LAMPIRAN A

No. Dok. : F-PBM-17 Tgl. Berlaku : 2 Januari 2021 No. Rev. : 00

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : http://polsri.ac.id, Pos El : info@polsri.ac.id

LEMBAR BIMBINGAN TUGAS AKHIR

Lembar : 1

Nama	:	Nopi Carlina
NPM	:	061940342328
Jurusan/Program Studi	:	Teknik Elektro
Judul Tugas Akhir	:	Sistem monitoring *vertical farming* menggunakan internet of things (IoT)
Pembimbing I	:	Yurni Oktarina,S.T.,M.T

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	9/2 2023	konsultasi tema tugas akhir => vertical farming	
2.	13/2 2023	Fix judul tugas akhir	
3.	27/2 2023	Pengajuan Bab I & Bab II revisi latar belakang, penambahan 5 penelitian yg serupa, batasan masalah Bab II => perbaiki susunan teori, & referensi (citasi)	
4.	16/3 2023	Revisi Bab I, Bab II & Bab III penambahan materi IoT & tanaman	
5.	30/3 2023	Revisi Bab II, III. Perbaiki diagram kendali, flow chart, + materi pembanding Blynk vs Node R.	
6.	3/4 2023	Acc Bab I, II, Revisi Bab III, cara kerja IoT, + Cara install Node Red	

No. Dok. : F-PBM-17

Tgl. Berlaku : 2 Januari 2021

No. Rev. : 00

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
7.	2/8 2023	Pgajuan Bab III revisi	
8.	2/8 2023	Pgajuan Bab IV, revisi, data, analisa	
9.	3/8 2023	Revisi Bab IV, tabel	
10.	4/8 2023	ACC BAB IV, Revisi BAB V	
11.	7/8 2023	Revisi Kesimpulan	
12.	8/8/2023	Acc tugas akhir → ujian TA	

Palembang, ...

Koordinator Program Studi

Sarjana Terapan Teknik Elektro

Masayu Anisah, S.T., M.T.

NIP 197012281993032001

Catatan:

*) melingkari angka yang sesuai.

- Ketua Jurusan harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Tugas Akhir sebelum menandatangani lembar bimbingan ini.
- Lembar pembimbingan LA ini harus dilampirkan dalam Laporan Tugas Akhir.

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar - Palembang 30139 Telepon (0711) 353414
Laman : http://polsri.ac.id, Pos El : info@polsri.ac.id

LEMBAR BIMBINGAN TUGAS AKHIR

Lembar : 1

Nama	:	Nopi Carlina
NPM	:	061940342328
Jurusan/Program Studi	:	Teknik Elektro
Judul Tugas Akhir	:	Sistem monitoring *vertical farming* menggunakan internet of things (IoT)
Pembimbing II	:	Sabilal Rasyad,S.T.,M.Kom

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	14/2 - 23	Diskusi Judul.	
2.	6/3 - 23	Buat Jurnal dan Tujuan & permasalahan di Pertanian	
3.	13/4 .23	Lanjut bab I.	
4.	15/6 23	Bab I. Perbaiki sesuai saran	
5.	22/6 -23	Bab I. dan Bab II perbaiki tujuan, gbr skematik, rangkai. ekivalen tabel pin	
6.	6/7 -23.	Bab 2 diperbaiki utk ilmu Dasar	

Lembar : 2

No. Dok. : F-PBM-17

Tgl. Berlaku : 2 Januari 2021

No. Rev. : 00

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
7.	15/7 -23	Bab II. Ok lanjut, Bab III. Permasalahan. yg perbaiki	
8.	25/7 -23	Bab 3 perbaiki. Sesuai saran	
9.	27/7 -23	Bab 3 diperbaiki lanjut bab 4	
10.	04/8 -23	Bab 4, penjelasan grafik dan tabel: penggunaan Mode [illegible]	
11.	09/8 -23	Bab IV. Ok lanjut Bab V	
12.	10/8 -23	Bab V. tg kesimpulan. Sesuai dg tujuan: Silahkan Seminar	

Palembang, ...
Koordinator Program Studi
Sarjana Terapan Teknik Elektro

Masayu Anisah, S.T., M.T.
NIP 197012281993032001

Catatan:

*) melingkari angka yang sesuai.

- Ketua Jurusan harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Tugas Akhir sebelum menandatangani lembar bimbingan ini.
- Lembar pembimbingan LA ini harus dilampirkan dalam Laporan Tugas Akhir.

No. Dok. : F-PBM-18 Tgl. Berlaku : 13 Desember 2010 No. Rev. : 00

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polisriwijaya.ac.id **E-mail :** info@polsri.ac.id

REKOMENDASI UJIAN TUGAS AKHIR (TA)

Pembimbing Tugas Akhir memberikan rekomendasi kepada,

Nama : Nopi Carlina

NIM : 061940342328

Jurusan/Program Studi : Sarjana Terapan Teknik Elektro

Judul Tugas Akhir : SISTEM MONITORING SUHU DAN KELEMBABAN PADA TANAMAN *VERTICAL FARMING* BERBASIS IOT.

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Tugas Akhir (TA) pada Tahun Akademik 2023

Palembang, 07 Agustus 2023

Pembimbing I,

Yurni Oktarina,S.T.,M.T.
NIP. 197710162008122001

Pembimbing II,

Sabilal Rasyad.S.T.,M.Kom.
NIP. 197409022005011003

POLITEKNIK NEGERI SRIWIJAYA

HIMPUNAN MAHASISWA JURUSAN

TEKNIK ELEKTRO

Jalan Srijaya Negara Bukit Besar, Palembang 30139
Telepon 0711 353414 Faximili 0711-355918
Website :*hmjelektro.polsri.ac.id*, Email:*hmjelektro@polsri.ac.id*, CP: 088287135861

Letter of Acceptance

04 Agustus 2023

Kepada,
Nopi Carlina
Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya
Palembang, Sumatera Selatan

Dengan ini kami menyampaikan bahwa, berdasarkan hasil seleksi yang dilakukan oleh *reviewer Electro National Conference (ENACO)* dengan tema ***"Explore for Sustainable our Knowledge in Industrial Sector to Reach the Improvement Technology of SDGs 2030",*** makalah dengan rincian:

Judul	: Sistem *Monitoring* Suhu dan Kelembapan pada Tanaman *Vertical Farming* Berbasis IOT
Penyaji	: Nopi Carlina
Email	: nopicarlina2@gmail.com

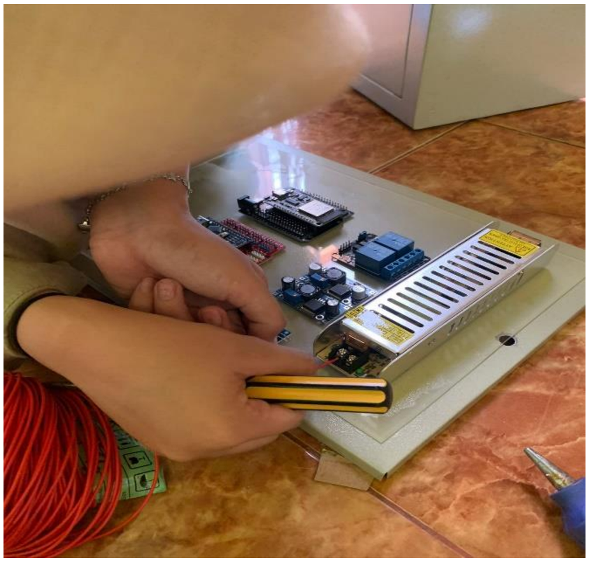
Dinyatakan **DITERIMA** untuk dipublikasikan di dalam buku Prociding *Electro National Conference (ENACO)* dengan e-ISSN 2797-0515 dan p-ISSN 2777-0958. Artikel tersebut akan tersedia secara *online* di https://enacoelektropolsri.com.

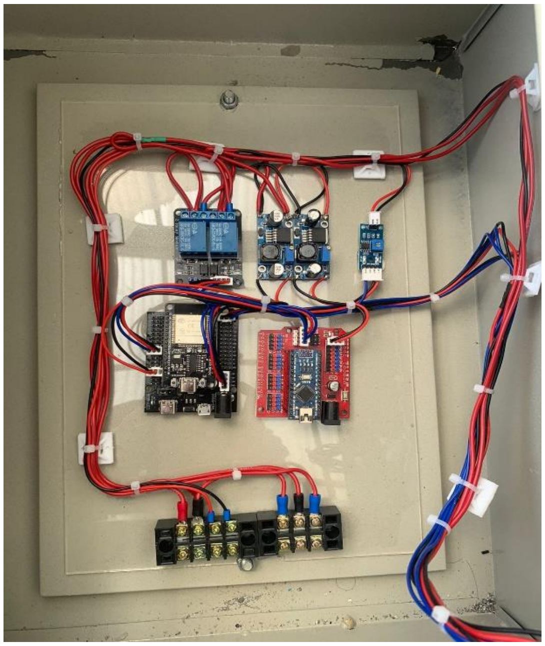
Demikian hal ini kami sampaikan, atas perhatian dan kerja samanya kami ucapkan terima kasih.

Ketua Panitia,

Destra Andika Pratama, S.T., M.T
NIP 197712202008121001

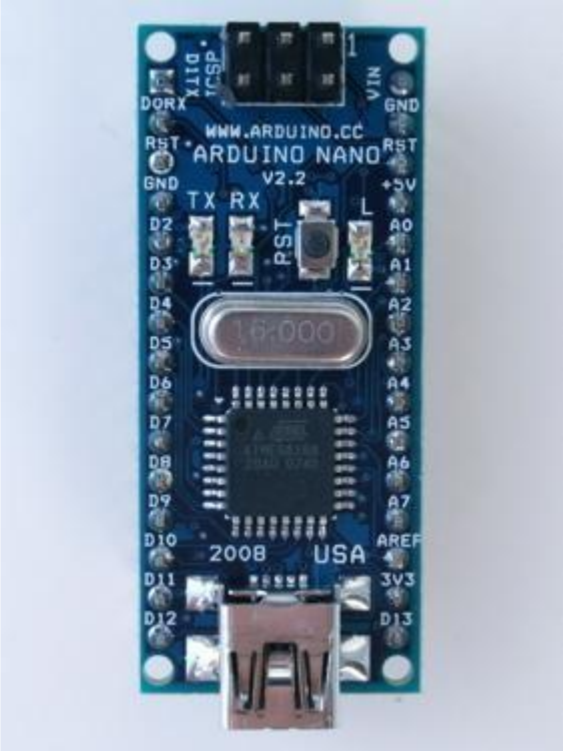
LAMPIRAN B



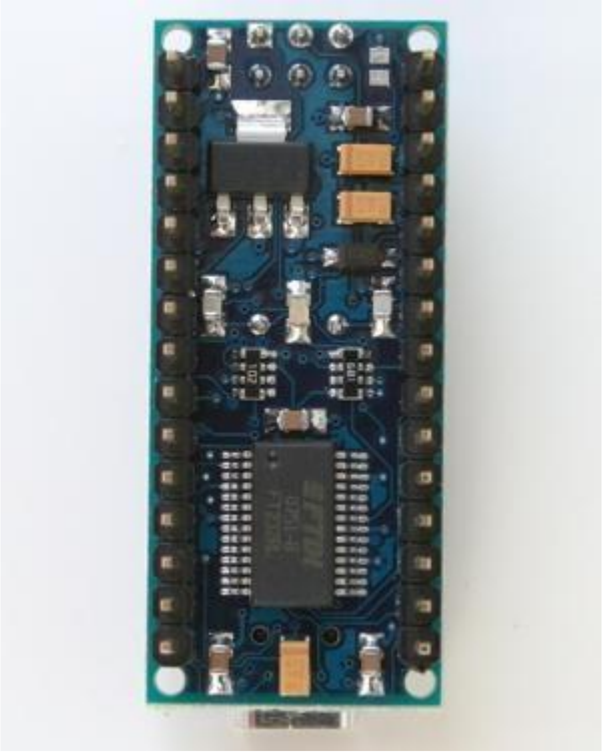


LAMPIRAN C

Arduino Nano



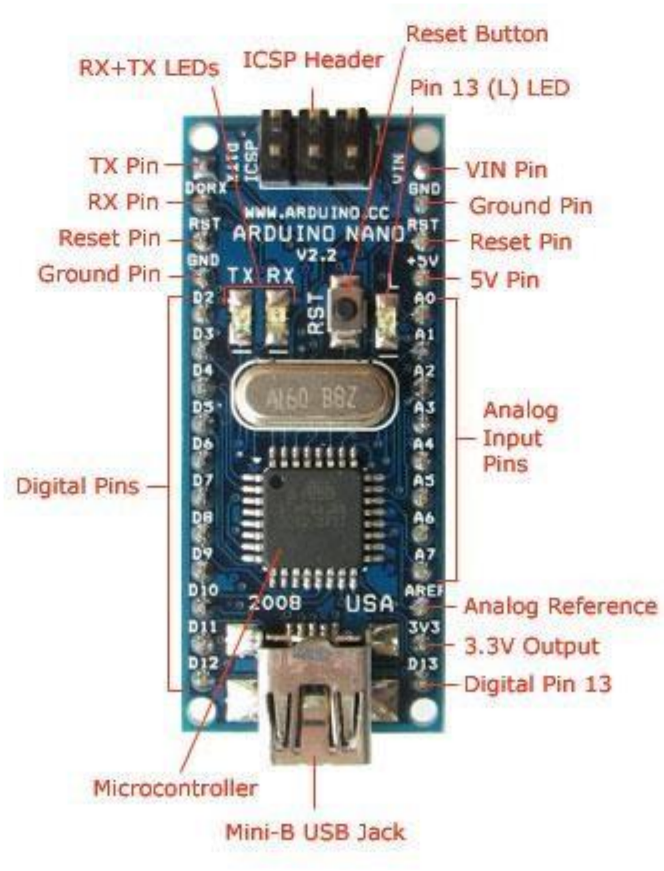
Arduino Nano Front

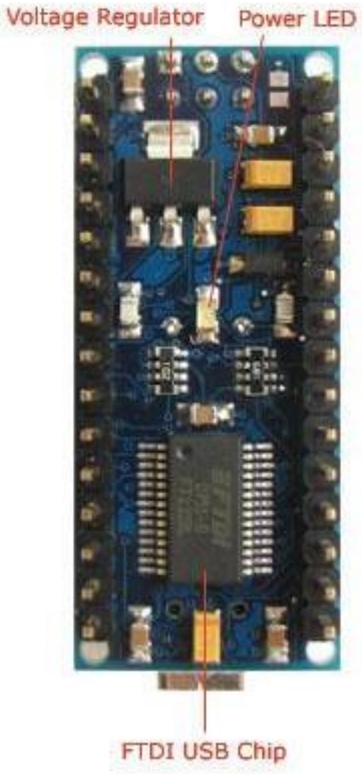


Arduino Nano Rear

Overview

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.0) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech.





Schematic and Design

Arduino Nano 3.0 (ATmeGA328): schematic, Eagle files.

Arduino Nano 2.3 (ATmeGA168): manual (pdf), Eagle files. *Note:* since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

Specifications:

Microcontroller Atmel ATmega168 or ATmeGA328
Operating Voltage (logic level)
Input Voltage (recommended)
5 V

7-12 V
Input Voltage (limits) 6-20 V
Digital I/O Pins 14 (of which 6 provide PWM output)
Analog Input Pins 8
DC Current per I/O Pin 40 mA
Flash Memory 16 KB (ATmeGA168) or 32 KB (ATmeGA328) of which 2 KB

The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Additionally, some pins have specialized functionality:

I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().
Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmeGA168 ports.

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmeGA168 and ATmeGA328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Nano's digital pins.
The ATmeGA168 and ATmeGA328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmeGA168 or ATmeGA328 datasheet.

Programming

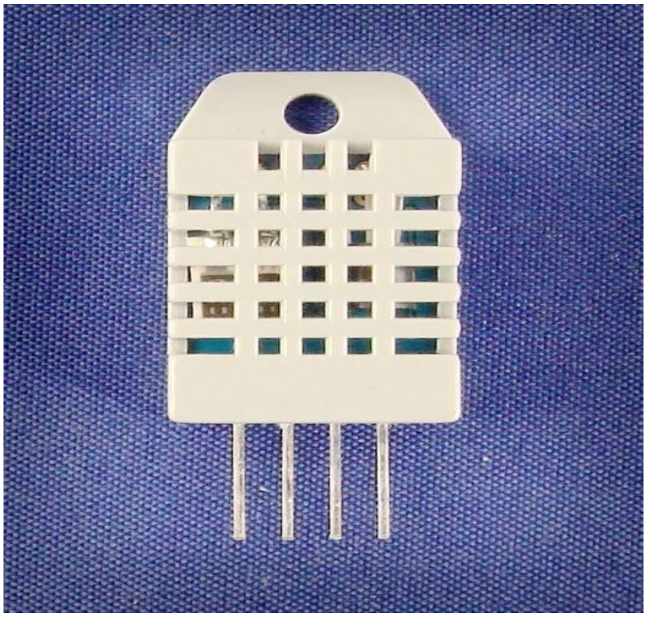
The Arduino Nano can be programmed with the Arduino software (download).
Select "Arduino Diecimila, Duemilanove, or Nano w/ ATmeGA168" or "Arduino Duemilanove or Nano w/ ATmeGA328" from the **Tools**

> **Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.
The ATmega168 or ATmega328 on the Arduino Nano comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).
You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 or ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.
This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

DATASHEET DHT-22



1. Feature & Application:

* Full range temperature compensated * Relative humidity and temperature measurement
* Calibrated digital signal *Outstanding long-term stability *Extra components not needed
* Long transmission distance * Low power consumption *4 pins packaged and fully interchangeable

2. Description:

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability.Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions.

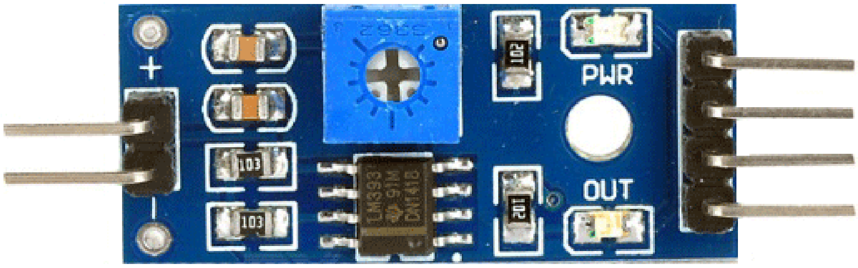
Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor

Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +-2%RH(Max +-5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+-0.3%RH
Long-term Stability	+-0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

Soil Moisture Sensor Module



This **soil moisture sensor module** is used to detect the moisture of the soil. It measures the volumetric content of water inside the soil and gives us the moisture level as output. The module has both digital and analog outputs and a potentiometer to adjust the threshold level.

Soil Moisture Sensor Module Pinout Configuration

Pin Name	**Description**
VCC	The Vcc pin powers the module, typically with +5V
GND	Power Supply Ground

DO	Digital Out Pin for Digital Output.
AO	Analog Out Pin for Analog Output

Soil Moisture Sensor Module Features & Specifications

- Operating Voltage: 3.3V to 5V DC
- Operating Current: 15mA
- Output Digital - 0V to 5V, Adjustable trigger level from preset
- Output Analog - 0V to 5V based on infrared radiation from fire flame falling on the sensor
- LEDs indicating output and power
- PCB Size: 3.2cm x 1.4cm
- LM393 based design
- Easy to use with Microcontrollers or even with normal Digital/Analog IC
- Small, cheap and easily available

LAMPIRAN C

```
#include <DHT.h>
#include <DHT_U.h>

// NodeRed
#include <ESP8266WiFi.h>
#include <DHT.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char* ssid = "vivo";
const char* password = "12345678";
ESP8266WebServer server(80);

// Sensor Soil
#define sensorSoil analogRead(0)
int sensor;

// LCD 16x2 I2C
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

// Sensor DHT22
#define DHTPIN D6
String temp;
String hum;
String kel;
float t, h;
#define DHTTYPE DHT22  // DHT 11
DHT dht(DHTPIN, DHTTYPE);

// Modul Relay
#define motorPump 14

// Timer Penyiraman
const int soilMoistureThreshold = 75;
const unsigned long wateringDuration = 5000;  // Durasi penyiraman dalam
milidetik (misalnya 5 detik)

unsigned long previousWateringTime = 0;
bool isWatering = false;

//----------------------------------------------------------
void handleRoot() {
  server.send(200, "text/plain", "hello from esp8266!");
```

```
}
//----------------------------------------------------------
void handleNotFound() {
  String message = "File Not Found\n\n";
  message += "URI: ";
  message += server.uri();
  message += "\nMethod: ";
  message += (server.method() == HTTP_GET) ? "GET" : "POST";
  message += "\nArguments: ";
  message += server.args();
  message += "\n";
  for (uint8_t i = 0; i < server.args(); i++) {
    message += " " + server.argName(i) + ": " + server.arg(i) + "\n";
  }
  server.send(404, "text/plain", message);
}

void setup() {
  Serial.begin(9600);
  lcd.begin();
  pinMode(motorPump, OUTPUT);
  digitalWrite(motorPump, 1);

  WiFi.begin(ssid, password);
  Serial.println("");

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());
  dht.begin();
  server.on("/", handleRoot);

  server.on("/dht-temp", []() {
    t = dht.readTemperature();
    temp = String(t);
    server.send(200, "text/plain", temp);
  });

  server.on("/soil-kel", []() {
    sensor = map(sensorSoil, 0, 1023, 0, 100);
    kel = String(sensor);
    server.send(200, "text/plain", kel);
  });

  server.on("/dht-hum", []() {
    h = dht.readHumidity();
    hum = String(h);
    server.send(200, "text/plain", hum);
  });

  server.onNotFound(handleNotFound);
  server.begin();
  Serial.println("HTTP server started");
}

void loop() {
  server.handleClient();
  LCDDisplay();
}
//--------------------------------------------
void LCDDisplay() {
  float s = dht.readTemperature();
  float f = dht.readHumidity();
  sensor = map(sensorSoil, 0, 1023, 0, 100);
  lcd.setCursor(0, 0);
  lcd.print("Penyiraman Cabai IoT");
  lcd.setCursor(0, 1);
  lcd.print("Suhu : ");
  lcd.print(s);
  lcd.print(" ");
  lcd.write(223);
  lcd.print("C");
  lcd.setCursor(0, 2);
  lcd.print("Humidity : ");
  lcd.print(f);
  lcd.print(" %");
  lcd.setCursor(0, 3);
  lcd.print("Sensor Soil : ");
  lcd.print(sensor);
  lcd.print(" %");
  if (sensor >= soilMoistureThreshold && !isWatering) {
    startWatering();
  } else  {
    stopWatering();
  }
  delay(250);
  lcd.clear();
}
```

POMPA

```
void startWatering() {
  digitalWrite(motorPump, 0);  // Aktifkan relay, nyalakan motor pump
  isWatering = true;
  // previousWateringTime = millis();
}

void stopWatering() {
  digitalWrite(motorPump, 1);  // Matikan relay, matikan motor pump
  isWatering = false;
}
```