LAMPIRAN

KODE PROGRAM ARDUINO UNO

```
<?php

namespace App\Controllers;

use App\Controllers\BaseController;
use App\Models\User;

class Auth extends BaseController
{
    public function __construct() {
        $this->tb_user = new User();
    }

    public function index() {
        return view('auth/login');
    }

    public function login() {
        $email = $this->request->getPost('email');
        $pass  = $this->request->getPost('password');
        $res   = $this->tb_user->where('email', $email)->get()->getRow();
        if (empty($res)) return redirect()->to(base_url('/auth'))->with('error',
'Autentikasi Gagal.!');
        if (!password_verify($pass, $res->password)) return redirect()-
>to(base_url('/auth'))->with('error', password_hash($pass,
PASSWORD_DEFAULT).'Password Salah.!');
        $data = array(
            'email' => $res->email,
            'isLoggedIn' => TRUE
        );
        session()->set($data);
        return redirect()->to(base_url());
    }

    public function logout() {
        $data = array('email', 'isLoggedIn');
        session()->remove($data);
        return redirect()->to(base_url());
    }

}
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <TroykaDHT.h>
#include <ArduinoJson.h>

#define LED       D4
#define LEDG      D5

// Inisialisasi Pin Input
#define SOILPIN   A0
#define DHT21PIN  D3

// Inisialisasi Pin Output
#define NOZZELPIN D6
#define BLOWER    D7
#define VALVE     D8

DHT dht(DHT21PIN, DHT21);

const char* ssid      = "Sinau Tech";
const char* password  = "HURUFKECIL@321";

const char* serverWrite = "http://192.168.18.138/singgihapp/public/sensor/write";
const char* serverRead = "http://192.168.18.138/singgihapp/public/sensor/read";

String email = "singgihrd7023@gmail.com";
String pass  = "12345678";
String payload = "", suhu, moist, soil;

int cycle = 0;

void setup() {
  pinMode(NOZZELPIN, OUTPUT);
  pinMode(BLOWER, OUTPUT);
  pinMode(VALVE, OUTPUT);
  pinMode(LED, OUTPUT);
  pinMode(LEDG, OUTPUT);
  digitalWrite(NOZZELPIN, HIGH);
  digitalWrite(BLOWER, LOW);
  digitalWrite(VALVE, LOW);
  digitalWrite(LEDG, LOW);
  dht.begin();
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    digitalWrite(LED, !digitalRead(LED));
    delay(50);
  }
  digitalWrite(LED, LOW);
  digitalWrite(LEDG, HIGH);
  Serial.begin(115200);
}
```

```
void loop() {
  getDHT();
  getSoil();
  if (cycle == 20) {
    setData();
    cycle = 0;
  }
  readData();
  digitalWrite(LED, !digitalRead(LED));
  digitalWrite(LEDG, !digitalRead(LEDG));
  Serial.print("Cycle => ");
  Serial.println(cycle);
  cycle++;
  delay(1000);
}
void setData() {
  Serial.println("\n\n===== SEND DATA =====");
  WiFiClient client;
  HTTPClient http;
  http.begin(client, serverWrite);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");
  String httpRequestData = "email=" + email + "&password=" + pass + "&suhu="
+ suhu + "&moist=" + moist + "&soil=" + soil;
  int httpResponseCode = http.POST(httpRequestData);
  payload = http.getString();
  Serial.print("Response Server    : ");
  Serial.println(payload);
  Serial.print("HTTP Response code : ");
  Serial.println(httpResponseCode);
  http.end();
}
void readData() {
  Serial.println("\n\n===== READ DATA =====");
  WiFiClient client;
  HTTPClient http;
  http.begin(client, serverRead);
  http.addHeader("Content-Type", "application/x-www-form-urlencoded");
  String httpRequestData = "email=" + email + "&password=" + pass;
  int httpResponseCode = http.POST(httpRequestData);
  payload = http.getString();
  Serial.print("Response Server    : ");
  Serial.println(payload);
  Serial.print("HTTP Response code : ");
  Serial.println(httpResponseCode);
  http.end();
  if (httpResponseCode == 200) control(payload);
}
void control(String payload) {
  DynamicJsonDocument doc(1024);
  deserializeJson(doc, payload);
  JsonObject obj = doc.as<JsonObject>();
  String modeNode = obj["mode"].as<String>();
  String nozzelControl = obj["nozzel"].as<String>();
  String dripControl = obj["drip"].as<String>();
  String blowerControl = obj["blower"].as<String>();
  Serial.println("Mode => " + modeNode);
  Serial.println("Set Nozzel => " + nozzelControl);
  Serial.println("Set Drip   => " + dripControl);
  Serial.println("Set Blower => " + blowerControl);

  if (modeNode == "auto") {
  }
  if (nozzelControl == "on") digitalWrite(NOZZELPIN, LOW);
  else digitalWrite(NOZZELPIN, HIGH);
  if (dripControl == "on") digitalWrite(VALVE, HIGH);
  else digitalWrite(VALVE, LOW);
  if (blowerControl == "on") digitalWrite(BLOWER, HIGH);
  else digitalWrite(BLOWER, LOW);
}
void getDHT() {
  dht.read();
  switch (dht.getState()) {
    case DHT_OK:
      suhu = (String)dht.getTemperatureC();
      moist = (String)dht.getHumidity();
      break;
    case DHT_ERROR_CHECKSUM:
      break;
    case DHT_ERROR_TIMEOUT:
      break;
    case DHT_ERROR_NO_REPLY:
      break;
  }
}
void getSoil() {
  int val = map(analogRead(SOILPIN), 1023, 0, 0, 100);
  soil = (String)val;
}
```

KODE PROGRAM WEBSITE

```
<?php

namespace App\Controllers;

use App\Models\Nodesensor;
use App\Models\Control;
use App\Models\User;
use App\Models\Jadwal;

class Home extends BaseController
{
    public function __construct() {
        $this->tb_node = new Nodesensor();
        $this->tb_control = new Control();
        $this->tb_user = new User();
        $this->tb_jadwal = new Jadwal();
    }
    public function index() {
        $data = $this->tb_node->orderBy('timestamp', 'DESC')->get()->getRow();
        $data = ['temp' => $data->suhu, 'hum' => $data->kelembaban, 'soil' =>
$data->tanah];
        return view('dashboard/index', $data);
    }
    public function setting() {
        $control = $this->tb_control->get()->getRow();
        $jadwal = $this->tb_jadwal->get()->getResult();
        $data = [
            'jadwal' => $jadwal,
            'drip' => $control->drip,
            'nozzel' => $control->nozzel,
            'blower' => $control->blower,
            'mode' => $control->mode
        ];
        return view('setting/index', $data);
    }
    public function graph() {
        $data = $this->tb_node->orderBy('timestamp DESC')->limit(7)->get()->getResult();
        return $this->response->setJSON($data);
    }
    public function realtime() {
        $data = $this->tb_node->orderBy('timestamp', 'DESC')->get()->getRow();
        $data = ['temp' => $data->suhu, 'hum' => $data->kelembaban, 'soil' => $data->tanah];
        return $this->response->setJSON($data);
    }
    public function dripper($val = null) {
        $this->tb_control->set('drip', $val)->update();
        return redirect()->to(base_url('/home/setting'));
    }
    public function nozzel($val = null) {
        $this->tb_control->set('nozzel', $val)->update();
        return redirect()->to(base_url('/home/setting'));
    }
    public function blower($val = null) {
        $this->tb_control->set('blower', $val)->update();
        return redirect()->to(base_url('/home/setting'));
    }
    public function mode($val = null) {
        $this->tb_control->set('mode', $val)->update();
        return redirect()->to(base_url('/home/setting'));
    }
}
```

```
<?php

namespace App\Controllers;

use App\Controllers\BaseController;
use App\Models\Nodesensor;
use App\Models\Control;
use App\Models\User;

class Sensor extends BaseController
{
    public function __construct() {
        $this->tb_node = new Nodesensor();
        $this->tb_control = new Control();
        $this->tb_user = new User();
    }
    public function index() {
        $email = $this->request->getPost('email');
        $password = $this->request->getPost('password');
        $user = $this->tb_user->where('email', $email)->get()->getRow();
        if (empty($user)) return $this->response->setJSON('Akses tidak diizinkan.!');
        if (!(password_verify($password, $user->password))) return $this->response->setJSON('Autentikasi Gagal.!');
        return $this->response->setJSON($_POST);
    }
    public function write() {
        $email = $this->request->getPost('email');
        $password = $this->request->getPost('password');
        $user = $this->tb_user->where('email', $email)->get()->getRow();
        if (empty($user)) return $this->response->setJSON('Akses tidak diizinkan.!');
        if (!(password_verify($password, $user->password))) return $this->response-
>setJSON('Autentikasi Gagal.!');
        $temp = $this->request->getPost('suhu');
        $moist = $this->request->getPost('moist');
        $soil = $this->request->getPost('soil');
        if (empty($temp)) return $this->response->setJSON('Tidak ada data
Temperature');
        if (empty($moist)) return $this->response->setJSON('Tidak ada data
Humidity');
        if (empty($soil)) return $this->response->setJSON('Tidak ada data Soil
Moisture');
        $data = [
            'suhu' => $temp,
            'kelembaban' => $moist,
            'tanah' => $soil
        ];
        $this->tb_node->insert($data);
        return $this->response->setJSON('Sukses Menambahkan Data');
    }
    public function read() {
        $email = $this->request->getVar('email');
        $password = $this->request->getVar('password');
        $user = $this->tb_user->where('email', $email)->get()->getRow();
        if (empty($user)) return $this->response->setJSON('Akses tidak diizinkan.!');
        if (!password_verify($password, $user->password)) return $this->response-
>setJSON('Autentikasi Gagal.!');
        $data = $this->tb_control->get()->getRow();
        return $this->response->setJSON(json_encode($data));
    }
}
<?php

namespace App\Models;
```

```
use CodeIgniter\Model;

class Control extends Model
{
    protected $table = 'tb_control';
    protected $allowedFields = [
        'id',
        'blower',
        'drip',
        'nozzel',
        'mode'
    ];
}
<?php

namespace App\Models;

use CodeIgniter\Model;

class Nodesensor extends Model
{
    protected $table = 'tb_nodesensor';
    protected $allowedFields = [
        'id',
        'suhu',
        'kelembaban',
        'tanah',
        'timestamp'
    ];
}
<?php
```

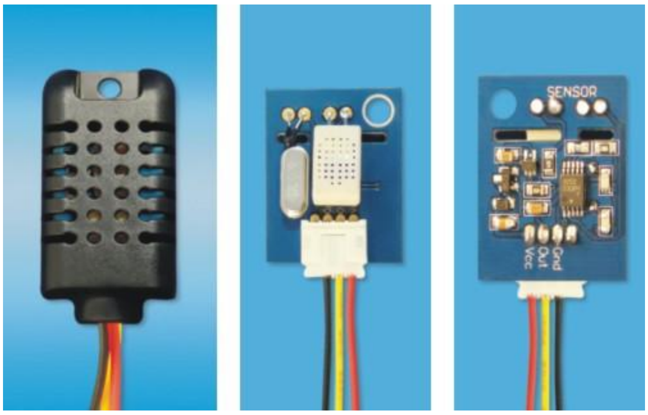
```
namespace App\Models;

use CodeIgniter\Model;

class User extends Model
{
    protected $table = 'tb_user';
    protected $allowedFields = [
        'id',
        'username',
        'email',
        'password',
    ];
}
```

DATA SHEET

AOSONG

Temperature and humidity module

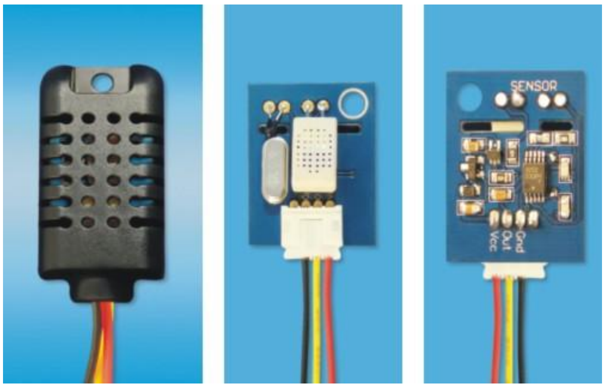
AM2301 Product Manual



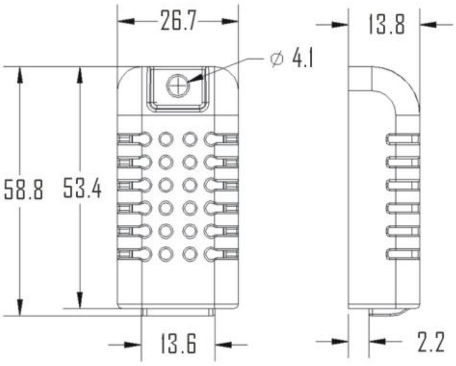
www.aosong.com

1、Product Overview

AM2301 capacitive humidity sensing digital temperature and humidity module is the one that contains the compound has been calibrated digital signal output of the temperature and humidity sensor. Application of a dedicated digital modules collection technology and the temperature and humidity sensing technology, to ensure that the product has high reliability and excellent long−term stability. The sensor includes a capacitive sensor wet components and a high−precision temperature measurement devices, and connected with a high−performance 8−bit microcontroller. The product has excellent quality, fast response, strong anti−jamming capability, and high cost. Each sensor is extremely accurate humidity calibration chamber calibration. The form of procedures, the calibration coefficients stored in the microcontroller, the sensor within the processing of the heartbeat to call these calibration coefficients. Standard single−bus interface, system integration quick and easy. Small size, low power consumption, signal transmission distance up to 20 meters, making it the best choice of all kinds of applications and even the most demanding applications. Products for the 3−lead (single−bus interface) connection convenience. Special packages according to user needs.



Physical map



Dimensions (unit: mm)

2、Applications

HVAC, dehumidifier, testing and inspection equipment, consumer goods, automotive, automatic control, data loggers, home appliances, humidity regulator, medical, weather stations, and other humidity measurement and control and so on.

3、Features

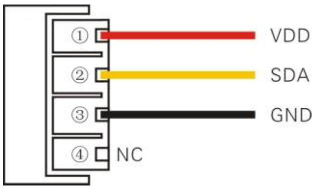
Ultra−low power, the transmission distance, fully automated calibration, the use of capacitive humidity sensor, completely interchangeable, standard digital single−bus output, excellent long−term stability, high accuracy temperature measurement devices.

4、The definition of single−bus interface

4.1 AM2301 Pin assignments

Table 1： AM2301 Pin assignments

Pin	Color	Name	Description
1	Red	VDD	Power (3.3V−5.2V)
2	Yellow	SDA	Serial data，Dual−port
3	Black	GND	Ground
4		NC	Empty



PIC1： AM2301 Pin Assignment

4.2 Power supply pins（VDD GND）

AM2301 supply voltage range 3.3V － 5.2V, recommended supply voltage is 5V.

4.3 Serial data（SDA）

SDA pin is tri structure for reading, writing sensor data. Specific communication timing, see the detailed description of the communication protocol.

5、Sensor performance

5.1 Relative humidity

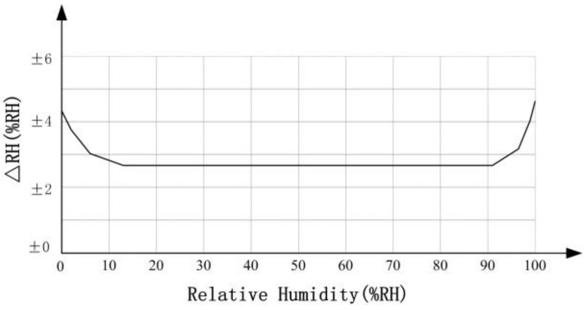
Table 2： AM2301 Relative humidity performance table

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		%RH
Range		0		99.9	%RH
Accuracy [1]	25℃		±3		%RH
Repeatability			±1		%RH
Exchange		Completely interchangeable			
Response [2]	1/e(63%)		<6		S
Sluggish			±0.3		%RH
Drift [3]	Typical		<0.5		%RH/yr

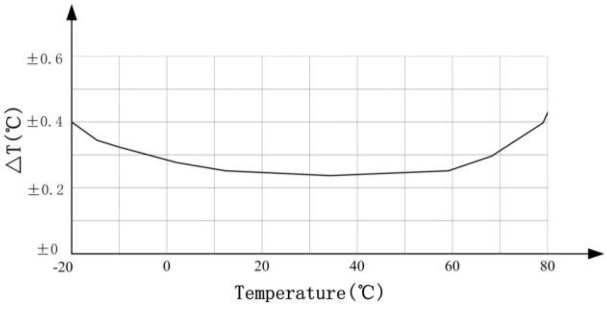
5.2 Temperature

Table 3： AM2301 Relative temperature performance

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		℃
			16		bit
Accuracy			±0.3	±1	℃
Range		−40		80	℃
Repeat			±0.2		℃
Exchange		Completely interchangeable			
Response	1/e(63%)		<10		S
Drift			±0.3		℃/yr



Pic2: At25℃ The error of relative humidity



Pic3： The maximum temperature error

6、Electrical Characteristics

Electrical characteristics, such as energy consumption, high, low, input, output voltage, depending on the power supply. Table 4 details the electrical characteristics of the AM2301, if not identified, said supply voltage of 5V. To get the best results with the sensor, please design strictly in accordance with the conditions of design in Table 4.

Table 4： AM2301 DC Characteristics

Parameter	Condition	min	typ	max	Unit
Voltage		3.3	5	5.2	V
Power consumption [4]	Dormancy	10	15		μA
	Measuring		500		μA
	Average		300		μA
Low level output voltage	I_{OL}[5]	0		300	mV
High output voltage	Rp<25 kΩ	90%		100%	VDD
Low input voltage	Decline	0		30%	VDD
Input High Voltage	Rise	70%		100%	VDD
Rpu[6]	VDD = 5V VIN = VSS	30	45	60	kΩ
Output current	turn on		8		mA
	turn off	10	20		μA
Sampling period		2			S

7、Single–bus communication（ONE–WIRE）

7.1 Typical circuits for single bus

Microprocessor and AM2301 connection typical application circuit is shown in Figure 4. Single bus communication mode, pull the SDA microprocessor I / O port is connected.

Special instructions of the single–bus communication：

1. Typical application circuit recommended in the short cable length of 30 meters on the 5.1K pull–up resistor pullup resistor according to the actual situation of lower than 30 m.
2. With 3.3V supply voltage, cable length shall not be greater than 100cm. Otherwise, the line voltage drop will lead to the sensor power supply, resulting in measurement error.
3. Read the sensor minimum time interval for the 2S; read interval is less than 2S, may cause the temperature and humidity are not allowed or communication is unsuccessful, etc..
4. Temperature and humidity values are each read out the results of the last measurement For real–time data that need continuous read twice, we recommend repeatedly to read sensors, and each read sensor interval is greater than 2 seconds to obtain accuratethe data.

[1] the accuracy of the factory inspection, the sensor 25 ° C and 5V, the accuracy specification of test conditions, it does not include hysteresis and nonlinearity, and is only suitable for non–condensing environment.

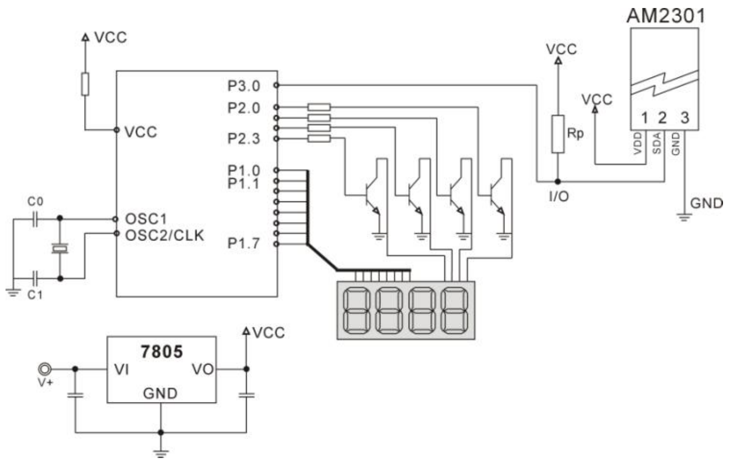
[2] to achieve an order of 63% of the time required under the conditions of 25 ℃ and 1m / s airflow.

[3] in the volatile organic compounds, the values may be higher. See the manual application to store information.

[4] this value at VDD = 5.0V when the temperature is 25 ℃, 2S / time, under the conditions of the average.

[5] low output current.

[6] that the pull–up resistor.



Pic4： AM2301 Typical circuits for single bus

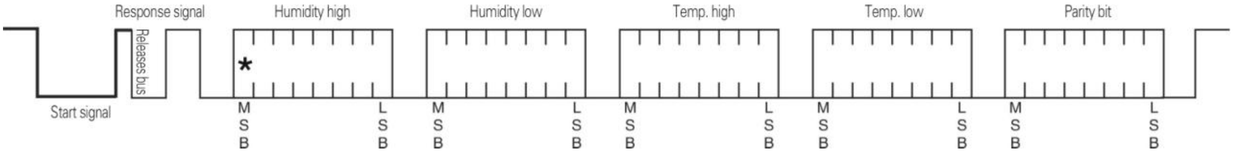
7.2、Single−bus communication protocol

◎Single bus Description

AM2301 device uses a simplified single−bus communication. Single bus that only one data line, data exchange system, controlled by the data line to complete. Equipment (microprocessor) through an open−drain or tri−state port connected to the data line to allow the device does not send data to release the bus, while other devices use the bus; single bus usually require an external about 5.1kΩ pull−up resistor, so when the bus is idle, its status is high. Because they are the master−slave structure, only the host calls the sensor, the sensor will answer, so the hosts to access the sensor must strictly follow the sequence of single bus, if there is a sequence of confusion, the sensor will not respond to the host.

◎Single bus to send data definition

SDA For communication and synchronization between the microprocessor and the AM2301, single−bus data format, a transmission of 40 data, the high first−out. Specific communication timing shown in Figure 5, the communication format is depicted in Table 5.



Pic5： AM2301 Single−bus communication protocol

Table 5：AM2301 Communication format specifier

Name	Single−bus format definition
Start signal	Microprocessor data bus (SDA) to bring down a period of time (at least 800μ s) [1] notify the sensor to prepare the data.
Response signal	Sensor data bus (SDA) is pulled down to 80μ s, followed by high−80μ s response to host the start signal.
Data format	Host the start signal is received, the sensor one−time string from the data bus (SDA) 40 data, the high first−out.
Humidity	Humidity resolution of 16Bit, the previous high; humidity sensor string value is 10 times the actual humidity values.
Temp.	Temperature resolution of 16Bit, the previous high; temperature sensor string value is 10 times the actual temperature value; The temperature is the highest bit (Bit15) is equal to 1 indicates a negative temperature, the temperature is the highest bit (Bit15) is equal to 0 indicates a positive temperature; Temperature in addition to the most significant bit (Bit14 ~ bit 0) temperature values.
Parity bit	Parity bit = humidity high + humidity low + temperature high + temperature low

◎**Single−bus data calculation example**

Example 1：40 Data received：

<u>0000 0010</u>	<u>1001 0010</u>	<u>0000 0001</u>	<u>0000 1101</u>	<u>1010 0010</u>
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

Calculate：

0000 0010+1001 0010 +0000 0001+0000 1101= 1010 0010（Parity bit）

Received data is correct：

humidity：0000 0010 1001 0010 = 0292H (Hexadecimal)= 2×256 + 9×16 + 2 = 658

=> Humidity = 65.8%RH

Temp.：0000 0001 0000 1101 = 10DH(Hexadecimal) = 1×256 + 0×16 + 13 = 269

=> Temp.= 26.9℃

◎**Special Instructions**：

When the temperature is below 0 ℃, the highest position of the temperature data.

Example： −10.1 ℃ Expressed as 1 000 0000 0110 0101

Temp.：0000 0000 0110 0101 = 0065H(Hexadecimal)= 6×16 +5 = 101

=> Temp. = −10.1℃

Example 2：40 received data：

<u>0000 0010</u>	<u>1001 0010</u>	<u>0000 0001</u>	<u>0000 1101</u>	<u>1011 0010</u>
High humidity 8	Low humidity 8	High temp. 8	Low temp. 8	Parity bit

Calculate：

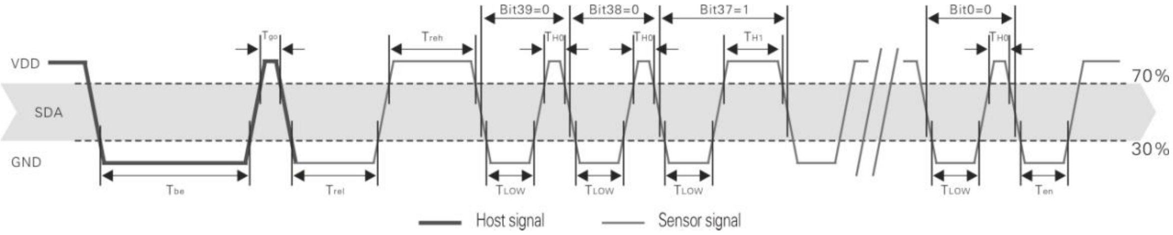
0000 0010+1001 0010 +0000 0001+0000 1101= 1010 0010 ≠ <u>1011 0010</u> （Validation error）

The received data is not correct, give up, to re−receive data.

7.3 Single—bus communication timing

User host (MCU) to send a start signal (data bus SDA line low for at least 800μ s) after AM2301 from Sleep mode conversion to high—speed mode.The host began to signal the end of the AM2301 send a response signal sent from the data bus SDA serial 40Bit's data, sends the byte high; data sent is followed by: Humidity high、Humidity low、Temperature high、Temperature low、Parity bit，Send data to the end of trigger information collection, the collection end of the sensor is automatically transferred to the sleep mode, the advent until the next communication.

Detailed timing signal characteristics in Table 6，Single—bus communication timing diagram Pic 6：



Pic 6：AM2301 Single—bus communication timing

Note: the temperature and humidity data read by the host from the AM2301 is always the last measured value, such as the two measurement interval is very long, continuous read twice to the second value of real—time temperature and humidity values, while two readtake minimum time interval be 2S.

Table 6: Single bus signal characteristics

Symbol	Parameter	min	typ	max	Unit
T_{be}	Host the start signal down time	0.8	1	20	mS
T_{go}	Bus master has released time	20	30	200	μS
T_{rel}	Response to low time	75	80	85	μS
T_{reh}	In response to high time	75	80	85	μS
T_{LOW}	Signal "0", "1" low time	48	50	55	μS
T_{H0}	Signal "0" high time	22	26	30	μS
T_{H1}	Signal "1" high time	68	70	75	μS
T_{en}	Sensor to release the bus time	45	50	55	μS

Note： To ensure the accurate communication of the sensor, the read signal, in strict accordance with the design parameters and timing in Table 6 and Figure 6.

7.4 Peripherals read step example

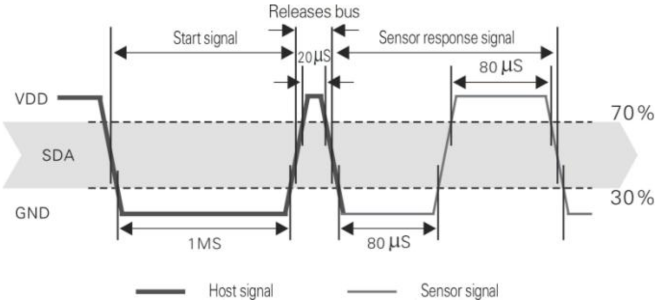
Communication between the host and the sensor can read data through the following three steps to complete.

Step 1

AM2301 have to wait for the power (on AM2301 power 2S crossed the unstable state, the device can not send any instructions to read during this period), the test environment temperature and humidity data, and record data, since the sensor into a sleep state automatically. AM2301 The SDA data line from the previous pull—up resistor pulled up is always high, the AM2301 the SDA pin is in input state, the time detection of external signal.

Step 2

Microprocessor I/O set to output, while output low, and low hold time can not be less than 800us, typical values are down 1MS, then the microprocessor I/O is set to input state, the release of the bus, due to the pull−up resistor, the microprocessor I/O AM2301 the SDA data line also will be high, the bus master has released the AM2301 send a response signal, that is, the output 80 microseconds low as the response signal, tightthen output high of 80 microseconds notice peripheral is ready to receive data signal transmission as shown to Pic7 :



Pic7: Single bus decomposition of the timing diagram

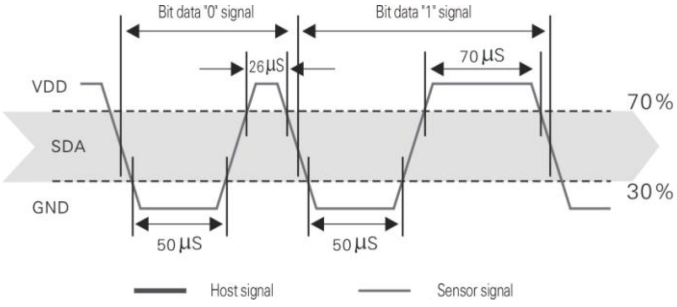
Step 3

AM2301 sending the response, followed by the data bus SDA continuous serial output 40 data, the microprocessor receives 40 data I/O level changes.

Bit data "0" format: 26−28 microseconds 50 microseconds low plus high;

Bit data "1" format: the high level of low plus, 50 microseconds to 70 microseconds;

Bit data "0" bit data "1" format signal shown to pic 8:

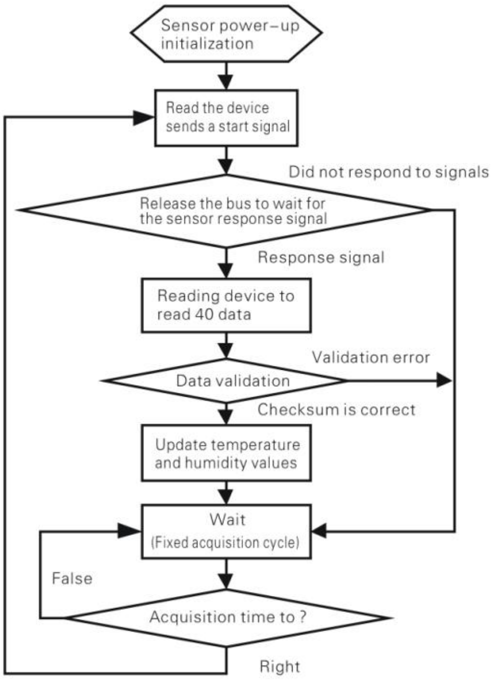


Pic 8: The single bus break down the timing diagram

AM2301 data bus SDA output 40 data continue to output the low 50 microseconds into the input state, followed by pull−up resistor goes high. AM2301 internal re−test environmental temperature and humidity data, and record the data, the end of the test records, the microcontroller automatically into hibernation. Microcontroller only after receipt of the start signal of the host wake−up sensor, into the working state.

7.5 Peripheral to read flow chart

AM2301 sensor read single bus flow chart diagram shown in Figure 9, we also provide the C51 read the code examples, customers need to download, please visit our website (www.aosong.com) related to downloadthis manual does not provide the code description.


Sensor power-up initialization
Read the device sends a start signal
Did not respond to signals
Release the bus to wait for the sensor response signal
Response signal
Reading device to read 40 data
Validation error
Data validation
Checksum is correct
Update temperature and humidity values
Wait (Fixed acquisition cycle)
False
Acquisition time to ?
Right

Pic9：Single-bus to read the flow chart

8、Application of information

1. Work and storage conditions

Outside the sensor the proposed scope of work may lead to temporary drift of the signal up to 300%RH. Return to normal working conditions, sensor calibration status will slowly toward recovery. To speed up the recovery process may refer to "resume processing". Prolonged use of non-normal operating conditions, will accelerate the aging of the product.

Avoid placing the components on the long-term condensation and dry environment, as well as the following environment.

A, salt spray

B, acidic or oxidizing gases such as sulfur dioxide, hydrochloric acid

Recommended storage environment

Temperature: 10 ~ 40℃ Humidity: 60%RH or less

2. The impact of exposure to chemicals

The capacitive humidity sensor has a layer by chemical vapor interference, the proliferation of chemicals in the sensing layer may lead to drift and decreased sensitivity of the measured values. In a pure environment, contaminants will slowly be released. Resume processing as described below will accelerate this process. The high concentration of chemical pollution (such as ethanol) will lead to the complete damage of the sensitive layer of the sensor.

3. The temperature influence

Relative humidity of the gas to a large extent dependent on temperature. Therefore, in the measurement of humidity, should be to ensure that the work of the humidity sensor at the same temperature. With the release of heat of electronic components share a printed circuit board, the installation should be as far as possible the sensor away from the electronic components and mounted below the heat source, while maintaining good ventilation of the enclosure. To reduce the thermal conductivity sensor and printed circuit board copper plating should be the smallest possible, and leaving a gap between the two.

4. Light impact

Prolonged exposure to sunlight or strong ultraviolet radiation, and degrade performance.

5. Resume processing

Placed under extreme working conditions or chemical vapor sensor, which allows it to return to the status of calibration by the following handler. Maintain two hours in the humidity conditions of 45℃ and <10% RH (dry); followed by 20−30℃ and> 70%RH humidity conditions to maintain more than five hours.

6. Wiring precautions

The quality of the signal wire will affect the quality of the voltage output, it is recommended to use high quality shielded cable.

7. Welding information

Manual welding, in the maximum temperature of 300℃ under the conditions of contact time shall be less than 3 seconds.

8. Product upgrades

Details, please the consultation Aosong electronics department.

9、The license agreement

Without the prior written permission of the copyright holder, shall not in any form or by any means, electronic or mechanical (including photocopying), copy any part of this manual, nor shall its contents be communicated to a third party. The contents are subject to change without notice.

The Company and third parties have ownership of the software, the user may use only signed a contract or software license.

10、Warnings and personal injury

This product is not applied to the safety or emergency stop devices, as well as the failure of the product may result in injury to any other application, unless a particular purpose or use authorized. Installation, handling, use or maintenance of the product refer to product data sheets and application notes. Failure to comply with this recommendation may result in death and serious personal injury. The Company will bear all damages resulting personal injury or death, and waive any claims that the resulting subsidiary company managers and employees and agents, distributors, etc. that may arise, including: a variety of costs, compensation costs, attorneys' fees, and so on.

11、Quality Assurance

The company and its direct purchaser of the product quality guarantee period of three months (from the date of delivery). Publishes the technical specifications of the product data sheet shall prevail. Within the warranty period, the product was confirmed that the quality is really defective, the company will provide free repair or replacement. The user must satisfy the following conditions:

① The product is found defective within 14 days written notice to the Company;

② The product shall be paid by mail back to the company;

③ The product should be within the warranty period.

The Company is only responsible for those used in the occasion of the technical condition of the product defective product. Without any guarantee, warranty or written statement of its products used in special applications. Company for its products applied to the reliability of the product or circuit does not make any commitment.

SURAT KEPUTUSAN
REKTOR IIB DARMAJAYA
NOMOR : SK. 0320/DMJ/DFIK/BAAK/V-22
Tentang
Dosen Pembimbing Skripsi
Semester Genap TA.2021/2022
Program Studi S1 Sistem Komputer

REKTOR IIB DARMAJAYA

Memperhatikan : 1. **Bahwa dalam rangka usaha peningkatan** mutu dan peranan IIB Darmajaya dalam melaksanakan Pendidikan Nasional perlu ditingkatkan kemampuan mahasiswa dalam **Skripsi.**
2. Laporan dan usulan Ketua Program Studi **S1 Sistem Komputer.**

Menimbang : 1. Bahwa untuk mengefektifkan tenaga pengajar dalam Skripsi **mahasiswa perlu ditetapkan Dosen Pembimbing Skripsi.**
2. **Bahwa untuk maksud tersebut dipandang perlu** menerbitkan Surat Keputusan Rektor.

Mengingat : 1. UU No.20 Tahun 2003 Tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah No.60 Tahun 2010 tentang Pendidikan Sekolah Tinggi
6. Surat Keputusan Menteri Pendidikan Nasional Republik Indonesia No.165/D/0/2008 tertanggal 20 Agustus 2008 tentang Perubahan Status STMIK-STIE Darmajaya menjadi Informatics and Business Institute (IBI) Darmajaya
7. STATUTA IBI Darmajaya
8. Surat Ketua Yayasan Pendidikan Alfian Husin No. IM.003/YP-AH/X-08 tentang Persetujuan Perubahan Struktur Organisasi .
6. Surat Keputusan Rektor 0383/DMJ/REK/X-08 tentang Struktur Organisasi.

Menetapkan

Pertama : Mengangkat nama-nama seperti tersebut dalam lampiran Surat Keputusan ini sebagai Dosen Pembimbing Skripsi mahasiswa Program Studi S1 Sistem Komputer.

Kedua : Pembimbing Skripsi berkewajiban melaksanakan tugasnya sesuai dengan jadwal yang telah ditetapkan.

Ketiga : Pembimbing Skripsi yang ditunjuk akan diberikan honorarium yang besarnya sesuai dengan ketentuan peraturan dan norma penggajian dan honorarium IBI Darmajaya.

Keempat : Surat Keputusan ini berlaku sejak tanggal ditetap kan dan apabila dikemudian hari terdapat kekeliruan dalam keputusan ini, maka keputusan ini akan ditinjau kembali.

Ditetapkan di : Bandar Lampung
Pada tanggal : 09 Mei 2022
a.n. Rektor IIB Darmajaya,
Dekan Fakultas Ilmu Komputer

INSTITUT INFORMATIKA DAN BISNIS
DARMA

Dr. Sutedi, S.Kom., M.T.I
NIK. 00590203

1. Kepala Program Studi S1 Sistem Komputer
2. Yang bersangkutan
3. Arsip

Lampiran : Surat Keputusan Rektor IIB Darmajaya
Nomor : SK. 0320/DMJ/DFIK/BAAK/V-22
Tanggal : 09 Mei 2022
Perihal : Pembimbing Penulisan Skirpsi Semester Genap TA. 2021/2022
Program Studi Strata Satu (S1) Sistem Komputer

Judul Skripsi Dan Dosen Pembimbing Skripsi Semester Genap TA. 2021/2022
Program Studi Strata Satu (S1) Sistem Komputer

No	NAMA	NPM	JUDUL	PEMBIMBING
17	Singgih Remili Dharma	1911068005P	Implementasi Sistem Monitoring Serta Kontrol Suhu Dan Kelembaban Pada Ruang Green House Tanaman Sawi Berbasis Internet of Things (IoT)	Dodi Yudo Setiawan.S.Si.,M.Ti
18	Rino Cahya Prasetia	1911068011P	Perancangan Sistem Kontrol Dan Monitoring Internet of Things (IoT) Pada Kandang Burung Dara	Dodi Yudo Setiawan.S.Si.,M.Ti
19	M Renansyah Anderha	1811060041	Penerapan (IoT) Internet of Things Green House Untuk Kontrol Dan Monitoring Kadar pH, Suhu dan Kelembaban Tanah Pada Tanaman Melon	Ari Widiantoko, S.Kom.,M.Tech
20	Muhammad Saifudin Mahfudz	1811060042	Sistem Pendeteksi kendaraan Bermotor Berbasis Image Processing Menggunakan Raspberry Pi Pada alat Pantau pajak Parkir Kendaraan Bermotor	Bayu Nugroho, S.Kom.,M.Eng
21	Faqsy Cahya Syahbana	1811060003	Rancang Bangun Smart Farming Untuk Kontrol dan Monitoring Kadar Pupuk NPK Pada Tanaman Terung Berbasis IoT	Iia RosmaliaS.T., M.Kom
22	Galang Tirtoaji Pranata	1811060001	Rancang Bangun Sistem Antrian Berbasis RFID	Bayu Nugroho, S.Kom.,M.Eng
23	Suhendro	1811060040	Perancangan Sistem Monitoring Suhu Udara Dan Penyemprot Pestisida Secara Otomatis Berbasis ESP32	Zaidir Jamal, S.T.,M.Eng
24	Ramadani	1811060019	Sistem Smart Farming pada Tanaman Hidroponik Berbasis IoT di Screen House Balai Pelatihan Pertanian (BPP) Lampung	Dodi Yudo Setiawan.S.Si.,M.Ti
25	Wayan Aditya Pranata	1811060026	Rancang Bangun Alat Pengemasan Otomatis	Novi Herawadi Sudibyo, S.Kom., M.Ti
26	Gwin Cyril Vertido Somera	1811060043	Perancangan Sistem Monitoring Kualitas air (Kekeruhan Dan PH) Pada Tanaman Hidroponik Berbasis NodeMCU ESP32	Abdi Darmawan, S.T.,M.Ti
27	Satirc Dero Mahendra	1811060025	Rancang Bangun Kontrol dan Monitoring Akuarium Berbasis Internet of Things (IoT)	Novi Herawadi Sudibyo, S.Kom., M.Ti
28	Fadli Zaman	1811060017	Implementasi IGT Pada Vertikal Farming Di Green House	Nurfiana, S.Kom.,M.Kom
29	Aditya Pangestu	1811060035	Sistem Pengisian Madu Klengkeng Otomatis Menggunakan Mikrokontroller ESP 32	Lia Rosmalia, S.T.,M.Kom
30	Cindy Imas Pratiwi Sugiono	1811060031	Rancang Bangun Penyiram Taman Otomatis Berbasis Internet Of Things	Melia Gripin S, S.Kom.,M.T
31	Wulandari Eka Saputri	1811060002	Rancang Bangun Sistem Pemupukan Otomatis Untuk Tanaman Padi Sawah	Nurfiana, S.Kom.,M.Kom
32	Lintang Ayu Zahiroh	1811060022	Rancang Bangun Mesin Dan Penyiraman Otomatis Larutan Jeruk Nipis Pada Penetas Telur Berbasis Internet of Things (IoT)	Melia Gripin S, S.Kom.,M.T

Institut Informatika & Bisnis

DARMAJAYA

Yayasan Alfian Husin

Jl. Zainal Abidin Pagar Alam No. 93 Bandar Lampung 35142 Telp 787214 Fax. 700261 http://darmajaya.ac.id

FORMULIR

BAGIAN ADMINISTRASI AKADEMIK KEMAHASISWAAN (BAAK)

FORM KONSULTASI/BIMBINGAN SKRIPSI/TUGAS AKHIR *)

NAMA : SINGGIH KAMILI DHARMA

NPM : 1911068005P

PEMBIMBING I : Dodi Yudo Setyawan S.Si., M.T.I

PEMBIMBING II : ....

JUDUL LAPORAN : IMPLENTASI SISTEM MONITORING SERTA KONTROL SUHU DAN KELEMBABAN PADA RUANG GREEN HOUSE TANAMAN SAWI BERBASIS IOT

TANGGAL SK : 09-mei-2022 s.d .... (6+2 bulan)

No	HARI/TANGGAL	HASIL KONSULTASI	PARAF
1	Selasa/2-06-2022	observasi di lab IOT	
2	Selasa/21-06-2022	konsultasi hasil Revisi seminar proposal	
3	Rabu/13-Juni-2022	Perubahan Flowchart	
4	Rabu/13-Juni-2022	Revisi Bab I dan III	
5	Senin/25-Juli-2022	Progres Alat penelitian	
6	Kamis/28-Juli-2022	Revisi Alat	
7	Jumat/05-08-2022	Revisi Bab IV	
8	Selasa/09-08-2022	Revisi Bab IV Dan Bab V	
9	Senin/15-08-2022	pengecekan sistem Dan alat	
10	Kamis/18-Agustus-2022	Revisi naskah full (Acc)	

*) *Coret yang tidak perlu*

Bandar Lampung, 22-08-2022

Ketua Jurusan

(Novi Herawati Susibyo, S.kom., M.T.I

NIK. 11690310

Institut Informatika & Bisnis

DARMAJAYA

Yayasan Alfian Husin

Jl. Zainal Abidin Pagar Alam No. 93 Bandar Lampung 35142 Telp 787214 Fax. 700261 http://darmajaya.ac.id

FORMULIR

KARTU SEMINAR PROPOSAL

MA : SINGGIH . REMILI . DHARMA

M : 1911068005P

OGRAM STUDI : TI / SI / MI / [✓ SK] / TK / MA / AK / AD *)

KULTAS : [✓ ILMU KOMPUTER] / ILMU BISNIS DAN EKONOMI

o.	Tanggal	Nama Mahasiswa	Judul	Paraf Pembahas
1	30-MEI-2022	NASRIL AHMAD	Implementasi Internet of Things (IOT) Pada Perancangan sistem Kontrol Dan monitoring Pengeringan Pada (Gabah)	
2	31-05-2022	Ahmad Yudi Firmansyah	Rancang Bangun sistem Pengendali monitoring suhu Dan kelembaban pada rumah Jamur Berbasis web	
3	31-05-2022	Aliu Dona	Peroncongan Sistem Keamanan Brangkas menggunakan Face Recognition Berbasis ESP32	
4	02-06-2022	Rino Canya .P	Perancangan Sistem Kontrol Dan monitoring Internet of Things (IOT) Pada Kandang burung dara	
5	02-06-2022	M. Renansyah .A	Penerapan IOT GreenHouse untuk Kontrol Dan monitoring Kadar PH, suhu dan kelembaban	
6				

Coret yang tidak perlu

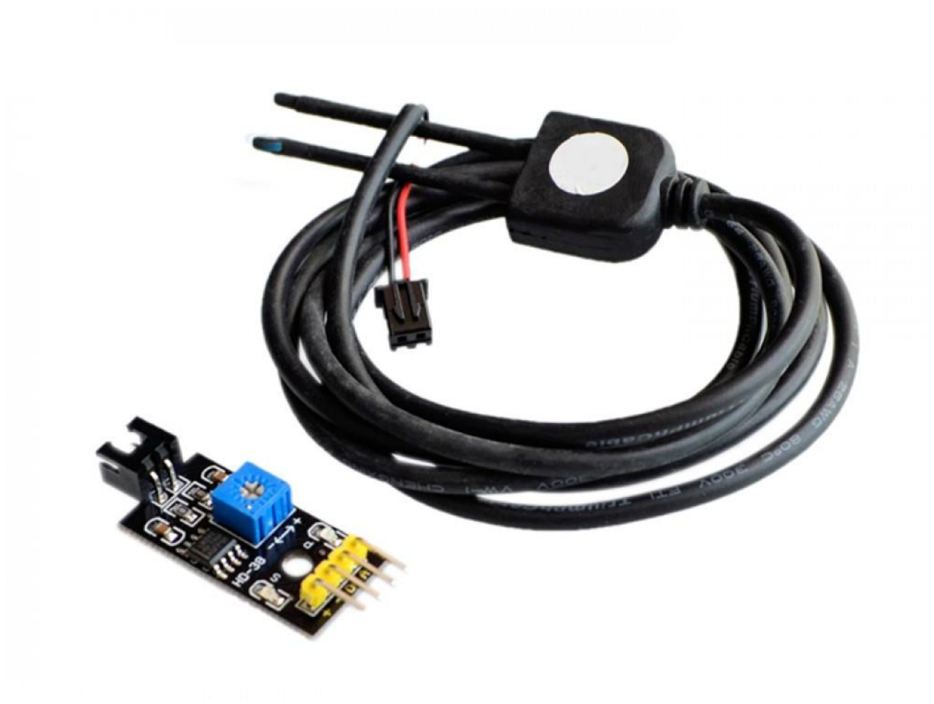
Catatan :

Mahasiswa wajib menghadiri seminar proposal minimal 5(lima) kali sebelum tampil seminar proposal

Bandar Lampung,
Ka. Jurusan

(Novi Herawadi Sudibyo, S.Kom, M.T.)
NIK. 11690310

Soil Moisture Hygrometer Detection Sensor Module W/ Corrosion Resistance Probe DC 3.3-12V for Arduino



Description:

NO.	NO.	Value
1	Name	Soil humidity sensor
2	Size	36*15*7mm
3	Voltage	DC 3.3-12V
4	Current	<20mA; <30mA (output)
5	Interface	+ - DO AO; DO digital value; AO analog value
6	Operating Temperature	-25~85 Celsius

Application:

It is used for detecting soil humidity.

Principle:

The soil humidity probe is for detecting humidity, and the voltage comparator is for judging the humidity amount. When the humidity is larger than set value, DO will output to the low level.

Using Introduction:

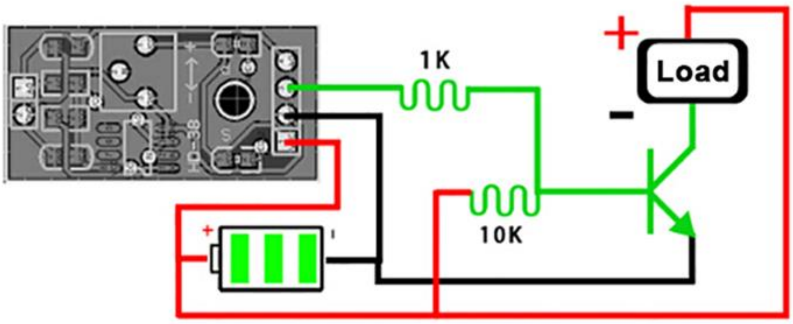
1. "+" connects power positive; "-" connects power negative. Please do not connect reversely, or it will burn the chip. After connection, "P" power indicator lamp will be on and the module will operate normally

2. Clockwise/counter clockwise adjusts the blue potentiometer, and it can improve/reduce the detection sensitivity. Take the "Thermal Sensor" as an example: if it defaults to a 25 Celsius trigger, then the DO output low level is valid. After clockwise adjusting, that is when the temperature is lower than 25 Celsius, it can trigger. If you are counter clockwise adjusting, then higher than 25 Celsius can trigger.

3. When clockwise adjusts the potentiometer so the "S" output indicator is on, it shows that the adjusting range has surpassed the testing threshold. If counter clockwise adjusting is extended to the max, it surpasses the threshold. When adjusting, please do no let them over the threshold, or it cannot be used normally. Potentiometer adjustment is only used for the DO port, and has no relevance to AO output.

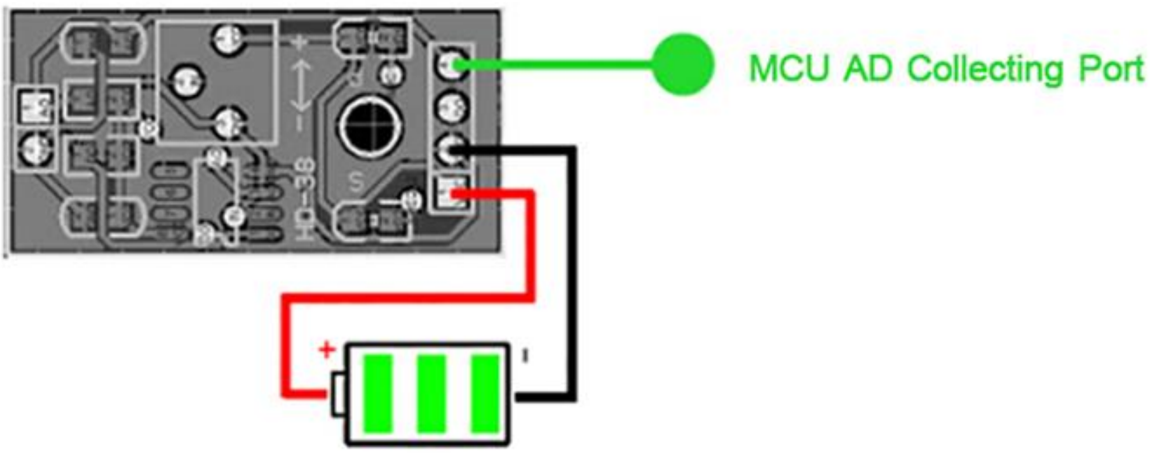
4. AO output is analog value (voltage), and it can detect via MCU AD port

Wiring Diagram 1:



When the detected value is more than the set value, DO will output low level, the triode will be connected and the load will operate. 1K is current-limiting resistor, and 10K is pull-up resistor on DO port.

Wiring Diagram 2:



No more complex circuits, AO port can collect voltage via MCU AD port.