >> outputValue(ph());

thing["TEMPERATURE"] >> outputValue(temp1());

thing["TURBIDITY"] >> outputValue(tur());

thing["DISSOLVED OXYGEN"] >> outputValue(oxy());

thing["WATER LEVEL"] >> outputValue(level());


  //Serial.begin(9600); //Begin the Serial at 9600 Baud
  stringOne = String("/");
  stringTwo = String("/");
  stringThree = String();
  //Print the message if RTC is not available
  if (! rtc.begin()) {
    //Serial.println("Couldn't find RTC");
    while (1);
  }

  //Setup of time if RTC lost power or time is not set
  if (rtc.lostPower()) {
    //Sets the code compilation time to RTC DS3231
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }




}

void loop() {
thing.handle();
pson data;
data["DATE Y_M_D"] = DATE_RASADIN();
```

```
data["PH"] = ph();
data["TEMPERATURE"] = temp1();
data["TURBIDITY"] = tur();
data["DISSOLVED OXYGEN"] = oxy();
data["WATER LEVEL"] = level();

thing.write_bucket("bucket1", data);
}



String DATE_RASADIN() {
  //Set now as RTC time
  DateTime now = rtc.now();
int a= (now.year());

int b= (now.month());

int c= (now.day());

stringThree =  a + stringOne + b + stringTwo + c;
return stringThree;


}
```

```
float ph() {
   for(int i=0;i<10;i++)
 {
  buf[i]=analogRead(analogInPin);
  delay(10);
 }
 for(int i=0;i<9;i++)
 {
  for(int j=i+1;j<10;j++)
  {
   if(buf[i]>buf[j])
   {
    temp=buf[i];
    buf[i]=buf[j];
    buf[j]=temp;
   }
  }
 }
 avgValue=0;
 for(int i=2;i<8;i++)
 avgValue+=buf[i];
 float pHVol=(float)avgValue*5.0/1024/6;
 float phValue = -5.70 * pHVol + 21.34;

 return phValue;
 }


float temp1(){
  DS_temp.requestTemperatures();
  return DS_temp.getTempCByIndex(0);
  }


float tur() {
  int sensorValue = analogRead(A1);
  float NTU = sensorValue ;
  //float final_tur= -1120.4*NTU*NTU+5742.3*NTU-4352.9;
  return NTU;
 }

float level() {
LevelValue = analogRead(analogInPin_level);
float sensor_noise= 0;
float sensor_pos= -1;
float water_level = LevelValue*100/1024;

if (water_level<1){

float final_water_level= water_level-
sensor_noise*sensor_pos;

return final_water_level;

 }

 else{

float final_water_level= water_level-sensor_noise;

return final_water_level;


 }

 }


 float oxy() {

sensor_rasadin_oxy_Value = analogRead(analogInPinOxy);

  float oxy_rasadin = sensor_rasadin_oxy_Value/100;

 //mg/L

  return oxy_rasadin;

 }
```