#include <ESP8266WiFi.h>

#include <DHT.h>  //library sensor yang telah diimportkan
#include <DHT_U.h>

DHT dht( D2, DHT11);  //Pin, Jenis DHT
int kipas = 12;

WiFiClient client;

String MakerIFTTT_Key ;  // string adalah tipe data untuk teks yang merupakan gabungan huruf, angka, spacy, dsb.
;String MakerIFTTT_Event;

char *append_str(char *here, String s) {
  int i=0; while (*here++ = s[i]) {i++;};return here-1;
}

char *append_ul(char *here, unsigned long u) {
  char buf[20]; return append_str(here, ultoa(u, buf, 10));
}

char post_rqst[256];char *p;char *content_length_here;char *json_start;int compi;

void setup() {
  Serial.begin(115200);  //baud komunikasi serial
  WiFi.disconnect();
  delay(3000);
  Serial.println("Start");
  WiFi.begin("Bastian","bastian123");
  while (!(WiFi.status() == WL_CONNECTED)) {
    delay(300);
  }
  dht.begin();  //prosedur memulai pembacaan module sensor
  Serial.println(".................");
}
Serial.println("connected");
}

void loop() {

float kelembaban = dht.readHumidity();  //Pembacaan untuk data kelembaban
float suhu = dht.readTemperature();  //Pembacaan untuk data Suhu

Serial.print("kelembaban: ");
Serial.print(kelembaban);
Serial.print(" ");
Serial.print("suhu: ");
Serial.println(suhu);

delay(2000);
if (suhu <= 24 )
{
  analogWrite(kipas,512);
  if (client.connect("maker.ifttt.com",80)) {
    MakerIFTTT_Key ="gDPLowlR5vDZvsmOXOrVoU7s268FUrLrrhPEqrhhXr";
    MakerIFTTT_Event ="Bastian";
    p = post_rqst;
    p = append_str(p, "POST /trigger/");
    p = append_str(p, MakerIFTTT_Event);
    p = append_str(p, "/with/key/");
    p = append_str(p, MakerIFTTT_Key);
    p = append_str(p, " HTTP/1.1\r\n");
    p = append_str(p, "Host: maker.ifttt.com\r\n");
    p = append_str(p, "Content-Type: application/json\r\n");
    p = append_str(p, "Content-Length: ");
    p = append_str(p, ");
    send(p);
  }
}
}
p = append_str(p, "Content-Type: application/json\r\n\n");

p = append_str(p, "Content-Length: ");

content_length_here = p;

p = append_str(p, "NN\r\n\n\n\n");

json_start = p;

p = append_str(p, "{\"value1\":\""");

p = append_str(p, "Kondisi Suhu Normal");

p = append_str(p, "}\"");

compi = strlen(json_start);

content_length_here[0] = '0' + (compi/10);

content_length_here[1] = '0' + (compi%10);

client.print(post_rqst);

}

Serial.println("Pesan Dikirim............");

Serial.println("Kondisi Suhu Normal");

delay(3000);

}

else if ((suhu >24)&&(suhu < 26))

{

analogWrite(kipas,767);

}

else

{

analogWrite(kipas,1024);

if (client.connect("maker.ifttt.com",80)) 
{
MakerIFTTT_Key ="gDPLow1R5vDZvsmOXOrVpVoU7s268FUrrchPEqrhhXr"
MakerIFTTT_Event ="Bastian"

p = post_rqst;
p = append_str(p, "POST /trigger/");
p = append_str(p, MakerIFTTT_Event);
p = append_str(p, "/with/key/");
p = append_str(p, MakerIFTTT_Key);
p = append_str(p, " HTTP/1.1
Host: maker.ifttt.com
Content-Type: application/json
Content-Length: ");
content_length_here = p;
p = append_str(p, "NN
"
json_start = p;
p = append_str(p, 
"\value1":"");
p = append_str(p, "Kondisi Suhu Buruk");
p = append_str(p, 
"");

compi= strlen(json_start);
content_length_here[0] = '0' + (compi/10);
content_length_here[1] = '0' + (compi%10);
client.print(post_rqst);
}
Serial.println("Pesan Dikirim.............");
Serial.println("Kondisi Suhu Buruk");
delay(3000);
INTRODUCTION TO
NodeMCU ESP8266
DEVKIT v1.0 JULY 2017

www.einstronic.com
NodeMCU ESP8266 ESP-12E WiFi Development Board

NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term “NodeMCU” by default refers to the firmware rather than the DevKit. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.

Features

- Version: DevKit v1.0
- Breadboard Friendly
- Light Weight and small size.
- 3.3V operated, can be USB powered.
- Uses wireless protocol 802.11b/g/n.
- Built-in wireless connectivity capabilities.
- Built-in PCB antenna on the ESP-12E chip.
- Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.
- Uses CP2102 USB Serial Communication interface module.
- Arduino IDE compatible (extension board manager required).
- Supports Lua (alike node.js) and Arduino C programming language.

PINOUT DIAGRAM

NodeMCU ESP8266 v1.0

Source
https://iotbytes.wordpress.com/nodemcu-pinout/
# Specifications of ESP-12E WiFi Module

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Wireless Standard</strong></td>
<td>IEEE 802.11 b/g/n</td>
</tr>
<tr>
<td><strong>Frequency Range</strong></td>
<td>2.412 - 2.484 GHz</td>
</tr>
<tr>
<td><strong>Power Transmission</strong></td>
<td>802.11b : +16 ± 2 dBm (at 11 Mbps)</td>
</tr>
<tr>
<td></td>
<td>802.11g : +14 ± 2 dBm (at 54 Mbps)</td>
</tr>
<tr>
<td></td>
<td>802.11n : +13 ± 2 dBm (at HT20, MCS7)</td>
</tr>
<tr>
<td><strong>Receiving Sensitivity</strong></td>
<td>802.11b : -93 dBm (at 11 Mbps, CCK)</td>
</tr>
<tr>
<td></td>
<td>802.11g : -85 dBm (at 54 Mbps, OFDM)</td>
</tr>
<tr>
<td></td>
<td>802.11n : -82 dBm (at HT20, MCS7)</td>
</tr>
<tr>
<td><strong>Wireless Form</strong></td>
<td>On-board PCB Antenna</td>
</tr>
<tr>
<td><strong>IO Capability</strong></td>
<td>UART, I2C, PWM, GPIO, 1 ADC</td>
</tr>
<tr>
<td><strong>Electrical Characteristic</strong></td>
<td>3.3 V Operated</td>
</tr>
<tr>
<td></td>
<td>15 mA output current per GPIO pin</td>
</tr>
<tr>
<td></td>
<td>12 - 200 mA working current</td>
</tr>
<tr>
<td></td>
<td>Less than 200 uA standby current</td>
</tr>
<tr>
<td><strong>Operating Temperature</strong></td>
<td>-40 to +125 °C</td>
</tr>
<tr>
<td><strong>Serial Transmission</strong></td>
<td>110 - 921600 bps, TCP Client 5</td>
</tr>
<tr>
<td><strong>Wireless Network Type</strong></td>
<td>STA / AP / STA + AP</td>
</tr>
<tr>
<td><strong>Security Type</strong></td>
<td>WEP / WPA-PSK / WPA2-PSK</td>
</tr>
<tr>
<td><strong>Encryption Type</strong></td>
<td>WEP64 / WEP128 / TKIP / AES</td>
</tr>
<tr>
<td><strong>Firmware Upgrade</strong></td>
<td>Local Serial Port, OTA Remote Upgrade</td>
</tr>
<tr>
<td><strong>Network Protocol</strong></td>
<td>IPv4, TCP / UDP / FTP / HTTP</td>
</tr>
<tr>
<td><strong>User Configuration</strong></td>
<td>AT + Order Set, Web Android / iOS, Smart Link APP</td>
</tr>
</tbody>
</table>
Related Sites

NodeMCU official site
http://nodemcu.com/index_en.html

NodeMCU Documentation
https://nodemcu.readthedocs.io/en/master/

NodeMCU Firmware (GitHub)
https://github.com/nodemcu/nodemcu-firmware

Project tagged with NodeMCU, HACKADAY.IO
https://hackaday.io/projects?tag=NodeMCU

ESP8266 Getting started, by ACROBOTIC industries
http://learn.acrobotic.com/tutorials/post/esp8266-getting-started

Quick Start to Nodemcu (ESP8266) on Arduino IDE
by Magesh Jayakumar

GETTING STARTED WITH PLATFORMIO AND ESP8266 NODEMCU
by Brandon Cannaday
https://www.losant.com/blog/getting-started-with-platformio-esp8266-nodemcu

Programming ESP8266 ESP-12E NodeMCU V1.0 With Arduino IDE
Into Wireless Temperature Logger
by Shin Teo

For more details, we can be reached at the addresses below.
Terms & Condition apply.
DHT11 Humidity & Temperature Sensor

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output.
DHT 11 Humidity & Temperature Sensor

1. Introduction

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.
Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor’s internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users’ request.

2. Technical Specifications:

Overview:

<table>
<thead>
<tr>
<th>Item</th>
<th>Measurement Range</th>
<th>Humidity Accuracy</th>
<th>Temperature Accuracy</th>
<th>Resolution</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHT11</td>
<td>20-90%RH 0-50 °C</td>
<td>±5%RH</td>
<td>±2°C</td>
<td>1</td>
<td>4 Pin Single Row</td>
</tr>
</tbody>
</table>
**Detailed Specifications:**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Conditions</th>
<th>Minimum</th>
<th>Typical</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Humidity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>1%RH</td>
<td>1%RH</td>
<td>1%RH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8 Bit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repeatability</td>
<td>±1%RH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>25°C</td>
<td>±4%RH</td>
<td></td>
<td>±5%RH</td>
</tr>
<tr>
<td></td>
<td>0-50°C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Interchangeability</strong></td>
<td>Fully Interchangeable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Measurement Range</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>0°C</td>
<td>30%RH</td>
<td>90%RH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25°C</td>
<td>20%RH</td>
<td>90%RH</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50°C</td>
<td>20%RH</td>
<td>80%RH</td>
<td></td>
</tr>
<tr>
<td><strong>Response Time (Seconds)</strong></td>
<td>1/e(63%)25°C, 1m/s Air</td>
<td>6 S</td>
<td>10 S</td>
<td>15 S</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>±1%RH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Long-Term Stability</td>
<td>Typical</td>
<td>±1%RH/year</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Temperature</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>1°C</td>
<td>1°C</td>
<td>1°C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8 Bit</td>
<td>8 Bit</td>
<td>8 Bit</td>
<td></td>
</tr>
<tr>
<td>Repeatability</td>
<td>±1°C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>±1°C</td>
<td></td>
<td></td>
<td>±2°C</td>
</tr>
<tr>
<td><strong>Measurement Range</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution</td>
<td>0°C</td>
<td></td>
<td>50°C</td>
<td></td>
</tr>
<tr>
<td><strong>Response Time (Seconds)</strong></td>
<td>1/e(63%)</td>
<td>6 S</td>
<td>30 S</td>
<td></td>
</tr>
</tbody>
</table>
3. Typical Application (Figure 1)

![Figure 1 Typical Application](image)

Note: 3Pin – Null; MCU = Micro-computer Unite or single chip Computer

When the connecting cable is shorter than 20 metres, a 5K pull-up resistor is recommended; when the connecting cable is longer than 20 metres, choose a appropriate pull-up resistor as needed.

4. Power and Pin

DHT11’s power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.


Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms.

Data consists of decimal and integral parts. A complete data transmission is **40bit**, and the sensor sends **higher data bit** first.

**Data format:** 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".
5.1 Overall Communication Process (Figure 2, below)
When MCU sends a start signal, DHT11 changes from the low-power-consumption mode to the running-mode, waiting for MCU completing the start signal. Once it is completed, DHT11 sends a response signal of 40-bit data that include the relative humidity and temperature information to MCU. Users can choose to collect (read) some data. Without the start signal from MCU, DHT11 will not give the response signal to MCU. Once data is collected, DHT11 will change to the low-power-consumption mode until it receives a start signal from MCU again.

![Figure 2 Overall Communication Process](image)

**Figure 2 Overall Communication Process**

5.2 MCU Sends out Start Signal to DHT (Figure 3, below)
Data Single-bus free status is at high voltage level. When the communication between MCU and DHT11 begins, the programme of MCU will set Data Single-bus voltage level from high to low and this process must take at least 18ms to ensure DHT’s detection of MCU’s signal, then MCU will pull up voltage and wait 20-40us for DHT’s response.

![Figure 3 MCU Sends out Start Signal & DHT Responses](image)

**Figure 3 MCU Sends out Start Signal & DHT Responses**
5.3 DHT Responses to MCU (Figure 3, above)

Once DHT detects the start signal, it will send out a low-voltage-level response signal, which lasts 80us. Then the programme of DHT sets Data Single-bus voltage level from low to high and keeps it for 80us for DHT’s preparation for sending data.

When DATA Single-Bus is at the low voltage level, this means that DHT is sending the response signal. Once DHT sent out the response signal, it pulls up voltage and keeps it for 80us and prepares for data transmission.

When DHT is sending data to MCU, every bit of data begins with the 50us low-voltage-level and the length of the following high-voltage-level signal determines whether data bit is "0" or "1" (see Figures 4 and 5 below).

![Figure 4 Data "0" Indication](image)

**Figure 4 Data "0" Indication**
If the response signal from DHT is always at high-voltage-level, it suggests that DHT is not responding properly and please check the connection. When the last bit data is transmitted, DHT11 pulls down the voltage level and keeps it for 50us. Then the Single-Bus voltage will be pulled up by the resistor to set it back to the free status.

6. Electrical Characteristics

VDD=5V, T = 25℃ (unless otherwise stated)

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Minimum</th>
<th>Typical</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply DC</td>
<td>3V</td>
<td>5V</td>
<td>5.5V</td>
</tr>
<tr>
<td>Current Supply</td>
<td>Measuring</td>
<td>0.5mA</td>
<td>2.5mA</td>
</tr>
<tr>
<td>Average</td>
<td>0.2mA</td>
<td>1mA</td>
<td></td>
</tr>
<tr>
<td>Standby</td>
<td>100uA</td>
<td></td>
<td>150uA</td>
</tr>
<tr>
<td>Sampling period</td>
<td>Second</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Note: Sampling period at intervals should be no less than 1 second.

7. Attentions of application

(1) Operating conditions

Applying the DHT11 sensor beyond its working range stated in this datasheet can result in 3%RH signal shift/discrepancy. The DHT11 sensor can recover to the calibrated status gradually when it gets back to the normal operating condition and works within its range. Please refer to (3) of
this section to accelerate its recovery. Please be aware that operating the DHT11 sensor in the non-normal working conditions will accelerate sensor’s aging process.

(2) Attention to chemical materials
Vapor from chemical materials may interfere with DHT’s sensitive-elements and debase its sensitivity. A high degree of chemical contamination can permanently damage the sensor.

(3) Restoration process when (1) & (2) happen
Step one: Keep the DHT sensor at the condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;
Step two: Keep the DHT sensor at the condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Temperature Affect
Relative humidity largely depends on temperature. Although temperature compensation technology is used to ensure accurate measurement of RH, it is still strongly advised to keep the humidity and temperature sensors working under the same temperature. DHT11 should be mounted at the place as far as possible from parts that may generate heat.

(5) Light Affect
Long time exposure to strong sunlight and ultraviolet may debase DHT’s performance.

(6) Connection wires
The quality of connection wires will affect the quality and distance of communication and high quality shielding-wire is recommended.

(7) Other attentions
* Welding temperature should be bellow 260Celsius and contact should take less than 10 seconds.
* Avoid using the sensor under dew condition.
* Do not use this product in safety or emergency stop devices or any other occasion that failure of DHT11 may cause personal injury.
* Storage: Keep the sensor at temperature 10-40°C, humidity <60%RH.

Disclaimer
This is a translated version of the manufacturer's data sheet. OSEPP is not responsible for the accuracy of the translated information.
Mouser Electronics

Authorized Distributor

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

OSEPP Electronics:
HUMI-01