

LAMPIRAN

KODE PROGRAM NODEMCU ESP8266

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include "DHT.h"
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 16, 2); // sda scl pin d1/d2

#define DHTPIN 14 //d5 dht
#define DHTTYPE DHT11
#define ONE_WIRE_BUS 2 //d4 suhu air
#define phUP 12 // d7
#define phDOWN 13 // d8
#define SensorpH 16 //Sensor pH pada pin A0

const int ph_Pin =A0;
float Po = 0;
float PH_Step;
int nilai_analogPH;
double teganganPh;

//Nilai kalibrasi
float Ph4 = 3.132; //Nilai kalibrasi
float Ph7 = 2.690;
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

DHT dht(DHTPIN, DHTTYPE);
void setup()
{
  pinMode(ph_Pin,INPUT);
```

```

sensors.begin();
dht.begin();
// Start serial communication for debugging purposes
Serial.begin(9600);
pinMode (phUP,OUTPUT);
pinMode (phDOWN,OUTPUT);
// Start up the library

lcd.begin(); // run lcd
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Monitoring");
lcd.setCursor(0,1);
lcd.print("Hidroponik");
delay(2000);
lcd.clear();
}

void loop(){
nilai_analogPH = analogRead(ph_Pin);
Serial.print("adc ph :");
Serial.println(nilai_analogPH);
teganganPh = 3.3 / 1024.0 * nilai_analogPH;
Serial.print("teganganPh :");
Serial.println(teganganPh,3);

PH_Step=(Ph4 - Ph7) / 3;
Po = 7.00 + ( (Ph7 - teganganPh) / PH_Step);
//po= 7.00 + ((Ph4 -teganganPh) / PH_Step ));

float t = dht.readTemperature(); //suhu
float h = dht.readHumidity(); //kelembaban

```

```
    if (isnan(t) || isnan(h)) {
      Serial.println("no sensor");
      return;
    }

    //sensor suhu air
    sensors.requestTemperatures();
    Serial.print("Suhu Air: ");
    Serial.println(sensors.getTempCByIndex(0));

    Serial.print("Kadar PH: ");
    Serial.println(Po,2);
    //suhu udara
    Serial.print("Suhu : ");
    Serial.print(t);
    Serial.println(" *C");
    Serial.print("Kelembaban: ");
    Serial.print(h);
    Serial.println(" %");
    Serial.println(" _____");
    //LCD
    lcd.setCursor(0,0);
    lcd.print("Suhu Air:");
    lcd.print(sensors.getTempCByIndex(0));
    lcd.setCursor(0,1);
    lcd.print("Kadar PH: ");
    lcd.print(Po,2);
    delay(5000);
    //LCD

    lcd.setCursor(0,0);
```

```
lcd.print("Suhu : ");
lcd.print(t);
lcd.print(" *C ");
lcd.setCursor(0,1);
lcd.print("Lembab : ");
lcd.print(h);
lcd.print(" &");
delay (5000);
/* ph_up();
   ph_down ();
   ph_7();*/
// void up();
// void down();
if ( Po >=7 ){
    digitalWrite (phDOWN,HIGH);
    //digitalWrite (phUP,LOW);
}
if (Po <=6){
    digitalWrite (phUP,HIGH);
    //digitalWrite (phDOWN,LOW);
}
else {
    digitalWrite (phUP,LOW);
    digitalWrite (phUP,LOW);
}

}

/* void up(){
   if ( Po >=7 ){
       digitalWrite (phDOWN,HIGH);
       //digitalWrite (phUP,LOW);
   }
}
```

```
else {  
    digitalWrite (phDOWN,LOW);  
}  
}  
  
void down(){  
    if (Po <=6){  
        digitalWrite (phUP,HIGH);  
        //digitalWrite (phDOWN,LOW);  
    }  
    else {  
        digitalWrite (phUP,LOW);  
    }  
}
```

KODE PROGRAM SENSOR pH

```

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2); // address lcd mungkin berbeda silahkan baca
dulu addressnya lihat di www.anakkendali.com untuk program membaca alamat
i2c
float voltage;
float pHValue;
//deklarasi pin analog
int adcPH;
void setup() {
  // put your setup code here, to run once:
  lcd.begin();
  Serial.begin(9600);
}
void loop() {
  // put your main code here, to run repeatedly:
  adcPH = analogRead(A0); //menggunakan pin A0 untuk membaca output sensor
  pH
  voltage = adcPH *5.0/1022;
  pHValue = (6.4516*voltage)-5.7742;
  Serial.print("sensor ph : ");
  lcd.println(pHValue);
  lcd.print("sensor ph : ");
  lcd.print(pHValue);
  delay(1000);
}

```

DATA SHEET

FireBeetle ESP8266 IOT Microcontroller SKU: DFR0489

[\(http://www.dfrobot.com/\)](http://www.dfrobot.com/) **Home** (<https://www.dfrobot.com/>) > **Arduino** (<https://www.dfrobot.com/index.php?route=product/category&path=35>) > **Microcontroller** (https://www.dfrobot.com/index.php?route=product/category&path=35_104)

Contents

- 1 Introduction
- 2 Specification
- 3 Board Overview
- 4 Tutorial
 - 4.1 Requirements
 - 4.2 Setup Arduino IDE Development Environment
 - 4.3 Sample Code - **Blink**
 - 4.4 Sample Code - **Scan WIFI**
- 5 FAQ
- 6 More Documents



(<https://www.dfrobot.com/product-1634.html>)
 FireBeetle ESP8266 IOT Microcontroller
 (<https://www.dfrobot.com/product-1634.html>)

Introduction

DFRobot FireBeetle is a series of low-power-consumption development hardware designed for Internet of Things (IoT). Firebeetle ESP8266 is a development board integrated with IoT WIFI, TCP/IP, 32-bit MCU, 10-bit ADC and multiple interfaces such as HSPi, UART, PWM, I2C and I2S. In DTIM10, the full power consumption to maintain WiFi connection reached to 1.2mW. Equipped with 16MB outer SPI flash memory, ESP8266 is available for programs and firmware storage.

Compatible with Arduino programming enables Firebeetle ESP8266 to lower the barrier of programming. Operator can implement Arduino programming codes directly onto ESP8266 to reduce the difficulty of operating and increase the stability of board.



(/wiki/index.php/File:Warning_yellow.png)

NOTE: There still remains some bugs to be detected and fixed by developers. In some cases some peripherals may not work perfectly by embedding Arduino sample codes in ESP8266. Much more functions implemented in Arduino are still under development and improvement. An alternative way is to change development tool such as RTOS and MicroPython towards a more operating-friendly experience in some conditions.

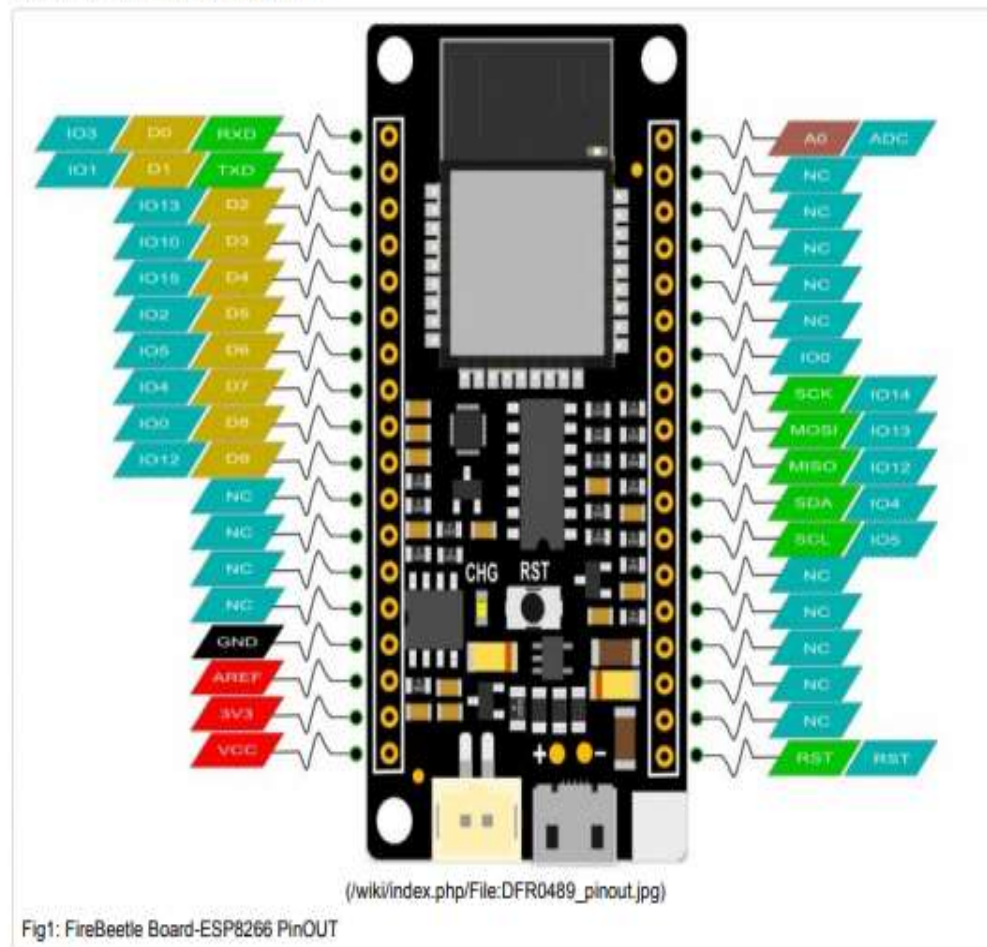
Specification

- Operating Voltage: 3.3V
- Input Voltage (limits): 3.3~5V (Lithium Battery:3.7V & USB:5V)
- Microcontroller: Tensilica L106 (32-bit MCU)
- Clock Speed: 80MHz (Maximum: 160MHz)
- SRAM: 50KB
- External Flash Memory: 16MB

- DC Current in the Low-Power-Consumption: 46uA
- Average Operating Current: 80mA
- Maximum Discharging Current: 600mA (LDO-3.3 Output)
- Maximum Charging Current: 500mA
- Digital Pin x10
- Analog Pin x1
- SPI interface x1
- I2C interface x1
- IR interface x1
- I2S interface x1
- Interface: XH2.54mm Pin (No soldering default)
- In Combination of Wi-Fi MAC/ BB/RF/PA/LNA
- WiFi: IEEE802.11 b/g/n (2.4 GHz~2.5 GHz), not support 5GHz WiFi
- Operating Temperature: -10°C~+55°C
- Dimension: 58 × 29(mm)
- Weight: 24g

Board Overview

FireBeetle Board - ESP8266 is not only compatible with ESP8266 PinMap, but also make a special compatible with Arduino IDE PinMap. Dx (x=0,1,2,3...9)

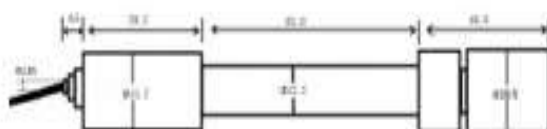


- CHG Blink = not connect battery; Light on = Charging; Light off = Charge complete

Introduction

Need to measure water quality and other parameters but haven't got any low cost pH meter? Find it difficult to use with Arduino? Here comes an analog pH meter, specially designed for Arduino controllers and has built-in simple, convenient and practical connection and features. It has an LED which works as the Power Indicator, a BNC connector and PH2.0 sensor interface. You can just connect the pH sensor with BNC connector, and plug the PH2.0 interface into any analog input on Arduino controller to read pH value easily.

Specification



SEN0161 dimension

- Module Power: 5.00V
- Circuit Board Size: 43mm×32mm
- pH Measuring Range: 0-14
- Measuring Temperature: 0-60 °C
- Accuracy: $\pm 0.1\text{pH}$ (25 °C)
- Response Time: $\leq 1\text{min}$
- pH Sensor with BNC Connector
- PH2.0 Interface (3 foot patch)
- Gain Adjustment Potentiometer
- Power Indicator LED

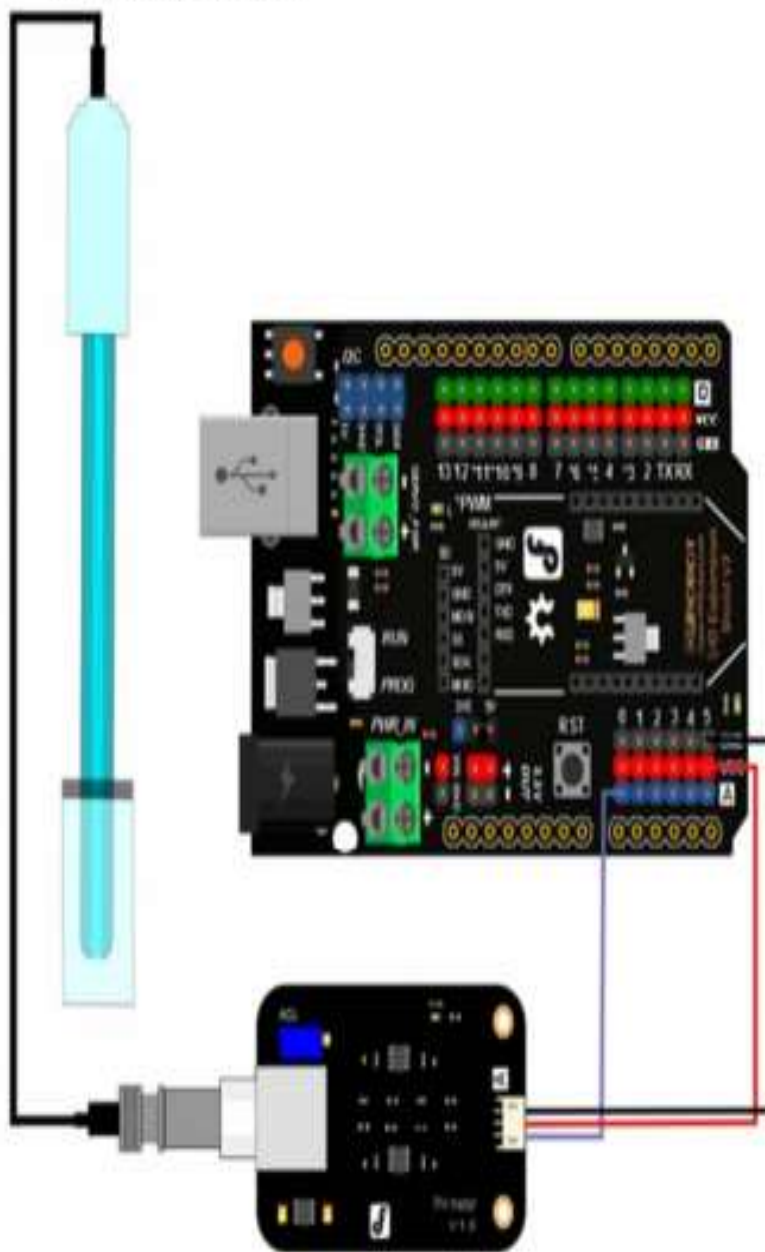
Precautions

- Before and after use of the pH electrode every time, you need to use (pure)water to clean it.
- The electrode plug should be kept clean and dry in case of short circuit.
- **Preservation:** Electrode reference preservation solution is the **3N KCL** solution.
- Measurement should be avoided staggered pollution between solutions, so as not to affect the accuracy of measurement.
- Electrode blub or sand core is defiled which will make PTS decline, slow response. So, it should be based on the characteristics of the pollutant, adapted to the cleaning solution, the electrode performance recovery.

NOTE: It is normal that if your reading is much different with the table since you are not reading from the electrode directly but from the voltage adapter, it has converted the original voltage (-5V ~ +5V) to Arduino compatible voltage, i.e. 0 ~ 5V. [See the discussion on Forum.](#)

Usage

Connecting Diagram



- Electrode when in use, the ceramic sand core and liquid outlet rubber ring should be removed, in order to make salt bridge solution to maintain a certain velocity.

NOTE: Differences between the probes, SEN0161 and SEN0169

Their usages/ specifications are almost the same. The differences locates at

Long-firing Operation: SEN0169 supports, while SEN0161 NOT, i.e. you can not immerse SEN0161 in water for Continuous Testing.

Life Span: In 25 °C, pure water, do Continuous Testing with them both, SEN0169 can work two years, while SEN0161 can only last for 6 months. And just for reference, if put them in turbid, strongly acid and alkali solution, 25°C, the life span would drop to one year (SEN0169), 1 month(or shorter, SEN0161).

Temperature, pH, turbidity of the water effect the probe life span a lot.

Waterproof: You can immerse the whole probe SEN0169 into the water, while you can only immerse the front part of the probe SEN0161, the electrode glass bulb, into water, the rear part, from the white shell to the cable, MUST NOT be under water.

Strongly Acid and Alkali: SEN0169 are preferred for strongly acid and alkali test. And if your testing range is usually within pH6~8, then SEN0161 is capable for that.

pH Electrode Characteristics

The output of pH electrode is Millivolts, and the pH value of the relationship is shown as follows (25 °C):

VOLTAGE (mV)	pH value	VOLTAGE (mV)	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

Grove - 2-Channel SPDT Relay

Relay



The Grove - 2-Channel SPDT Relay has two single pole - double throw (SPDT) switches. It only requires low-voltage and low current signals to control those switches. Specifically, you can use 5V DC to control max.250V AC or 110V DC. The best thing is that you can control the two channels separately. For instance, controlled by SIG1, you can connect the COM1 to NC1 or

NO1 as your wish. It is so convenient and reliable that it can be applied to massive products or projects which need to switch high voltage/high current devices.

Specification

Item	Value
Operating voltage	5V
Nominal Coil Current	89.3mA
TUV Certification Load	10A 250VAC/ 10A 30VDC
UL Certification Load	10A 125VAC/ 10A 28VDC
Max. Allowable Voltage	250VAC/110VDC
Power Consumption	abt. 0.45W
Contact Resistance	100mΩ Max.
Insulation Resistance	100MΩ Min. (500VDC)
Max. ON/OFF Switching	30 operation/min
Ambient Temperature	-40°C to +85°C
Operating Humidity	45% to 85%RH
Contact Material	AgCdO
Input interface	Digital SIG1/SIG2
Output Port	3 Pins DIP Female Screw Terminal-Green

Hardware Overview

Pin Map



- ④ GND: connect this module to the system GND
- ③ VCC: you can use 5V for this module
- ② SIG2: control signal of switch2, high-NO2/low-NC2
- ① SIG1: control signal of switch1, high-NO1/low-NC1
- ⑤ NC1: one throw, connected to COM1 by default
- ⑥ COM1: controlled by SIG1, connected to NC1 or NO1
- ⑦ NO1: the other throw of switch1
- ⑧ NC2: one throw, connected to COM2 by default
- ⑨ COM2: controlled by SIG2, connected to NC2 or NO2
- ⑩ NO2: the other throw of switch2

K1 is the Relay module, there is a coil between **pin1** and **pin3** of K1. Defaultly, the **COM1** will connect to **NC1**. If the pin3 of K1 connected to the grand, then this coil will be 'closed', so the **COM1** will connect to **NO1**. To open this coil, it requires about 90mA, however, normally the GPIO pin of Arduino only can afford 20mA(40mA max.). Therefor, we use a NPN transistors **S9013** which can provide 500mA. The **SIG1** is pulled down by the 10k R2, if there is no signal, the 'Gate' of **Q1** will be 0v, and Q1 is turned off, so that the K1 will be 'opened'. If **SIG1** becomes 5v, then the Q1 will be turned on. **Pin3** of k1 will be connected to the GND of the system, for the K1 there will be 5V between **pin3** and **pin1**, so the coil will be 'closed', and the **COM1** will connect to **NO1**.

DC Mini Submersible Water Pump



Micro dc 3-6v micro submersible pump mini water pump for fountain garden mini water circulation system diy project dc 3v to 6v submersible pump micro mini submersible water pump 3v to 6vdc water pump for diy dc pump for hobby kit mini submersible pump motor this is a lowcost, small size submersible pump motor which can be operated from a 2.5 ~ 6V power supply. It can take up to 120 liters per hour with very low current consumption of 220ma. Just connect tube pipe to the motor

FEATURES:

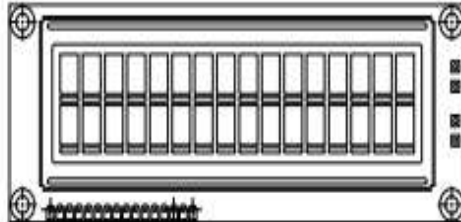
- Voltage: 2.5-6V
- Maximum lift: 40-110cm / 15.75"-43.4"
- Flow rate: 80-120L/H
- Outside diameter: 7.5mm / 0.3"
- Inside diameter: 5mm / 0.2"
- Diameter: Approx. 24mm / 0.95"
- Length: Approx. 45mm / 1.8"
- Height: Approx. 30mm / 1.2"
- Material: Engineering plastic
- Driving mode: DC design, magnetic driving

APPLICATIONS:

- Controlled fountain water flow
- Controlled Garden watering systems
- Hydroponic Systems
- Fresh water intake or exhaust systems for fish aquarium



16 x 2 Character LCD



FEATURES

- Type: Character
- Display format: 16 x 2 characters
- Built-in controller: ST 7066 (or equivalent)
- Duty cycle: 1/16
- 5 x 8 dots includes cursor
- + 5 V power supply (also available for + 3 V)
- LED can be driven by pin 1, pin 2, pin 15, pin 16 or A and K
- N.V. optional for + 3 V power supply
- Material categorization: For definitions of compliance please see www.vishay.com/doc?99912

RoHS
COMPLIANT

MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module Dimension	122.0 x 44.0	mm
Viewing Area	99.0 x 24.0	
Dot Size	0.92 x 1.10	
Dot Pitch	0.98 x 1.16	
Mounting Hole	115.0 x 37.0	
Character Size	4.84 x 9.66	

ABSOLUTE MAXIMUM RATINGS					
ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	V_{DD} to V_{SS}	- 0.3	-	7.0	V
Input Voltage	V_i	- 0.3	-	V_{DD}	

Note

- $V_{SS} = 0$ V, $V_{DD} = 5.0$ V

ELECTRICAL CHARACTERISTICS						
ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT
			MIN.	TYP.	MAX.	
Input Voltage	V_{DD}	$V_{DD} = + 5$ V	4.7	5.0	5.3	V
Supply Current	I_{DD}	$V_{DD} = + 5$ V	-	1.6	1.5	mA
Recommended LC Driving Voltage for Normal Temperature Version Module	V_{DD} to V_0	- 20 °C	-	-	5.2	V
		0 °C	-	-	4.5	
		25 °C	4.2	4.2	-	
		50 °C	3.8	-	-	
		70 °C	3.5	-	-	
LED Forward Voltage	V_F	25 °C	-	4.2	4.6	V
LED Forward Current - Array	I_F	25 °C	-	260	520	mA
EL Power Supply Current	I_{EL}	$V_{EL} = 110$ V _{AC} , 400 Hz	-	-	5.0	mA

OPTIONS									
PROCESS COLOR						BACKLIGHT			
TN	STN Gray	STN Yellow	STN Blue	FSTN B&W	STN Color	None	LED	EL	CCFL
x	x	x	x	x		x	x	x	

For detailed information, please see the "Product Numbering System" document.


www.vishay.com

LCD-016N002L1

Vishay

DISPLAY CHARACTER ADDRESS CODE

Display Position

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DD RAM Address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

INTERFACE PIN FUNCTION

PIN NO.	SYMBOL	FUNCTION
1	V _{SS}	Ground
2	V _{DD}	+ 3 V or + 5 V
3	V ₀	Contrast adjustment
4	RS	H/L register select signal
5	R/W	H/L read/write signal
6	E	H → L enable signal
7	DB0	H/L data bus line
8	DB1	H/L data bus line
9	DB2	H/L data bus line
10	DB3	H/L data bus line
11	DB4	H/L data bus line
12	DB5	H/L data bus line
13	DB6	H/L data bus line
14	DB7	H/L data bus line
15	A/V _{EE}	+ 4.2 V for LED (R _s = 0 Ω)/negative voltage output
16	K	Power supply for B/L (0 V)

DIMENSIONS in millimeters