LAMPIRAN

FAKULTAS TEKNOLOGI INDUST
Universitas Islam Sultan Agung (UNISSUL
Jl. Raya Kaligawe Km.4 Telp. 024-6583584 Psw. 340 Faks. 024-6582
Semarang 50112 http://www.unissula.a

LEMBAR REVISI dan TUGAS UJIAN SARJANA

Berdasarkan Rapat Tim Penguji Ujian Sarjana

Hari	:	Selasa
Tanggal	:	26 Maret 2019
Tempat	:	R Sidang

Memutuskan bahwa mahasiswa :

Nama	:	Muhammad Muhlish
NIM	:	030601401564
Judul TA	:	Monitoring Suhu dan Kelembaban Udara pada Pembibitan Stroberi Berbasis Internet Of Things

wajib melakukan perbaikan dan membuat tugas seperti tercantum dibawah ini:

NO	REVISI	BATAS REVISI
1	Tulisan Gambar, Tabel bukan bold	1 minggu
2	Teori diperdalam, cara kerja sensor.	
3	Pembahasan MCU mulai yang lengkap	
4	Tulisan asing Italic	
5	Bab dan gambar ? di cek [illegible]	
6	Lampiran : Data Sheet	
7	Rangkaian lengkap (lampiran)	
8	[illegible] tabel / gbr	

Acc

NO	TUGAS

Semarang, 26 Maret 2019
Penguji, I

Ir. Budi Pramono Jati, MM, MT
NIDN. 0623126501

Mengetahui,
Ketua Tim Penguji

Ir. Budi Pramono Jati, MM, MT
NIDN. 0623126501

FAKULTAS TEKNOLOGI INDUST
Universitas Islam Sultan Agung (UNISSUL
Jl. Raya Kaligawe Km.4 Telp. 024-6583584 Psw. 340 Faks. 024-6582-
Semarang 50112 http://www.unissula.a

LEMBAR REVISI dan TUGAS UJIAN SARJANA

Berdasarkan Rapat Tim Penguji Ujian Sarjana

Hari	:	Selasa
Tanggal	:	26 Maret 2019
Tempat	:	R Sidang

Memutuskan bahwa mahasiswa :

Nama	:	Muhammad Muhlish
NIM	:	030601401564
Judul TA	:	Monitoring Suhu dan Kelembaban Udara pada Pembibitan Stroberi Berbasis Internet Of Things

wajib melakukan perbaikan dan membuat tugas seperti tercantum dibawah ini:

NO	REVISI	BATAS REVISI
	- judul - rumusan masalah - Tujuan - kesimpulan	Acc

NO	TUGAS

Mengetahui,
Ketua Tim Penguji

Ir. Budi Pramono Jati, MM, MT
NIDN. 0623126501

Semarang, 26 Maret 2019
Penguji, III

Ir.H. Sukarno Budi Utomo, MT
NIDN. 0619076401

FAKULTAS TEKNOLOGI INDUST
Universitas Islam Sultan Agung (UNISSUL
Jl. Raya Kaligawe Km.4 Telp. 024-6583584 Psw. 340 Faks. 024-6582
Semarang 50112 http://www.unissula.a

LEMBAR REVISI dan TUGAS UJIAN SARJANA

Berdasarkan Rapat Tim Penguji Ujian Sarjana

Hari	:	Selasa
Tanggal	:	26 Maret 2019
Tempat	:	R Sidang

Memutuskan bahwa mahasiswa :

Nama	:	Muhammad Muhlish
NIM	:	030601401564
Judul TA	:	Monitoring Suhu dan Kelembaban Udara pada Pembibitan Stroberi Berbasis Internet Of Things

wajib melakukan perbaikan dan membuat tugas seperti tercantum dibawah ini:

NO	REVISI	BATAS REVISI
	- Dasar Teori ditambah / dilengkapi - Flow cart. - Foto Rangkaian / dokumentasi - Daftar Pustaka - Cara Registrasi Thingspeak	Acc 5/4/2019 Munaf.

NO	TUGAS

Mengetahui,
Ketua Tim Penguji

Ir. Budi Pramono Jati, MM, MT
NIDN. 0623126501

Semarang, 26 Maret 2019
Penguji, II

Munaf Ismail, ST, MT
NIDN. 210616054

PEMANTAUAN SUHU DAN KELEMBABAN UDARA PADA PEMBIBITAN STROBERI BERBASIS IOT

by muhammad muhlish

Dosen Pembimbing I	Dosen Pembimbing II
(Agus Suprajitno, S.T.,M.T.)	(Jenny Putri Hapsari S.T.,M.T)

Submission dat e : 08- Apr- 2019 08.4 6AM (UT C+0800)
Submission ID: 1097 154 638
File name : jilid_2.pdf (996.61K)
Word count : 9015
Charact e r count : 5834 9

PEMANTAUAN SUHU DAN KELEMBABAN UDARA PADA PEMBIBITAN STROBERI BERBASIS IOT

ORIGINALITY REPORT

20%	18%	0%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

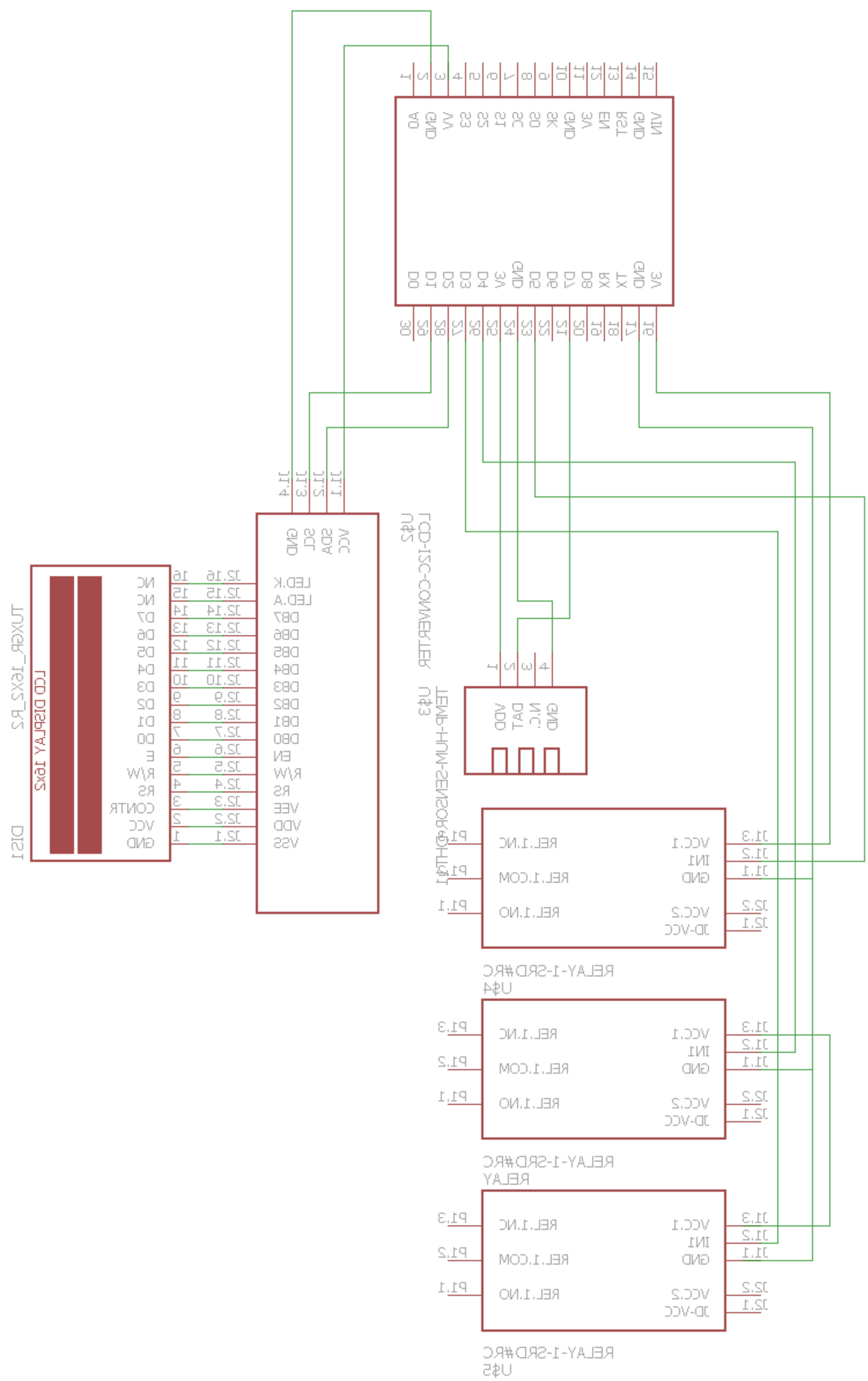
1	Submitted to Sultan Agung Islamic University Student Paper	3%
2	Submitted to Universitas Brawijaya Student Paper	2%
3	mybtechprojects.tech Internet Source	1%
4	Submitted to Jordan Matthews High School Student Paper	1%
5	Submitted to Universitas Muria Kudus Student Paper	1%
6	eprints.umm.ac.id Internet Source	1%
7	repository.usu.ac.id Internet Source	1%
8	Submitted to Universiti Malaysia Pahang Student Paper	1%
9	media.neliti.com	

```
#include <DHT.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ThingSpeak.h>
#include <Wire.h>  // I2C library already built in Arduino IDE
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7);
#define DHTPIN 13  //data dht11 di pin D7
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
const char* ssid = "Tugasakhir";  //nama wifi
const char* password = "kupukupuh";  //password wifi
WiFiClient client;
unsigned long myChannelNumber = 7195081;  //channel id
const char * myWriteAPIKey = "D9TOU5SVRPTIF4N7";  //write Api Key
uint8_t temperature, humidity;  //deklarasi suhu dan kelembaban
int in = 14; //relay1
int in1 = 0 ; //relay2
int in2 = 2 ; //relay3
void setup()
{
  Serial.begin(115200);
  lcd.begin(16,2);   // inisialisasi LCD 16 x 2
  lcd.setBacklightPin(3,POSITIVE); // Enable or Turn On the backlight
  lcd.setBacklight(HIGH);
  dht.begin();  //inisialisai dht11
  pinMode (in, OUTPUT); //output relay1 low triggered
```

```
  digitalWrite(in, HIGH);
    pinMode (in1, OUTPUT); //output relay2 high triggered
      digitalWrite(in1, LOW);
        pinMode (in2, OUTPUT); //output relay3 high triggered
          digitalWrite(in2, LOW);

 Serial.println();
 Serial.print("Connecting to ");  //menghubungkan ke wifi
 Serial.println(ssid);
 WiFi.begin(ssid, password);
 while (WiFi.status() != WL_CONNECTED)
 {
  delay(500);
  Serial.print(".");
 }
 Serial.println("");
 Serial.println("WiFi connected");
 Serial.println(WiFi.localIP());  //menampilkan IP
 ThingSpeak.begin(client);
}
void loop()
{
 lcd.home();
 lcd.print("Suhu : ");
 lcd.print(dht.readTemperature()); //LCD Suhu
 lcd.print(" C");
 lcd.setCursor(0, 1);
```

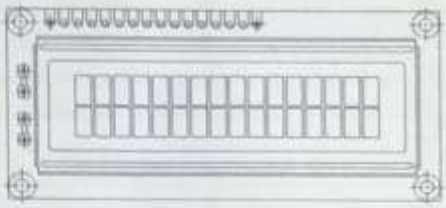
```
lcd.print("Kel  : ");
lcd.print(dht.readHumidity()); //LCD kelembaban
lcd.print(" %");
delay (200);
static boolean data_state = false;
temperature = dht.readTemperature();  //deklarasi suhu
humidity = dht.readHumidity();  //deklarasi kelembaban
Serial.print("Suhu :");  //serial monitor suhu
Serial.print(temperature);
Serial.println(" C");
Serial.print("Kelembaban :");  //serial monitor kelembaban udara
Serial.print(humidity);
Serial.println(" %");
if ( temperature >27){ //jika suhu lebih besar dari 27
  digitalWrite(in1, HIGH); //relay menyala
  digitalWrite(in2, HIGH);
}

else {
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
}
 if ( humidity <66){ //jika kelembaban lebih besar dari 66
  digitalWrite(in, LOW); //relay menyala
}

else {
```

```
    digitalWrite(in, HIGH);
  }
  // Write to ThingSpeak. There are up to 8 fields in a channel, allowing you to store up to 8 different
  // pieces of information in a channel. Here, we write to field 1.
  if( data_state )
  {
   ThingSpeak.writeField(myChannelNumber, 1, temperature, myWriteAPIKey);
   data_state = false;
  }
  else
  {
   ThingSpeak.writeField(myChannelNumber, 2, humidity, myWriteAPIKey);
   data_state = true;
  }
  delay(30000); //waktu yang diperlukan minimald untuk mengirim data di sett 30d
}
```



LCD-I2C-CONVERTER
U$2
TEMP-HUM-SENSOR
U$3
RELAY-1-SRD#RC
U$4
RELAY-1-SRD#RC
RELAY
RELAY-1-SRD#RC
U$5
LCD DISPLAY 16x2
TUXGR_16X2_R2
DIS1

16 x 2 Character LCD



FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A,K (LED)
- N.V. optional for + 3V power supply

MECHANICAL DATA

ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0	mm
Viewing Area	66.0 x 16.0	mm
Dot Size	0.56 x 0.66	mm
Character Size	2.96 x 5.56	mm

ABSOLUTE MAXIMUM RATING

ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	- 0.3	–	7.0	V
Input Voltage	VI	- 0.3	–	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS

ITEM	SYMBOL	CONDITION		STANDARD VALUE			UNIT
				MIN.	TYP.	MAX.	
Input Voltage	VDD	VDD = + 5V		4.7	5.0	5.3	V
		VDD = + 3V		2.7	3.0	5.3	V
Supply Current	IDD	VDD = 5V		–	1.2	3.0	mA
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - V0	- 20 °C		–	–	–	V
		0°C		4.2	4.8	5.1	
		25°C		3.8	4.2	4.6	
		50°C		3.6	4.0	4.4	
		70°C		–	–	–	
LED Forward Voltage	VF	25°C		–	4.2	4.6	V
LED Forward Current	IF	25°C	Array	–	130	260	mA
			Edge	–	20	40	
EL Power Supply Current	IEL	Vel = 110VAC:400Hz		–	–	5.0	mA

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														0F
DD RAM Address	40	41														4F

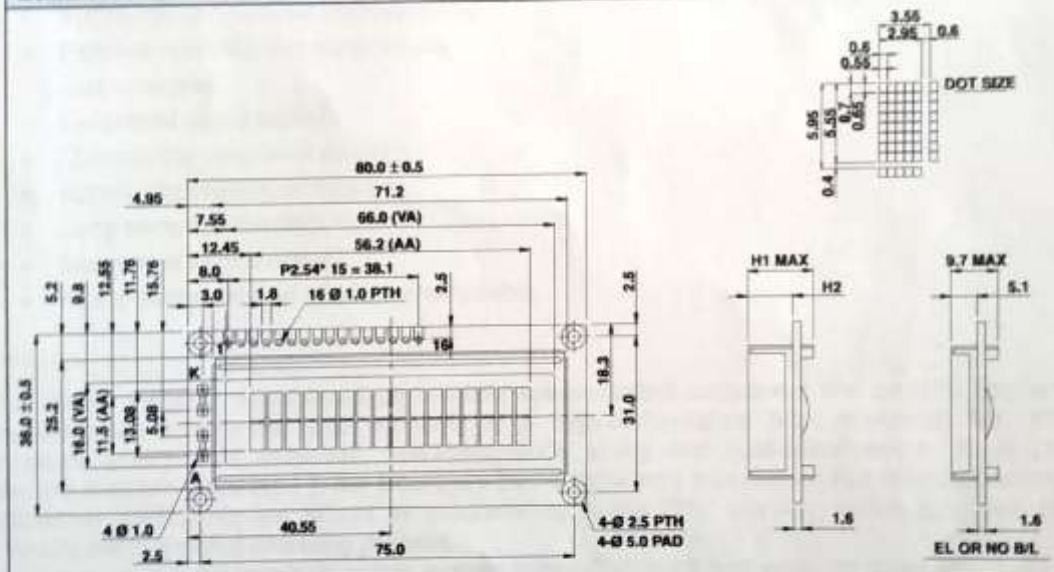
LCD-016M002B

Vishay

16 x 2 Character LCD

PIN NUMBER	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	+ 3V or + 5V
3	Vo	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/$\overline{W}$	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/Vee	+ 4.2V for LED/Negative Voltage Output
16	K	Power Supply for B/L (0V)

DIMENSIONS in millimeters



LED - H/L B/L		
	HIGH	LOW
H1	13.2	12.1
H2	8.6	7.5

SUNROM TECHNOLOGIES

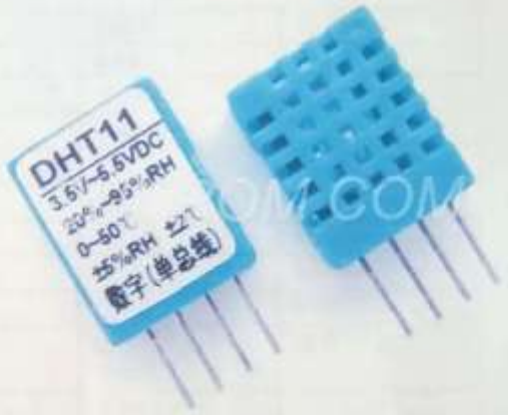
For more products visit our website http://www.sunrom.com

Document: Datasheet | Date: 20-Jun-12 | Model #: 3732 | Product's Page: www.sunrom.com/p-1141.html

DHT11 - Humidity and Temperature Sensor

The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed).

Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds.



Features

- 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

Details

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.

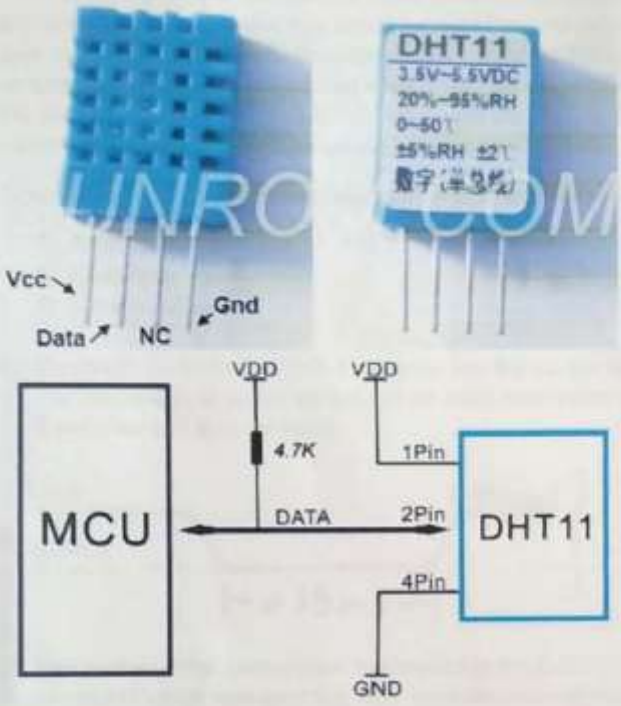
Specifications

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5%RH	±2℃	1	4 Pin Single Row

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3	5	5.5	V
Current supply	Measuring	0.5		2.5	mA
	Stand-by	100	Null	150	uA
	Average	0.2	Null	1	mA

Typical Application



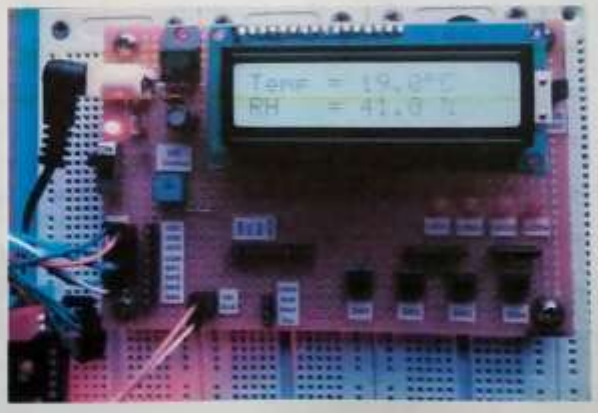
DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

SDK (Software Development Kit)

Download source code + project articles by clicking following link

http://www.sunrom.com/files/3732.zip

It contains details for AVR, PIC and Arduino projects.



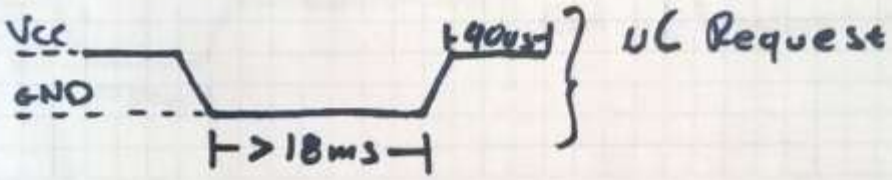
Communication Process: Serial Interface (Single-Wire Two-Way)

The interesting thing in this module is the protocol that uses to transfer data. All the sensor readings are sent using a single wire bus which reduces the cost and extends the distance. In order to send data over a bus you have to describe the way the data will be transferred, so that transmitter and receiver can understand what says each other. This is what a protocol does. It describes the way the data are transmitted. On DHT-11 the 1-wire data bus is pulled up with a resistor to VCC. So if nothing is occurred the voltage on the bus is equal to VCC.

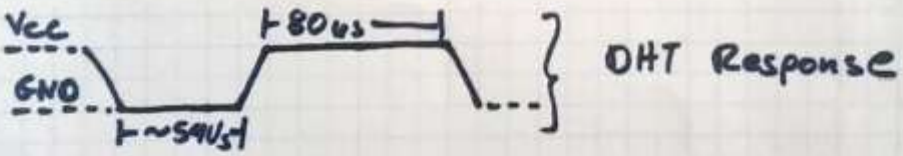
Communication Format can be seperated into three stages

1) Request
2) Response
3) Data Reading

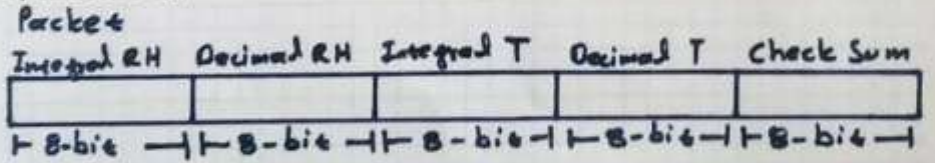
1) **Request:** To make the DHT-11 to send you the sensor readings you have to send it a request. The request is, to pull down the bus for more than **18ms** in order to give DHT time to understand it and then pull it up for **40uS**.



2) **Response:** What comes after the request is the DHT-11 response. This is an automatic reply from DHT which indicates that DHT received your request. The response is ~54uS low and 80uS high.



3) **Data Reading:** What will come after the response is the sensor data. The data will be packed in a packet of 5 segments of 8-bits each. Totally 5×8 =40bits.



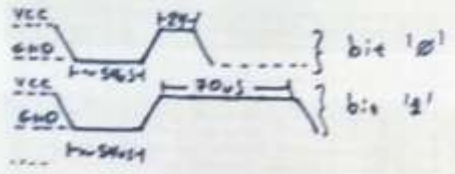
First two segments are Humidity read, integral & decimal. Following two are Temperature read in Celsius, integral & decimal and the last segment is the Check Sum which is the sum of the 4 first

segments. If Check Sum's value isn't the same as the sum of the first 4 segments that means that data received isn't correct.

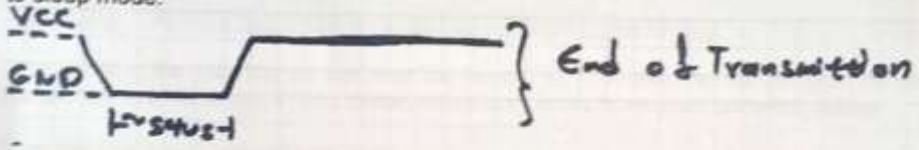
How to Identify Bits: Each bit sent is a follow of ~54uS Low in the bus and ~24uS to 70uS High depending on the value of the bit.

Bit '0' : ~54uS Low and ~24uS High

Bit '1' : ~54uS Low and ~70uS High



End Of Frame: At the end of packet DHT sends a ~54uS Low level, pulls the bus to High and goes to sleep mode.



Logic Analyzer Snapshots: In the following image you can see the request sent from the MCU to the DHT and following the packet. Because the request has very long duration as you can see is about 20mS and packet received is in uS we can't view the data bits. So it is exapanded in next view.

If we zoom at the data bits we can read the values. You can see after the Request follows the Response, and Data bits. I have drawn some color notes to be more understandable.

If we decode the above data we have.

Humidity 0b00101011.0b00000000 = 43.0% (43 is integral part and .0 is decimal part)

Temperature 0b00010111 = 23 C.

The last two segments can't be seen in this image because of zoom.

Implementation:

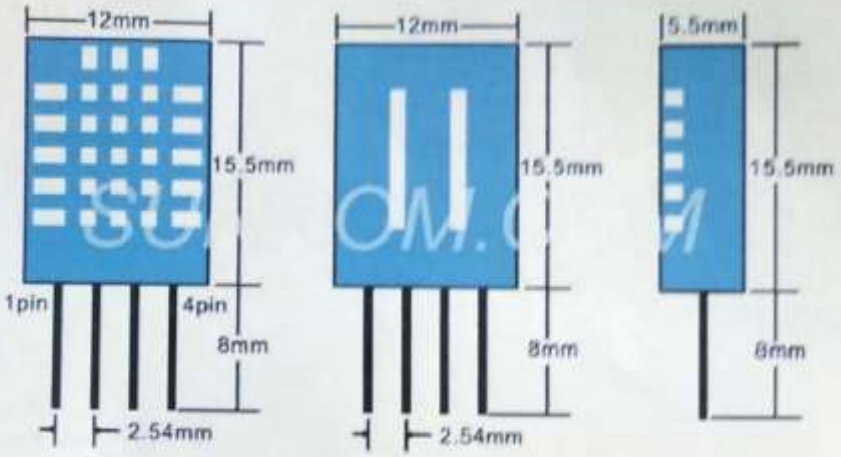
What we have to do to read a DHT-11 sensor is:

1) Send request
2) Read response
3) Read each data segment and save it to a buffer
4) Sum the segments and check if the result is the same as CheckSum

If the CheckSum is correct, the values are correct so we can use them. If CheckSum is wrong we discard the packet.

To read the data bits can use a counter and start count uSeconds of High level. For counts > 24uS we replace with bit '1'. For counts <=24 we replace with bit'0'

Dimensions (mm)

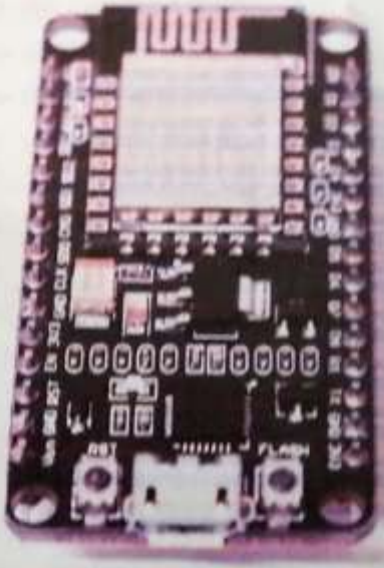


EINSTRONIC
TURN ON THE FUTURE

INTRODUCTION TO

NodeMCU ESP8266

DEVKIT v1.0 JULY 2017



www.einstronic.com

NodeMCU ESP8266

Front View

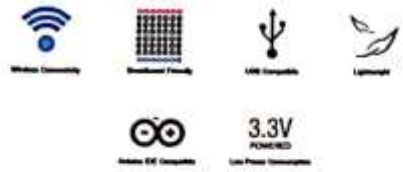
Front View

Specifications of ESP-12E WiFi Module

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

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.

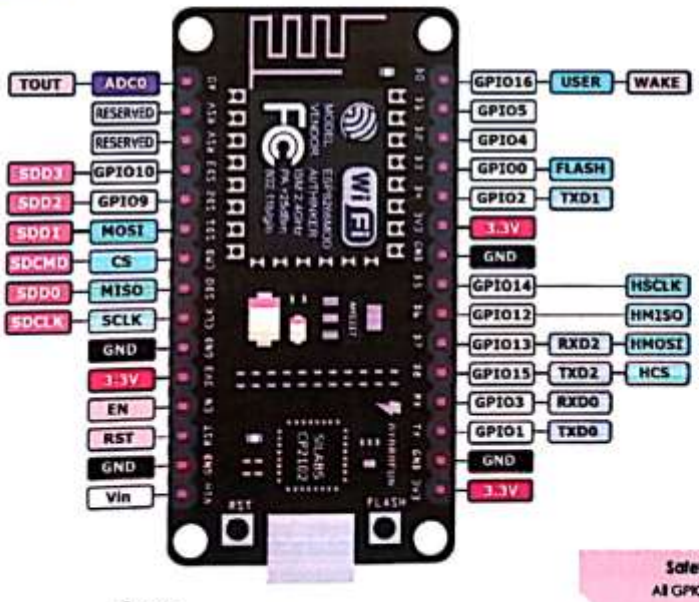


Features

- 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



Safety Precaution
All GPIO runs at 3.3V !!

Source
https://iotbytes.wordpress.com/nodemcu-pinout/

TEC1-12706

Performance Specifications

Hot Side Temperature (°C)	25°C	50°C
Qmax (Watts)	50	57
Delta Tmax (°C)	66	75
Imax (Amps)	6.4	6.4
Vmax (Volts)	14.4	16.4
Module Resistance (Ohms)	1.98	2.30

Performance curves: