LAMPIRAN



Gambar 1. Bimbingan Pertama terkait Studi Literatur



Gambar 2. Bimbingan Kedua Penyerahan file kartu kontrol dan capstone desgin project



Gambar 3. Bimbingan Ketiga terkait pertanyaan wawancara



Gambar 4. Wawancara dengan penambak ikan



Gambar 5. Bimbingan Keempat terkait hasil wawancara



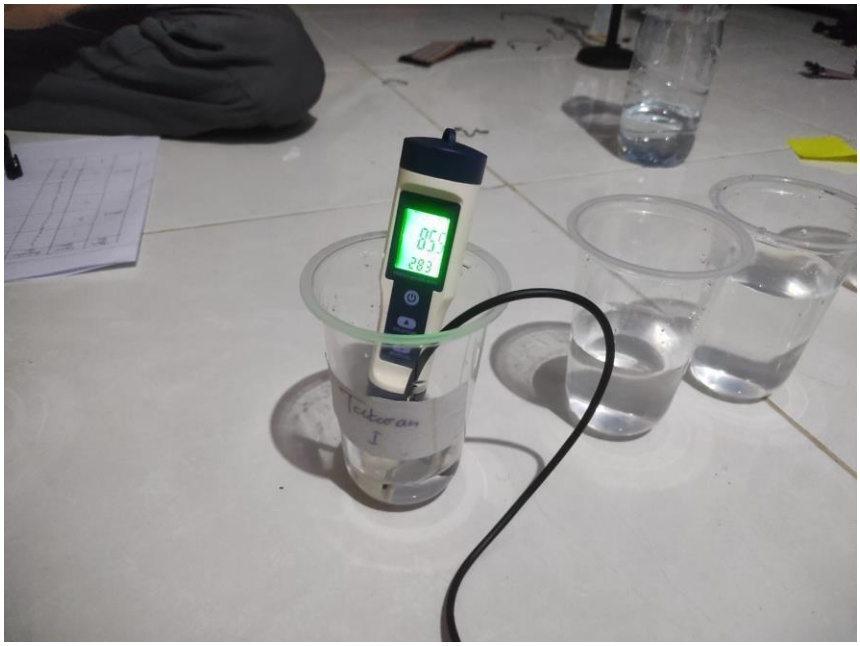
Gambar 6. Pengecekan sensor pH, turbidity, tds, dan suhu



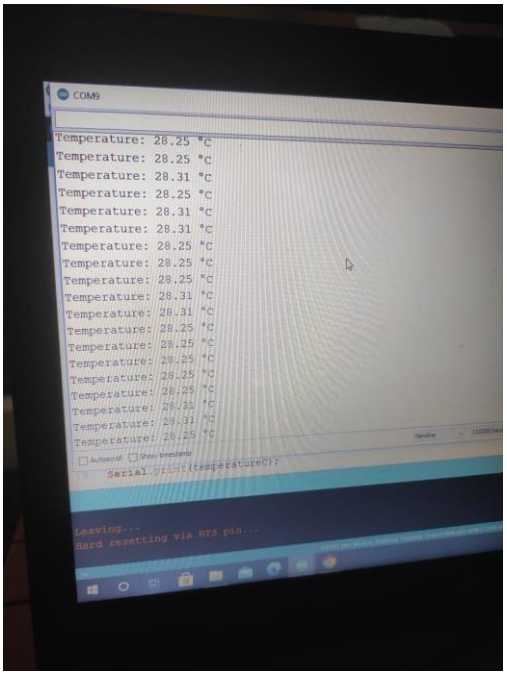
Gambar 7. Kalibrasi dan Pengujian sensor pH



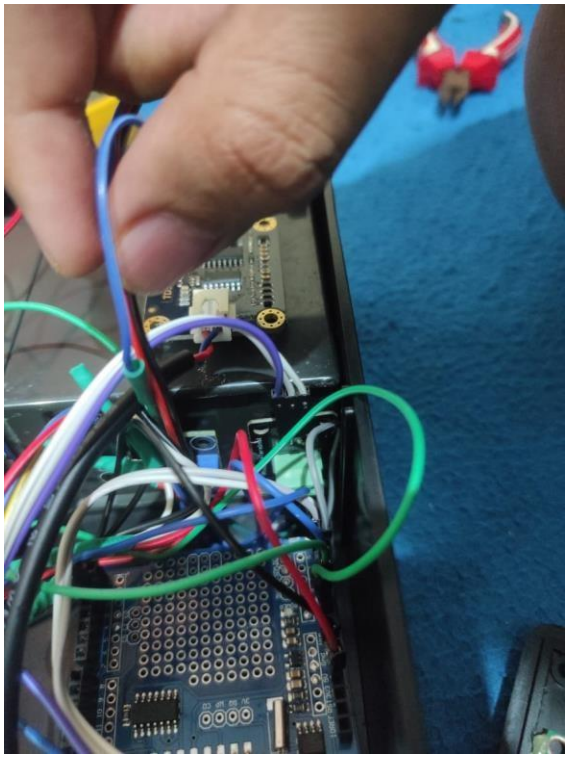
Gambar 8. Kalibrasi dan Pengujian sensor turbidity



Gambar 9. Kalibrasi dan Pengujian sensor TDS



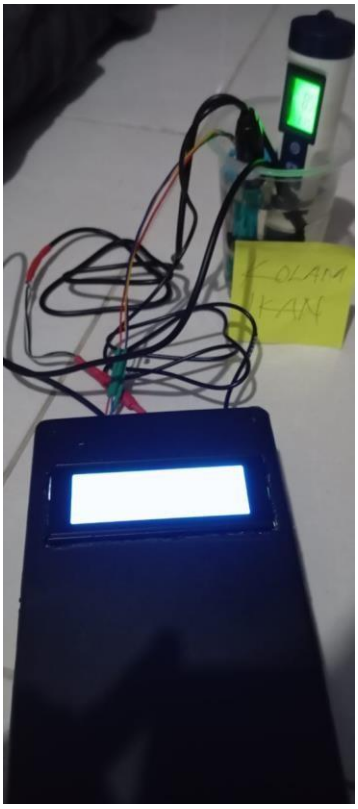
Gambar 10. Kalibrasi dan Pengujian sensor Suhu



Gambar 11. Perakitan alat monitoring kualitas air



Gambar 12. Bimbingan Ujicoba alat monitoring kualitas air



Gambar 13. Pengujian alat ujicoba alat monitoring kualitas Air



Gambar 14. Tampilan di LCD pada saat Pengujian alat monitoring kualitas Air



Gambar 15. Bimbingan terkait penyusunan laporan capstone



Gambar 16. Pengambilan air kolam budidaya ikan di Suppa, Pinrang

SKETCH PROGRAM

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <RTClib.h>
#include <SD.h>
#define PH_PIN A0
#define TURBIDITY_PIN A1
#define TDS_PIN A2
#define ONE_WIRE_BUS A3
#define SD_CS 10
// === Objek Sensor dan Modul ===
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature suhuSensor(&oneWire);
LiquidCrystal_I2C lcd(0x27, 20, 4);
RTC_DS1307 rtc;
// === TDS ===
#define VREF 5.0
#define SCOUNT 30
// === Kalibrasi Turbidity ===
#define OFFSET 2500
#define SCALE -500
#define MAX_NTU 3000
// === Waktu Sampling ===
unsigned long previousMillis =
0;
const unsigned long interval = 30000; // 30 detik
void setup() {
  Serial.begin(115200);
  suhuSensor.begin();
  lcd.init(); lcd.backlight(); lcd.clear();
  if (!rtc.begin()) {
    lcd.setCursor(0, 0); lcd.print("RTC Tidak
Ditemukan");
    while (1);
  }

  if (!rtc.isrunning()) {
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }

  if (!SD.begin(SD_CS)) {
    lcd.setCursor(0, 0); lcd.print("SD Gagal!");
    while (1);
  }

  lcd.setCursor(0, 0); lcd.print("Sistem Siap");
  delay(2000);
}
```

```
void loop() {
  if (millis() - previousMillis >= interval) {
    previousMillis = millis();
    suhuSensor.requestTemperatures();
    float suhu = suhuSensor.getTempCByIndex(0);
    float ph = bacaPH();
    float ntu = bacaTurbidity();
    float tds = bacaTDS(suhu);
    int persen = map(ntu, 0, MAX_NTU, 0, 100);
    persen = constrain(persen, 0, 100);
    DateTime now = rtc.now();
    // === Tampilkan ke LCD
    === lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("pH: "); lcd.print(ph, 2);
    lcd.setCursor(11, 0);
    lcd.print(now.timestamp(DateTime::TIMESTAMP_TIME));
    lcd.setCursor(0, 1);
    lcd.print("TDS: "); lcd.print(tds, 0); lcd.print("
ppm");
    lcd.setCursor(0, 2);
    lcd.print("NTU: "); lcd.print(ntu, 1); lcd.print("
(");
    lcd.print(persen); lcd.print("%)");
    lcd.setCursor(0, 3);
    lcd.print("Temp: "); lcd.print(suhu, 1);
lcd.print((char)223); lcd.print("C");
    // === Simpan ke SD Card (CSV) ===
    File dataFile = SD.open("log.csv",
    FILE_WRITE); if (dataFile) {
      dataFile.print(now.timestamp());
      dataFile.print(","); dataFile.print(ph, 2);
      dataFile.print(","); dataFile.print(tds, 0);
      dataFile.print(","); dataFile.print(ntu, 1);
      dataFile.print(","); dataFile.print(persen);
      dataFile.print(","); dataFile.println(suhu, 1);
      dataFile.close();
    } else {
      Serial.println("Gagal membuka file SD");
    }

float bacaPH() {
  unsigned int pH_Value = 0;
  for (int i = 0; i < 10; i++) {
    pH_Value += analogRead(PH_PIN);
    delay(10);
  }
```

```
  float avgAnalog = (float)pH_Value / 10.0;
  float tegangan = avgAnalog * (5.0 / 1023.0);
  float kalibrasi = 42.7 ;
  float ph = -10.15* tegangan + kalibrasi;
  return ph;
}

float bacaTurbidity() {
  int sensorValue = analogRead(TURBIDITY_PIN);
  float voltage = sensorValue * (5.0 / 1023.0);
  float ntu = OFFSET + SCALE * voltage;
  return constrain(ntu, 0, MAX_NTU);
}

float bacaTDS(float temperature) {
  int buffer[SCOUNT];
  for (int i = 0; i < SCOUNT; i++)
    {          buffer[i]          =
    analogRead(TDS_PIN); delay(5);
  }

  float avgVoltage = getMedianNum(buffer, SCOUNT)
* VREF / 1024.0;
  float compensationCoefficient = 1.0 + 0.02 *
(temperature - 25.0);
  float compensationVoltage = avgVoltage /
compensationCoefficient;
  float tdsValue = (133.42 *
pow(compensationVoltage, 3)- 255.86 *
pow(compensationVoltage, 2)
+ 857.39 * compensationVoltage) * 0.5;
  return tdsValue;
}

int getMedianNum(int bArray[], int iFilterLen) {
  int bTab[iFilterLen];
  for (byte i = 0; i < iFilterLen; i++) bTab[i] =
bArray[i];
  int i, j, bTemp;
  for (j = 0; j < iFilterLen - 1; j++) {
    for (i = 0; i < iFilterLen - j - 1; i++) {
      if (bTab[i] > bTab[i + 1]) {
        bTemp = bTab[i];
        bTab[i] = bTab[i + 1];
        bTab[i + 1] = bTemp;
      }

    }
```

```
  }

  if ((iFilterLen & 1) > 0)
    bTemp = bTab[(iFilterLen - 1) / 2];
  else
    bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2
- 1]) / 2;
  return bTemp;
}
```

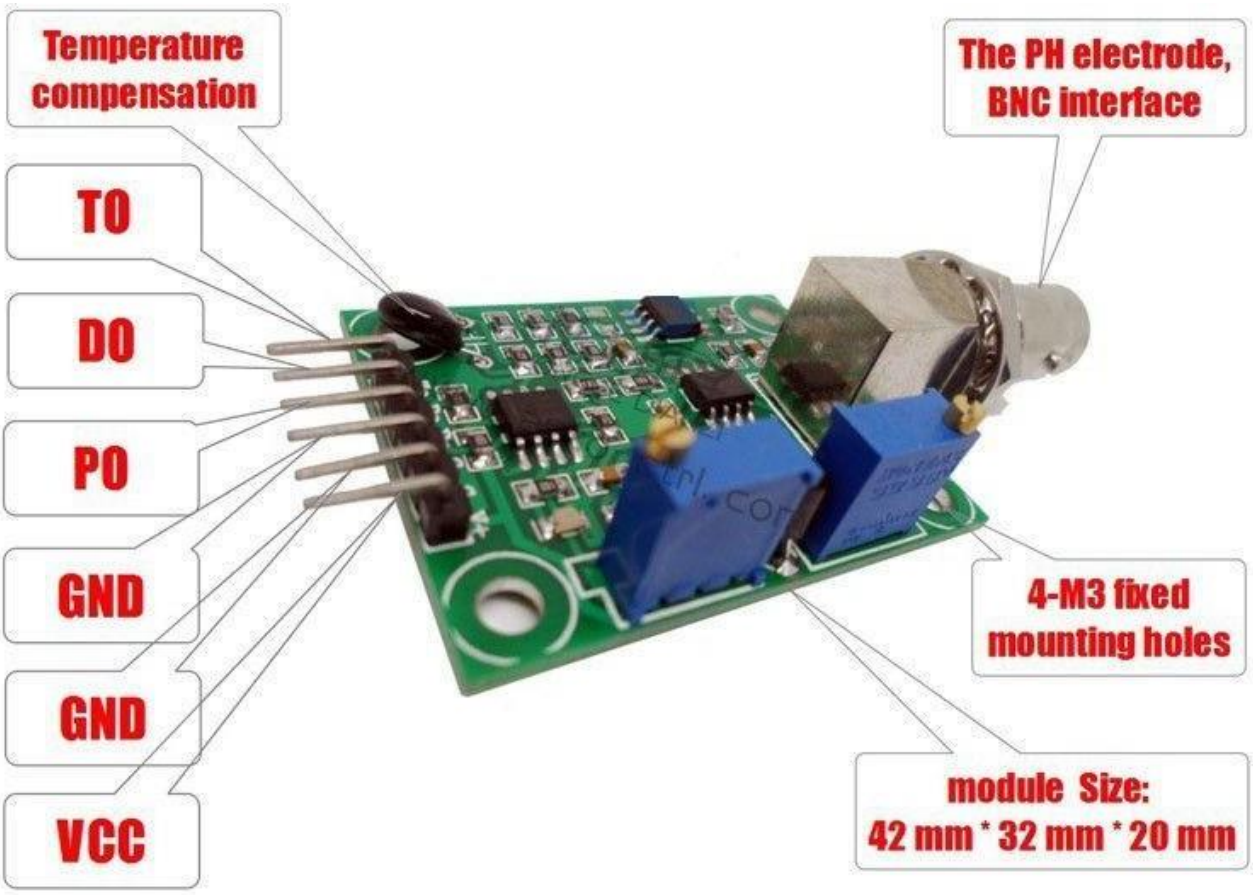
How to use a PH probe and sensor

If you worked with PH metering before you will know that PH values range from 0-14. Where PH 0 Will be very acidic, PH 7 will be neutral and PH 14 very alkaline. Water is near a PH 7 and this is usually around here that we will need to monitor PH of many things. A swimming pool, for example, should be slightly alkaline at 7.2, hydroponics systems around 6 (for optimum plant nutrition takeup) and aquaponics around 6.8.

I wrote this **PH probe and sensor** "how to" because it is not as straightforward as one would think (but quite easy when you understand the ins and outs) mostly because there is not a lot of information on this on the Internet, surely not detailed information.

We will first look at the ph probe module board and then the PH probe because both the PH probe and sensor have to be set correctly:

- offset setting
- limit setting
- sketch to test the board analogue range
- sketch for PH reading and calibration.
- calibration of PH probe
- PH probe usage
  The ph probe module in this tutorial is available on our site here: PH probe module BNC conector



PH Probe Sensor Pinout

TO – Temperature output
DO – 3.3V Output (from ph limit pot)
PO – PH analog output ==> **Arduino A0**
Gnd – Gnd for PH probe (can come from Arduino GND pin) **==> Arduino GND**
Gnd – Gnd for board (can also come from Arduino GND pin) **==> Arduino GND**
VCC – 5V DC (can come from Arduino 5V pin) **==> Arduino 5V pin**
POT 1 – Analog reading offset (Nearest to BNC connector)
POT 2 – PH limit setting

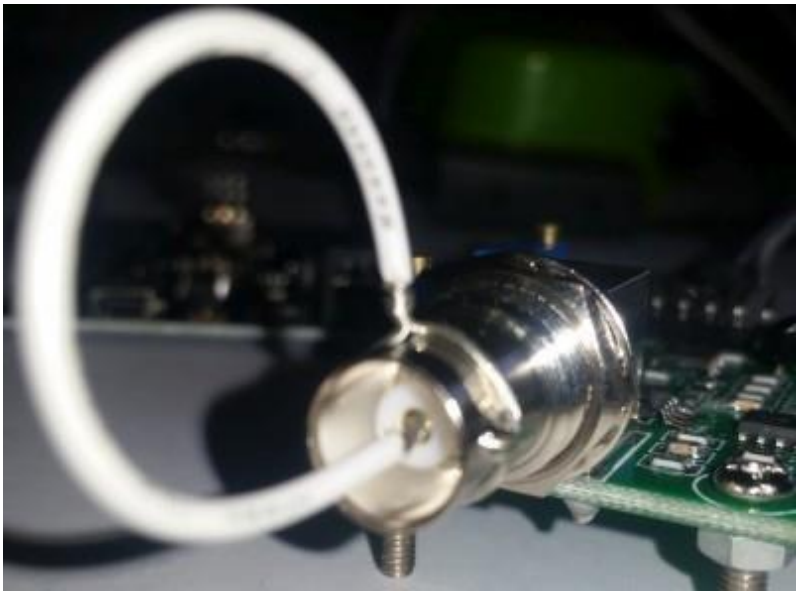
PH probe module Offset and how to use it.

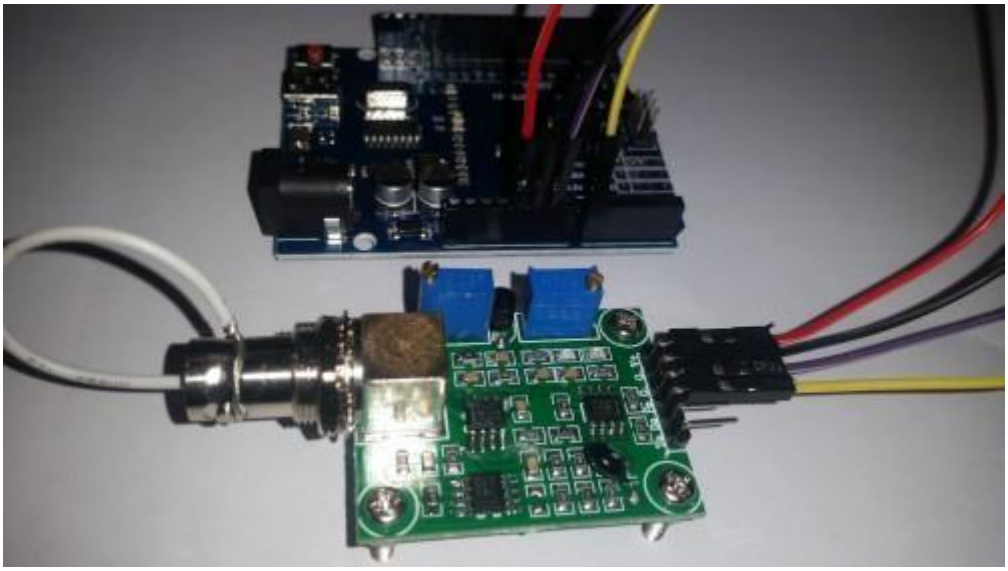
This board have the ability to supply a voltage output to the analogue board that will represent a PH value just like any other sensor that will connect to an analog pin. Ideally, we want a PH 0 represent 0v and a PH of 14 to represent 5V.

BUT there is a catch……, this board by default have PH 7 set to 0V (or near it, it differs from one PH probe to another, that is why we have to calibrate the probe as you will see later on), This means that the voltage will go into the minuses when reading acidic PH values and that cannot be read by the analog Arduino port. The offset pot is used to change this so that a PH 7 will read the expected 2.5V to the Arduino analog pin, the analog pin can read voltages between 0V and 5V hence the 2.5V that is halfway between 0V and 5V as a PH 7 is halfway between PH 0 and PH 14,

You will need to turn the offset potentiometer to get the right offset, The offset pot is the blue pot nearest to the BNC connector.

To set the offset is easy. First, you need to disconnect the probe from the circuit and short-circuit the inside of the BNC connector with the outside to simulate a neutral PH (PH7). I took a piece of wire, strip both sides, wrap the one side around the outside of the BNC connector and push the other side into the BNC hole. This short-circuit represents about a neutral PH reading of 7.





There are two ways you can do the adjustment.

If you have a multimeter handy you can measure the value of the PO pin and adjust the offset potentiometer until PO measures 2.5V.

I prefer to just use the sketch below. Just download it to your Arduino as you will with any other sketch, open serial monitor and view the reading there. All this sketch does is to print the volts it receives from the analog pin and print it to the serial monitor. It of course first changes the digital value to volts to make it easier. Now simply turn the offset pot until it is exactly 2.5V. You can learn more about reading voltages and digital representation of volts here: https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage

Offset sketch

```
void setup() {
 // initialize serial communication at 9600 bits per second:
 Serial.begin(9600);
}


// the loop routine runs over and over showing the voltage on A0
void loop() {
 // read the input on analog pin 0:
 int sensorValue = analogRead(A0);
 // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
 float voltage = sensorValue * (5.0 / 1023.0);
 // print out the value you read:
 Serial.println(voltage);
 delay(300);
}
```

PH limit setting

There is another pot that acts like a limit switch. Basically, the D0 pin on the sensor board will supply 3.3V to the pin until a preset PH value (that you set with the limit pot) is reached, at this point a red LED will light up and the pin will go down to about 0V.

I did not play with this much but suppose it can be handy if you want to activate a buzzer or something if a certain PH is reached, it will work great on an Arduino digital port – that will go high from about 2V up.

This will work if the PH value goes higher than the set value. If you want it to trigger something when the PH goes lower, you need to monitor the digital pin to trigger when the digital pin goes low.

You will unfortunately not be able to set this limit between two values, either if the pH goes up to high or if the PH drop to low. Programmatically you of cause can do an upper and lower limit.

Connecting and calibrating the PH probe.

The hard part is over and this offset does not have to be set again, even if you change PH probes.
We have PH probes available here: PH probe Electrode BNC connector
Here is a couple of things to know about PH probes:

1. The probes readings change over time and need to be calibrated every now and again to make sure the value is still the same and be adjusted if it did change.
2. You need at least one PH buffer solution to calibration your PH probe. They are available at many different PH values, A buffer solution of 6.86 and 4.01 is most common as it covers the range of most applications. If you are only going to use one buffer solution make sure its value is near the value range you will use in your normal tests – if it is pool water a buffer solution of 6.86 is usually near enough.
3. Buffers come in pre-made solutions or as a powder. I prefer the powder because it is cheaper and does not have an expiration date. The powder is easy to make up as well, I suppose it depends on the power you will use, the one I use you add the powder to 250ml distilled water and stir until all powder is dissolved. It will last about a month once you added water to it.



A PH probe **takes some time** to get to the right value, allow it to be in the liquid you want to measure for at least two minutes or longer, it does not mean it will be stable at one ph value,

it will jump around a bit between 3 or 4 values but on the last digit, for example, between 6.84 – 6.88

4. PH values differ in different temperatures, although that might sound cumbersome, in the temperature range between 10 – 30 degrees Celcius the PH does not differ and from 30 degrees Celcius it goes up with about a pH of 0.01 to 50 degrees Celcius that is about 0.06. In most uses, it will be below 30 degrees Celcius and temperature do not have to be calculated in.

Hook up your PH probe after you removed the wire you used to short-circuit the BNC connector and download the sketch below.

PH measurement sketch

```
float calibration = 0.00; //change this value to calibrate
const int analogInPin = A0;
int sensorValue = 0;
unsigned long int avgValue;
float b;
int buf[10],temp;
void setup() {
Serial.begin(9600);
}

void loop() {
 for(int i=0;i<10;i++)
 {
 buf[i]=analogRead(analogInPin);
 delay(30);
 }
 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 + calibration;
 Serial.print("sensor = ");
 Serial.println(phValue);


 delay(500);
}
```

A note on buffer solutions: do not **CROSS CONTAMINATE!** What I mean by this is to not take the probe from one buffer solution to another or from a liquid sample you tested to a buffer solution without rinsing it thoroughly with distilled water first. You will change your buffer solutions ph and your calibration will be off.

When you place the probe in the first solution you might be surprised at how far off it can be, it's normal though. **Remember to leave the probe in the solution for at least two minutes to stabilise.** In the script's top line you will see a variable called ***"calibration".*** Change this value to the difference between what you see in serial monitor and the buffer solutions value, for example, if you read 5.81 and the buffer solution is 6.86 you should change the variable's value to 2.05.

Upload the changed sketch and see how the value looks now.

Some ideas for you.

You can add a potentiometer to your project and program that to change the calibration for you. You always run a risk that the pot might be adjusted my mistake so a button with a 5- second delay can be programmed to put the unit in calibration mode. This becomes great if you add an LCD screen to it and if you add the calibration value in EEPROM so it holds it even if the PH project is powered off.

How about adding a buzzer to the 3.3V limit output to notify you when the PH is out of range. Usually an upper or lower will be enough, in most applications, you will either be worried about high or low PH values but not both. If this is a problem and you need the PH to be in a certain range you can easily do that programmatically and have the buzzer on a digital pin. If it is crucial you can even add a GSM module to this so you can get an SMS or how about a dosing pump to automatically get the PH back in range.

I just might, but do not hold your breath (so much to do but so little time) :-), do something like that in the future with our cheap Skeleton Duino.

PH probe and sensor conclusion

If you go through these steps once it becomes as easy as pie. All the parts, even those I mention in the ideas section is available on our site.

[RETIRED] LCD DISPLAY 20X4 I2C WHITE/BLUE



DESCRIPTION

This product is retired and we won't sell it anymore. This page is for reference only.

LCD display with HD44780 controller enables showing text, symbols, value or anything else that comes from Croduino/Arduino. It is capable of showing 16 symbols in two lines and therefore can be used in many projects that need to send clearly visible and legible output information. An I2C LCD adapter is soldered to this LCD screen, which enables characters to be displayed on the LCD screen via the I2C bus. It also has an easyC connector for a really simple connection.

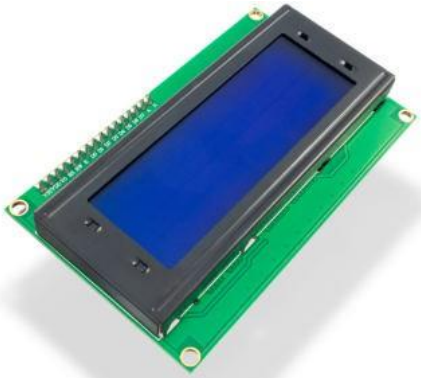
FEATURES

- Color of symbols: White
- Backlight: Blue
- Number of chars: 20 chars in 4 rows
- Voltage: 3V3
- Communication protocol: I2C
- With soldered I2C adapter
- Display dimensions: 77 x 25,2 mm / 3 x 1 inch
- Dimensions: 98 x 60 x 12 mm / 3.9 x 2.4 x 0.5 inch

USEFUL LINKS

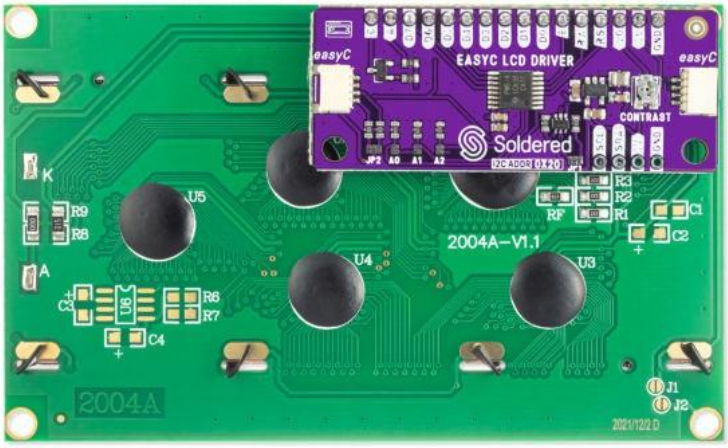
- Arduino library
- Datasheet

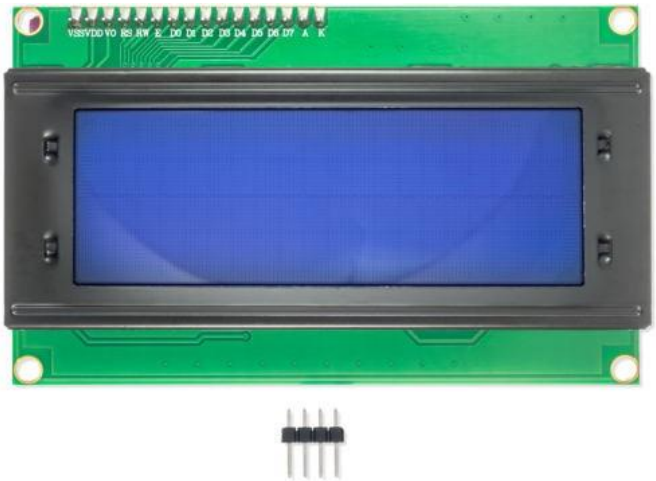
OTHER IMAGES











Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Soldered:

333172

DS18B20 Waterproof Temperature Sensor Cable



Product Description

This Maxim-made item is a digital thermo probe or sensor that employs DALLAS DS18B20. Its unique 1-wire interface makes it easy to communicate with devices. It can converts temperature to a 12-bit digital word in 750ms (max). Besides, it can measures temperatures from -55°C to +125°C (-67F to +257F). In addition, this thermo probe doesn't require any external power supply since it draws power from data line. Last but not least, like other common thermo probe, its stainless steel probe head makes it suitable for any wet or harsh environment.

The datasheet of this DS18B20 Sensor Cable can be found from:
http://www.quick-teck.co.uk/ElectronicElement/eeList.php?typeId=97#title

Feature:

Power supply range:	3.0V to 5.5V
Operating temperature range:	-55°C to +125°C (-67F to +257F)
Storage temperature range:	-55°CC to +125°C (-67F to +257F)
Accuracy over the range of -10°C to +85°C:	±0.5°C
3-pin 2510 Female Header Housing	

Waterproof Stainless steel sheath	
Stainless steel sheath	
Size of Sheath:	6*50mm
Connector:	RJ11/RJ12, 3P-2510, USB.
Pin Definition:	RED: VCC Yellow: DATA Black: GND
Cable length:	1meter, 2m, 3m, 4m are available upon request.

Application:

The DS18B20 Digital Temperature Probe provides 9 to 12 bit (configurable) temperature readings which indicate the temperature of the device. Information is sent to/from the DS18B20 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS18B20. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and process monitoring and control.

Details：



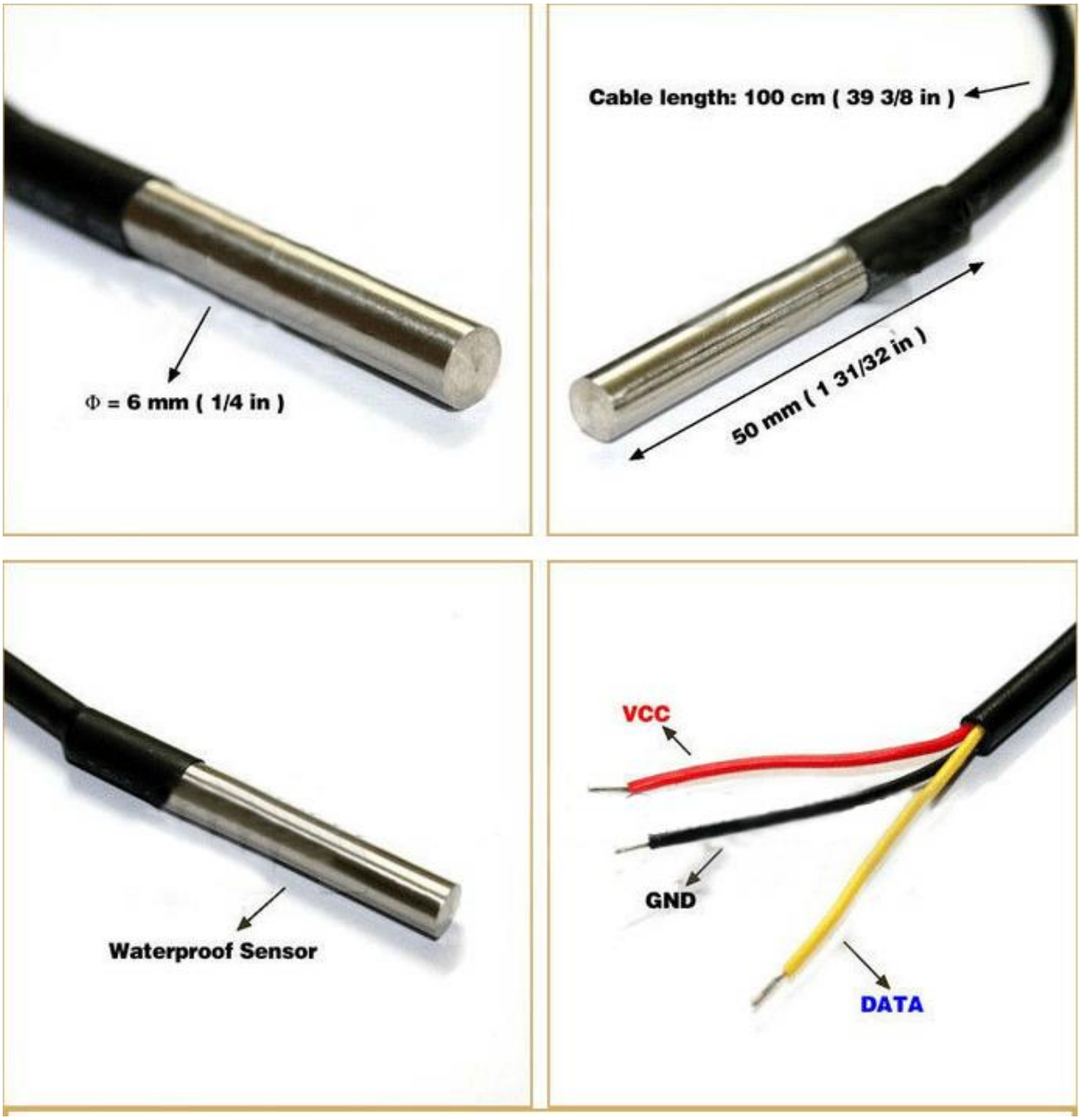

Φ = 6 mm (1/4 in)
Cable length: 100 cm (39 3/8 in)
50 mm (1 31/32 in)
Waterproof Sensor
VCC
GND
DATA

Figure 2



Gravity: Analog TDS Sensor / Meter For Arduino

SKU: SEN0244

TDS (Total Dissolved Solids) indicates that how many milligrams of soluble solids dissolved in one liter of water. In general, the higher the TDS value, the more soluble solids dissolved in water, and the less clean the water is. Therefore, the TDS value can be used as one of the references for reflecting the cleanliness of water.

TDS pen is a widely used equipment to measure TDS value. The price is affordable, and it is easy to use, but it is not able to transmit data to the control system for online monitoring to do some water quality analysis. The professional instrument has high accuracy and can send data to the control system, but the price is expensive for the ordinary people. To this end, we have launched an analog TDS sensor kit which is compatible with Arduino, plug and play, easy to use. Matching with Arduino controller, you can build a TDS detector easily to measure the TDS value of liquid.

This product supports 3.3 ~ 5.5V wide voltage input, and 0 ~ 2.3V analog voltage output, which makes it compatible with 5V or 3.3V control system or board. The excitation source is AC signal, which can effectively prevent the probe from polarization and prolong the life of the probe, meanwhile, increase the stability of the output signal. The TDS probe is waterproof, it can be immersed in water for long time measurement.

This product can be used in water quality application, such as domestic water, hydroponics. With this product, you can easily DIY a TDS detector to reflect the cleanliness of water to protect your health.

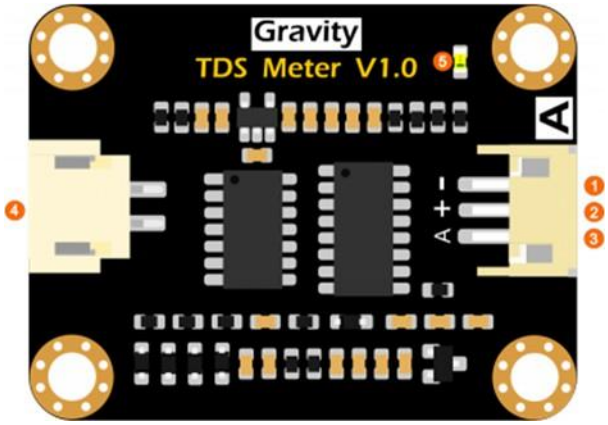
Attention:

1. The probe can not be used in water above 55 degrees centigrade.
2. The probe can not be left too close to the edge of the container, otherwise it will affect the reading.
3. The head and the cable of the probe are waterproof, but the connector and the signal transmitter board are not waterproof. Please be careful.

Specification

- **Signal Transmitter Board**
  Input Voltage: 3.3 ~ 5.5V
  Output Voltage: 0 ~ 2.3V
  Working Current: 3 ~ 6mA
  TDS Measurement Range: 0 ~ 1000ppm
  TDS Measurement Accuracy: ± 10% F.S. (25 ℋ)
  Module Size: 42 * 32mm
  Module Interface: PH2.0-3P
  Electrode Interface: XH2.54-2P
- **TDS probe**
  Number of Needle: 2
  Total Length: 83cm
  Connection Interface: XH2.54-2P
  Colour: Black
  Other: Waterproof Probe

Board Overview



Analog TDS Sensor / Meter For Arduino

Num	Label	Description
1	-	Power GND(0V)
2	+	Power VCC(3.3 ~ 5.5V)
3	A	Analog Signal Output(0 ~ 2.3V)
4	TDS	TDS Probe Connector
5	LED	Power Indicator

Tutorial

This tutorial will show you how to measure the TDS value of the water. Please read this tutorial carefully, and pay attention to the steps and details.



The probe can not to be used in water above 55 degrees centigrade.
The probe can not be too close to the edge of the container, otherwise it will affect the reading.
The head and the cable of the probe are waterproof, but the connector and the signal transmitter board are not waterproof.Please pay attention to use.

Requirements

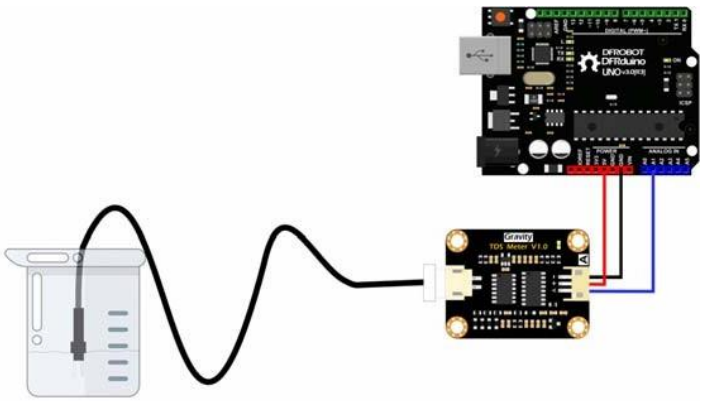
- **Hardware**
  DFRduino UNO R3 (or similar) x 1
  Analog TDS Sensor / Meter Module x 1
  TDS Probe x1
  Jumper Wires x3
  tested liquid x1

- **Software**

Arduino IDE (Version requirements: V1.0.x or V1.8.x), Click to Download Arduino IDE from Arduino®

https://www.arduino.cc/en/Main/Software%7C

Connection Diagram



Sample Code

```
/*************************************************

 DFRobot Gravity: Analog TDS Sensor / Meter For Arduino

 <https://www.dfrobot.com/wiki/index.php/Gravity:_Analog_TDS_Sensor_/_Meter_For_Arduino_SKU:_SEN0244>


 Created 2017-8-22

 By Jason <jason.ling@dfrobot.com@dfrobot.com>


 GNU Lesser General Public License.

 See <http://www.gnu.org/licenses/> for details.

 All above must be included in any redistribution


 /***********Notice and Trouble shooting***************

 1. This code is tested on Arduino Uno and Leonardo with Arduino IDE 1.0.5 r2 and 1.8.2.

 2. More details, please click this link: <https://www.dfrobot.com/wiki/index.php/Gravity:_Analog_TDS_Sensor_/_Meter_For_Arduino_SKU:_SEN0244>

 *************************************************/


#define TdsSensorPin A1

#define VREF 5.0      // analog reference voltage(Volt) of the ADC

#define SCOUNT  30           // sum of sample point

int analogBuffer[SCOUNT];    // store the analog value in the array, read from ADC

int analogBufferTemp[SCOUNT];

int analogBufferIndex = 0,copyIndex = 0;

float averageVoltage = 0,tdsValue = 0,temperature = 25;


void setup()

{

    Serial.begin(115200);

    pinMode(TdsSensorPin,INPUT);

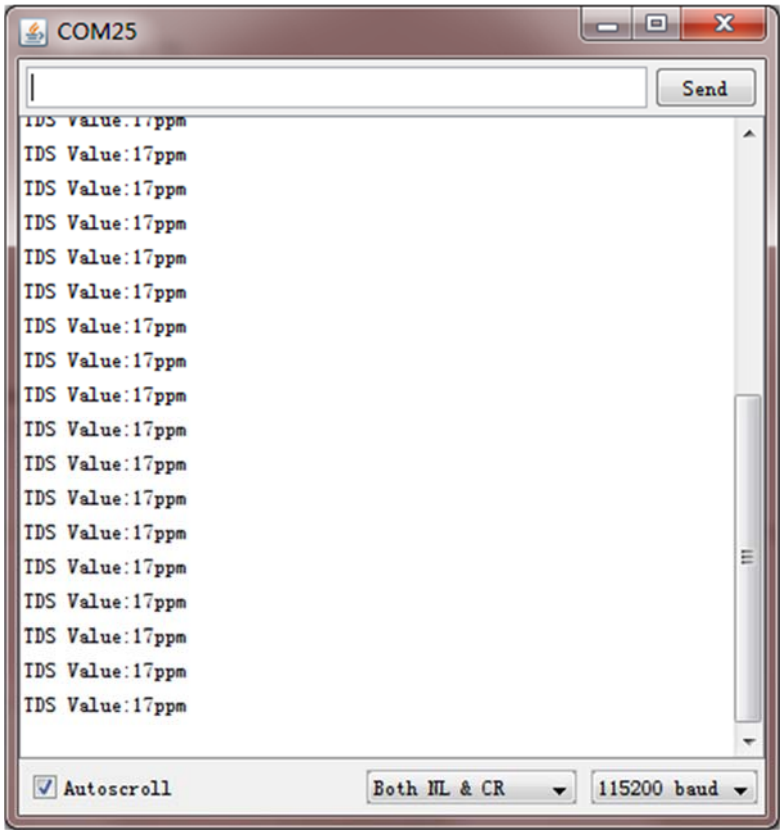
}
```

```
void loop()
{
   static unsigned long analogSampleTimepoint = millis();
   if(millis()-analogSampleTimepoint > 40U)     //every 40 milliseconds,re
ad the analog value from the ADC
   {
     analogSampleTimepoint = millis();
     analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin);    //read
the analog value and store into the buffer
     analogBufferIndex++;
     if(analogBufferIndex == SCOUNT)
         analogBufferIndex = 0;
   }
   static unsigned long printTimepoint = millis();
   if(millis()-printTimepoint > 800U)
   {
      printTimepoint = millis();
      for(copyIndex=0;copyIndex<SCOUNT;copyIndex++)
        analogBufferTemp[copyIndex]= analogBuffer[copyIndex];
      averageVoltage = getMedianNum(analogBufferTemp,SCOUNT) * (float)VREF
/ 1024.0; // read the analog value more stable by the median filtering alg
orithm, and convert to voltage value
      float  compensationCoefficient=1.0+0.02*(temperature-25.0);   //temp
erature compensation formula: fFinalResult(25^C) = fFinalResult(current)/(
1.0+0.02*(fTP-25.0));
      float compensationVolatge=averageVoltage/compensationCoefficient;  /
/temperature compensation
      tdsValue=(133.42*compensationVolatge*compensationVolatge*compensatio
nVolatge - 255.86*compensationVolatge*compensationVolatge + 857.39*compens
ationVolatge)*0.5; //convert voltage value to tds value
      //Serial.print("voltage:");
      //Serial.print(averageVoltage,2);
      //Serial.print("V   ");
      Serial.print("TDS Value:");
      Serial.print(tdsValue,0);
      Serial.println("ppm");
   }
```

```
}
int getMedianNum(int bArray[], int iFilterLen)
{
      int bTab[iFilterLen];
      for (byte i = 0; i<iFilterLen; i++)
      bTab[i] = bArray[i];
      int i, j, bTemp;
      for (j = 0; j < iFilterLen - 1; j++)
      {
      for (i = 0; i < iFilterLen - j - 1; i++)
          {
        if (bTab[i] > bTab[i + 1])
            {
            bTemp = bTab[i];
            bTab[i] = bTab[i + 1];
            bTab[i + 1] = bTemp;
         }
      }
      }
      if ((iFilterLen & 1) > 0)
    bTemp = bTab[(iFilterLen - 1) / 2];
      else
    bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
      return bTemp;
}
```

Expected Results

After uploading the sample code,open the serial monitor of the Arduino IDE. Then insert the TDS probe into the water, and gently stir it. Waiting for the reading to be stable, and you will get the TDS value of the water.



FAQ

Q1. Does this sensor have a temperature sensor? How to make the temperature compensation?

A. This TDS probe has no temperature sensor, but the temperature compensation algorithm is reserved in the sample code. The temperature variable in the sample code will default to 25 °C without a temperature sensor. You can add a waterproof temperature sensor to read the temperature,then update the temperature variable, to make automatic temperature compensation.

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum**.

Datasheet AZDM01

Turbidity sensor

- High reliability
- Fast response
- Waterproof design
- Long lifespan
- Strong anti-interference capability

Summary

AZDM01 is a photoelectric turbidity sensor with analog signal output. AZDM01 is powered by 5V DC with low power consumption and uses infrared light source with wavelength of 940nm. The light emitted by the source will pass through the target liquid and received by the light detector in the senor. The light intensity is measured by the light detector and converted to analog signal to represent the turbidity of target liquid. AZDM01 is designed to be able measure the turbidity in real-time. AZDM01 has outstanding features, such as simple structure, snap installation, waterproof, fast response, high reliability and stability, low cost, simple drive circuit etc.

Application scenario

AZDM01 can be used for turbidity detection of low viscosity and low corrosive liquids, such as washing water in washing machine, dishwasher, coffee machine, water in rivers and lakes.



Figure 1. AZDM01

1. Working principle

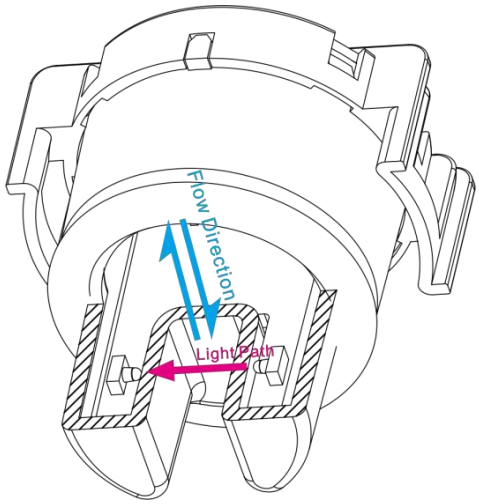


Figure 2. Sketch of AZDM01 with indication of flow direction and light path

AZDM01 has a built-in light detector and a light-emitting element as infrared light source. As shown in Figure 2, the target liquid flows in the direction indicated by the blue arrow and the emitted light pass through the liquid indicated by the red arrow. The light intensity is reduced due to the absorption in liquid and the absorption rate depends on the liquid turbidity, thus, the light intensity detected by the light detector is varied by the turbidity of liquid. The turbidity can be derived based on the output signal of the light detector.

1. Technical specification

Table 1. Technical specification

Supply voltage	DC:5.0 V (4.8-5.5 V)
Minimum supply current	20 mA
Typical supply current	25 mA
Maximum supply current	50 mA
Typical power	125 mW
Sampling period	>= 100 ms/time
Output method	Analog signal
Warm-up time	>= 100 ms
Operating temperature	4℃～85℃
Working humidity	0～95%RH
Service life	>10 years (25℃)

3. Pin assignment and drive circuit

3.1 Pin assignment

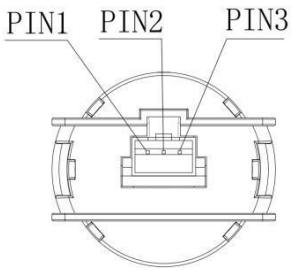


Figure 3. AZDM01 pinout

The connection interface of AZDM01 is XHB2.54mm 3P socket with buckle, which include 3 pins as shown in Figure 3. The definition of pins are shown in table 2.

Table 2. Pin definition

	Name	Description
PIN1	+5V	5V power supply
PIN2	PT	Output Signal
PIN3	IR_ADJ	Input regulated optical power

3.2 Drive circuit

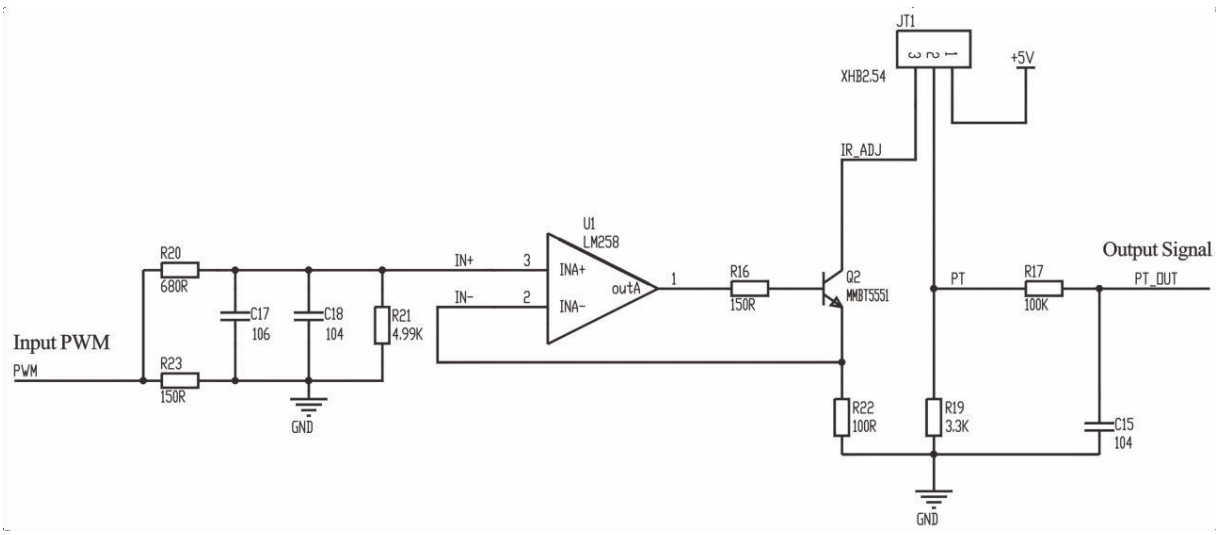


Figure 4. Recommended drive circuit

3.3 Output characteristic

Table 3. Output characteristic

	Minimum value	Typical values	Maximum value
Clear water(1)		3.8 V	
1000±30NTU milk	0.8 V		1.4 V

Note: (1) Turbidity of clear water is less than 1NTU.

3.4 Zero-point calibration and measurement method

3.4.1 Zero-point calibration

1. Place AZDM01 in clear water and avoid the interference of ambient light.
2. Adjust the duty cycle of input PWM until the output signal reaches 3.8 V and save the duty cycle for 0 NTU.

3.4.2 Measurement method

The turbidity of the medium is defined as f, which is the ratio of the voltage at 0 NTU (U_0 = 3.8V) and the real-time measured voltage (U_m). The lower the ratio factor, the higher the value of turbidity. The formula for the calculation of turbidity is:

$$f = U_m / U_0.$$

Table 4. The corresponding value range of f measured by AZDM01 at different value of turbidity

	0 NTU	500NTU	1000NTU	2000NTU	4000NTU
Upper limit value	100%	72%	51%	31%	15%
Average value	99.5%	64.3%	43.5%	23.6%	7.9%
Lower limit value	98%	57%	36%	16%	0.5%

4. Dimension

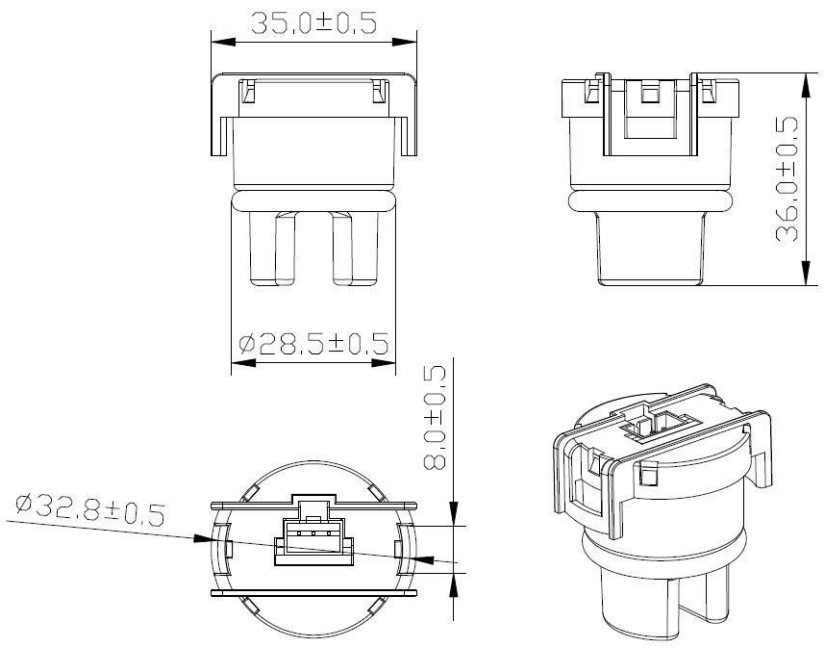


Figure 5. External dimensional drawing (unit: mm)

5. Installation Instruction

The mounting components AZDM01 are indicated in Figure 6.

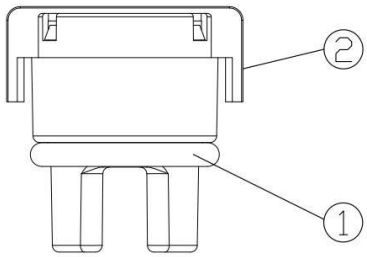


Figure 6. Mounting component

The AZDM01 should be mounted on a stable part of the machine by snap fastening, and the waterproof ring must be pressed tightly to achieve the ability of waterproof.

Table 5. Mounting component

NO.	Name	Description
①	Waterproof ring	Rubber gasket (black)
②	Snap	Plastic housing body

6. Note

6.1 Users must not disassemble the sensor to prevent irreversible damage.

6.2 Ensure that the water flow direction is perpendicular to the path of sensor light.

6.3 Please check and clean the sensor surface regularly, which is depending on the application scenario.

6.4 When cleaning the surface, please use materials, such as cotton gauze or sponge. Using hard material for cleaning or to wipe the translucent surface of the shell should be avoid to prevent damaging of sensor housing.

Warning and personal injury

Do not apply this product to safety protection devices or emergency stop equipment, and any other applications that may cause personal injury due to the product's failure. Do not use this product unless there is a special purpose or use authorization. Refer to the product data sheet and application guide before installing, handling, using or maintaining the product. Failure to follow this recommendation may result in death and serious personal injury.

The Company will not bear any compensation for personal injury and death arising therefrom, and will exempt the company's managers and employees, affiliated agents, distributors and any other claims that may arise therefrom, including: various costs, claims, lawyer fees, etc.

Quality assurance

Guangzhou Aosong Electronic Co., Ltd. provides the following quality guarantee to the direct buyers of its products (from the date of delivery), based on the technical specifications in the data manual of the products published by Aosong as the standard. Within the warranty period, if the product is confirmed to be defective, the company will provide free repair or replacement.

Description of warranty period

Accessories Category	Shelf Life
AZDM01 Turbidity sensor	6 months

The company is only responsible for products that are defective when used in applications that meet the technical conditions of the product. The company does not make any guarantees or written statements about the application of its products in those special applications. At the same time, the company does not make any promises about the reliability of its products when applied to products or circuits not provided by Aosong.

This manual may be changed at any time without notice.

Guangzhou Aosong Electronic Co., Ltd. reserves the right of final interpretation of this product.

Copyright ©2022, ASAIR ®

LEMBAR KONTROL
CAPSTONE DESIGN PROJECT

Tim Pelaksana	**Aprialdy Trimukti** NIM. 221180001	**Muhammad Arya Ismail** NIM. 221180003	PROGRAM STUDI **TEKNIK ELEKTRO** UNIVERSITAS MUHAMMADIYAH PAREPARE
Dosen Pembimbing	**1) Ir. A. Abd. Jabbar, MT.** Pembimbing I	**Alauddin Y, ST.,M.Kom** Pembimbing II	
Judul Proyek	Sistem Instrumentasi Pemantauan Kualitas Air pada Kolam Budidaya Ikan		

No	Tahapan	Kegiatan	Waktu Pelaksanaan	Tanda Tangan Pembimbing 1	Tanda Tangan Pembimbing 2	Tanda Tangan Instruktur
	Identifikasi Masalah	Pengumpulan Data & Analisis: 1) Studi Literatur standar kualitas air budidaya ikan 2) Observasi/Survei kondisi di Lokasi budidaya 3) Wawancara dengan pembudidaya ikan,ahli perikana dan teknisi kolam 4) Mengidentifikasi kendala yang dihadapi pembudidaya	Senin, 04 - 08 November 2024			
		Identifikasi Parameter Penting: 1) Range normal parameter air(pH, Suhu, DO, Kekeruhan) 2) Spesifikasi sensor yang dibutuhkan 3) Batasan nilai untuk setiap parameter 4) Frekuensi pengukuran yang dibutuhkan	Kamis 28/11/2024			
	Spesifikasi Sistem	Penyusunan Spesifikasi Teknis: 1) Hardware requirements 2) Software requirements 3) Performance requirements 4) Environmental requirements	Senin 25/11/2024			
		Pembuatan Timeline: 1) Pengukuran Parameter Kualitas air 2) Kinerja Produk : Range Pengukuran suhu, ph, DO dan turbidity.				

Catatan: Lembar Kontrol ini dicetak timbal balik di atas kertas karton berwarna merah muda ukuran A4

Dipindai dengan CamScanner

No	Tahapan	Kegiatan	Waktu Pelaksanaan	Tanda Tangan Pembimbing 1	Tanda Tangan Pembimbing 2	Tanda Tangan Instruktur
		3) Interval Pembacaan 5-15 menit 4) Peyimpanan data Karakteristik Produk 1) Dimensi unit sensor 2) Dimensi control box 3) Material housing 4) Sumber Daya 12 v DC, backup Battery Rencana Anggaran 1) Mikrokontroler dan PCB 2) Sensor Suhu, Ph, DO, turbidity 3) Komponen Elektronik Pendukung 4) Housing dan Kabel 5) Total estimasi (sesuai harga pasar)				
	Perancangan Sistem	Perangcangan Hadware 1. Skematik Rangkaian Sensor 2. Rangkaian Pengkondisi Sinyal 3. Sistem Power Supply 4. Desain PCB untuk control unit 5. Wiring Diagram Keseluruhan				
		Perangcangan software 1. Program pembacaan sensor meliputi kalibrasi, konversi data, filter digital, data logging 2. Interface pengguna seperti Dashboard monitoring,grafik trend parameter, sistem notifikasi.				
		Desain Sistem Keseluruhan 1. Arsiktektur Sistem ; Modul Sensor, Control Unit, Display Unit, Data logger 2. Alur data dan komunikasi antar modul				
		Desain Mekanik 1. Housing tahan air untuk sensor 2. Mounting sistem dikolam 3. Desain control box 4. Sistem kabel dan konektor tahan air				

Catatan: Lembar Kontrol ini dicetak timbal balik di atas kertas karton berwarna merah muda ukuran A4

Dipindai dengan CamScanner

No	Tahapan	Kegiatan	Waktu Pelaksanaan	Tanda Tangan Pembimbing 1	Tanda Tangan Pembimbing 2	Tanda Tangan Instruktur
4	Analisib Pengujian Sistem	Menentukan jenis data yang akan dikumpulkan dalam pengujian				
		Menunjukkan data hasil pengujian / pengambilan data dalam bentuk tabel				
		Menampilkan hasil penyajian data dalam bentuk tabel dan/atau grafik dalam dokumen				
5	Laporan dan Dokumentasi	Dokumentasi tiap Pelaksanaan Kegiatan				
		Penyusunan laporan hasil rangcangan perangkat keras, perangkat lunak dan analisis data sesuai format laporan.				

Catatan: Lembar Kontrol ini dicetak timbal balik di atas kertas karton berwarna merah muda ukuran A4

Dipindai dengan CamScanner

No	Tahapan	Kegiatan	Waktu Pelaksanaan	Tanda Tangan Pembimbing 1	Tanda Tangan Pembimbing 2	Tanda Tangan Instruktur
	Analisis Data Sub-Sistem dan Sistem Keseluruhan	Menampilkan analisis data pada setiap hasil pengujian pada dokumen Skripsi	19/2/2024			

Catatan: Lembar Kontrol ini dicetak timbal balik di atas kertas karton berwarna merah muda ukuran A4

Dipindai dengan CamScanner

LOGBOOK CAPSTONE DESIGN
PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS MUHAMMADIYAH PAREPARE

GAMBARAN UMUM CAPSTONE DESIGN PROJECT

Judul Capstone Design Project	**Sistem Instrumentasi Pemantauan Kualitas Air pada Kolam Budidaya Ikan Berbasis Mikrokontroler**		
Permasalahan:	- Penentuan lokasi observasi - parameter kualitas air yang dipakai	Solusi:	
Lokasi :			
Mahasiswa 1:	**APRIALDY TRIMUKTI**	NIM:	221180001
Mahasiswa 2:	**Muh. Arya Ismail**	NIM:	221180003
Dosen Pembimbing 1	**Ir. A. Abd. Jabbar M.T.**	NBM / NIDN	1030917
Dosen Pembimbing 2	**Alaudin Yunus S.T., M.Kom**	NBM / NIDN	1190376
Spesifikasi Desain (*Design Specification*):			
Durasi Proyek	Tanggal mulai: 5 November 2024	Tanggal rencana penyelesaian: 5 Februari 2025 Tanggal realisasi penyelesaian:	

Dipindai dengan CamScanner

LOGBOOK CAPSTONE DESIGN
PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS MUHAMMADIYAH PAREPARE

URAIAN PROGRES KEGIATAN

Waktu Pelaksanaan	Item Kegiatan dan Hasil	Bukti Kegiatan	Keterangan	Tanda Tangan Pembimbing
Senin, 4 Nov - 2024	Kami berdua melakukan Study literatur, dimana kami membuat resume dari 5 jurnal yang telah kita dapatkan. dan membuat tabel yang berisikan Parameter, Sensor, Standar nilai baku, dampak, dan harga.		Kami membahas tentang parameter yang akan kami pakai. Sekaligus membahas tentang lokasi wawancara	1. 2.
Selasa, 5 Nov 2024	kami melakukan bimbingan Pertama dengan dospem 1 yakni pak Abd. Jabbar kemudian dilanjutkan bimbingan dengan dospem 2 yakni pak Alaudin via online. dan memeriksa lembar kontrol berisi tahapan yang akan dikerjakan.		kami membahas tentang Parameter yang akan kami gunakan, membahas lokasi wawancara, menerima file berisi lembar kontrol dan tahapan yang akan dikerjakan.	1. 2.
Jumat, 8 Nov 2024	kami bimbingan dengan dengan dospem 2 yakni pak Alaudin mengenai Study literatur dan pertanyaan untuk wawancara yg telah dibuat sebelumnya.		kami menyiapkan Pertanyaan wawancara dan didapatkan 7 Pertanyaan wawancara yang akan dipakai.	1.

Dipindai dengan CamScanner

LOGBOOK CAPSTONE DESIGN
PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS MUHAMMADIYAH PAREPARE

Waktu Pelaksanaan	Item Kegiatan dan Hasil	Bukti Kegiatan	Keterangan	Tanda Tangan Pembimbing
Selasa, 9 Nov 2024	Kami berdua menuju lokasi wawancara di tambak ikan milik pak Nawir yang berlokasi di Suppa, Pinrang. Dan kami melakukan wawancara dengan pak Nawir.			1. 2.
Rabu, 13 Nov 2024	Kami berdua meminta dan mengesahkan surat izin penelitian. Setelah itu kami melanjutkan bimbingan dengan pak Jabbar, membahas hasil wawancara kami.			1. 2.
Sabtu, 16 Nov 2024	Kami berdua melakukan bimbingan kedua dengan pak Akiudin, serta mengerjakan logbook kami.			1. 2.

Dipindai dengan CamScanner

LOGBOOK CAPSTONE DESIGN
PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS MUHAMMADIYAH PAREPARE

URAIAN PROGRES KEGIATAN

Waktu Pelaksanaan	Item Kegiatan dan Hasil	Bukti Kegiatan	Keterangan	Tanda Tangan Pembimbing
Kamis, 21 Nov 2024	Kami berdua melakukan konsultasi EU100 mengenai rumusan masalah, latar belakang dan karakteristik solusi.			1.
Kamis, 28 Nov 2024	Kami berdua melakukan bimbingan dengan pembimbing 2 pak Alauddin mengenai koreksi pada rumusan masalah			1. 2.
Kamis, 05 okt 2024	Kami berdua melakukan bimbingan dengan pembimbing 1 pak Jabbar mengenai EU200			

Dipindai dengan CamScanner

LOGBOOK CAPSTONE DESIGN
PROGRAM STUDI TEKNIK ELEKTRO
UNIVERSITAS MUHAMMADIYAH PAREPARE

URAIAN PROGRES KEGIATAN

Waktu Pelaksanaan	Item Kegiatan dan Hasil	Bukti Kegiatan	Keterangan	Tanda Tangan Pembimbing
Sabtu 7/12/2024	Kami berdua berkunjung kerumah pak Ala untuk bimbingan mengenai EO200			7/12/2024
Senin, 20/01/2025	Kami berdua melakukan bimbingan dengan pak Ala mengenai lengkah selanjutnya yakni perancangan dan pengkalibrasian.			20/1/2025.
Senin, 19/05/25	Kami berdua melakukan konsultasi terkait program dan mekanik Alat secara keseluruhan			
Sabtu, 24/05/25	Kami berdua melakukan konsultasi terkait program dan mekanik Alat secara keseluruhan. dirumah pak Ala.			24/5/2025

Dipindai dengan CamScanner