Rees52

Planting The Seed of Innovation

Make a IOT weather station kit using DHT22 and OLED Interfacing with

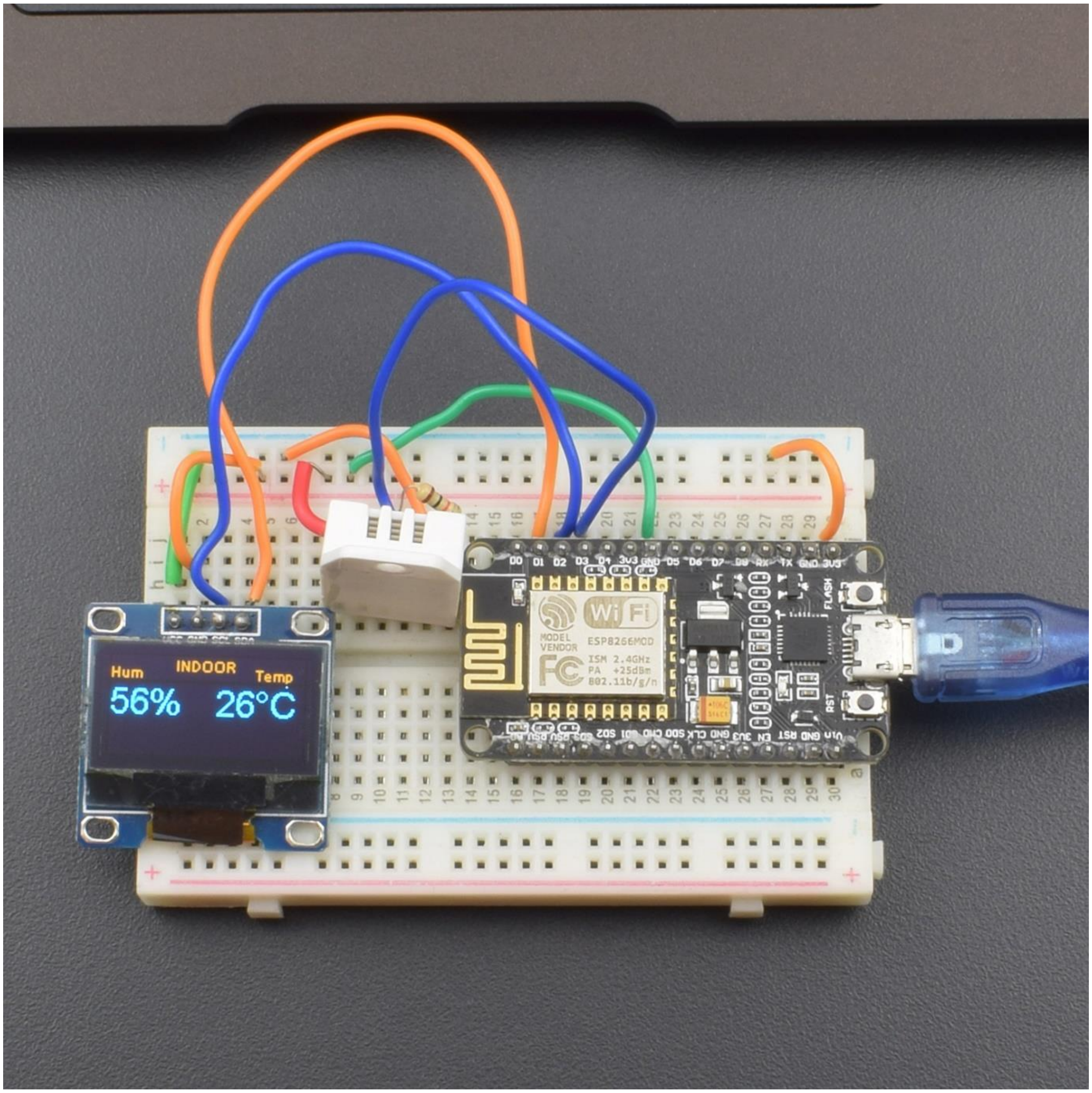
ESP8266-12E

EVERYTHING You NEED
for your first ARDUINO project

WWW.REES52.COM

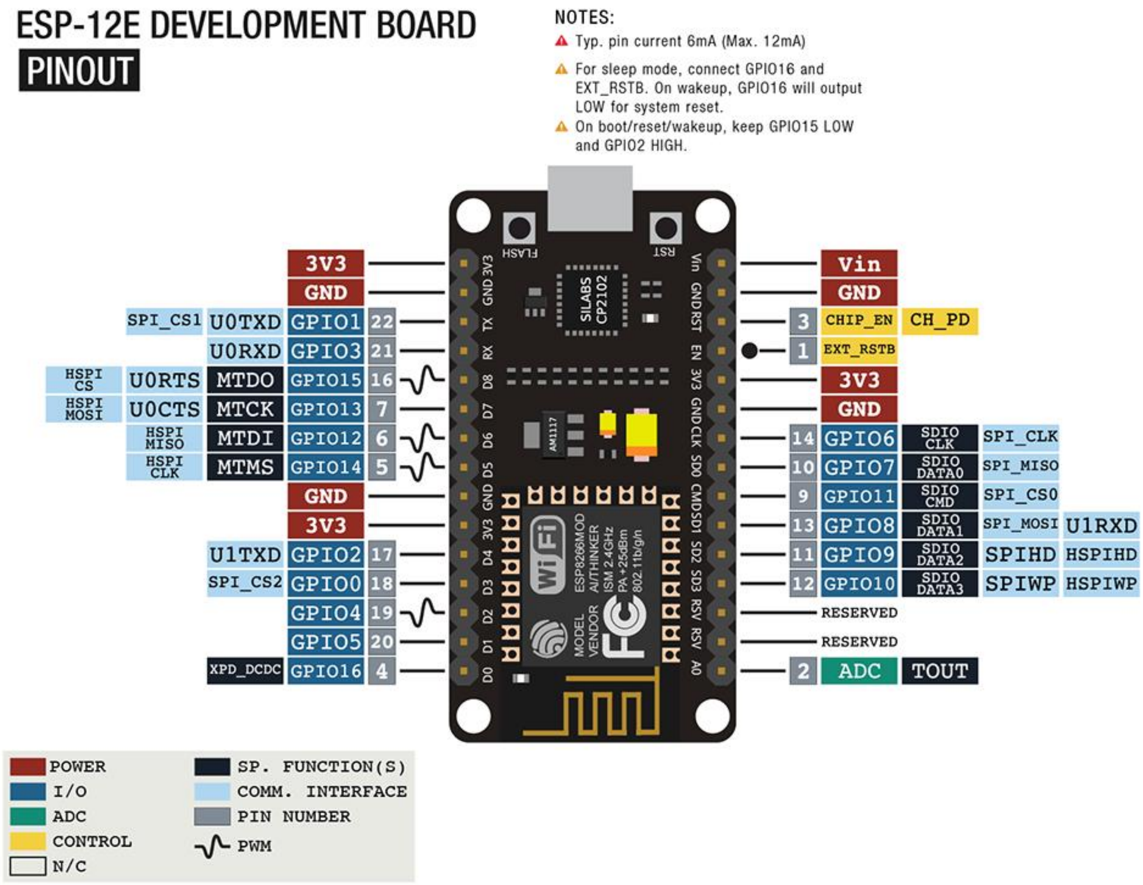
Technical support : +919599594520

Make an IOT Weather station kit using DHT22 and OLED interfacing with ESP8266-12E



SPECIFICATIONS

NODE MCU ESP8266 – 12E CP2102



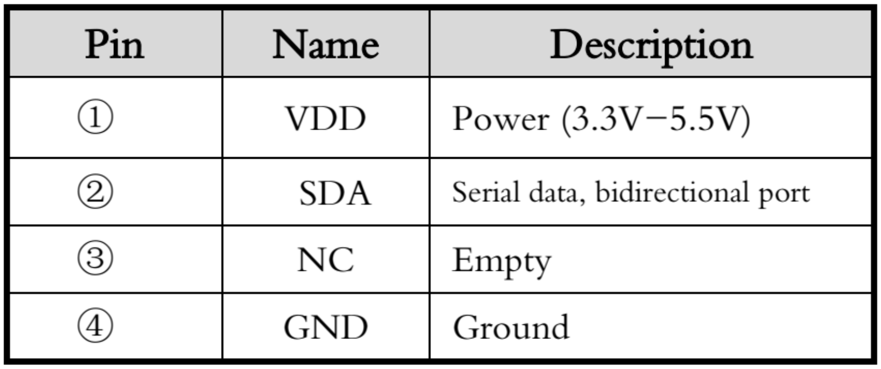
ESP-12E DEVELOPMENT BOARD
PINOUT
NOTES:
Typ. pin current 6mA (Max. 12mA)
For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.
FLASH
RST
SILABS CP2102
AM1117
ESP8266MOD
AI-THINKER
ISM 2.4GHz
PA +25dBm
802.11b/g/n
MODEL
VENDOR
FC
WiFi
3V3
GND
SPI_CS1 U0TXD GPIO1 22 TX
U0RXD GPIO3 21 RX
HSPI CS U0RTS MTDO GPIO15 16 D8
HSPI MOSI U0CTS MTCK GPIO13 7 D7
HSPI MISO MTDI GPIO12 6 D6
HSPI CLK MTMS GPIO14 5 D5
GND
3V3
U1TXD GPIO2 17 D4
SPI_CS2 GPIO0 18 D3
GPIO4 19 D2
GPIO5 20 D1
XPD_DCDC GPIO16 4 D0
Vin Vin
GND GND
RST 3 CHIP_EN CH_PD
EN 1 EXT_RSTB
3V3 3V3
GND GND
CLK 14 GPIO6 SDIO CLK SPI_CLK
SD0 10 GPIO7 SDIO DATA0 SPI_MISO
CMD 9 GPIO11 SDIO CMD SPI_CS0
SD1 13 GPIO8 SDIO DATA1 SPI_MOSI U1RXD
SD2 11 GPIO9 SDIO DATA2 SPIHD HSPIHD
SD3 12 GPIO10 SDIO DATA3 SPIWP HSPIWP
RSV RESERVED
RSV RESERVED
A0 2 ADC TOUT
POWER
I/O
ADC
CONTROL
N/C
SP. FUNCTION(S)
COMM. INTERFACE
PIN NUMBER
PWM

OLED LCD Display

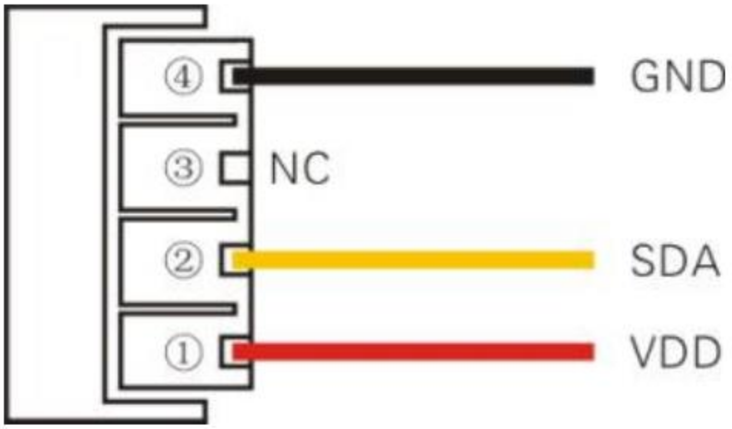
- Voltage: 3V ~ 5V DC. Working Temperature: -30 ℃ ~ 70 ℃
- Ultra-low power consumption: full screen lit 0.08W
- Supports many control chip: Fully compatible with Arduino, 51 Series, MSP430 Series, STM32 / 2, CSR IC, etc.
- High resolution: 128 * 64. Viewing angle:> 160 °

DHT22 AM2302 Temperature and Humidity Sensor

- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 0-100% humidity readings with 2-5% accuracy
- Good for -40 to 125°C temperature readings ±0.5°C accuracy
- No more than 0.5 Hz sampling rate (once every 2 seconds)
- Body size 15.1mm x 25mm x 7.7mm
- 4 pins with 0.1" spacing



Pin	Name	Description
①	VDD	Power (3.3V−5.5V)
②	SDA	Serial data, bidirectional port
③	NC	Empty
④	GND	Ground



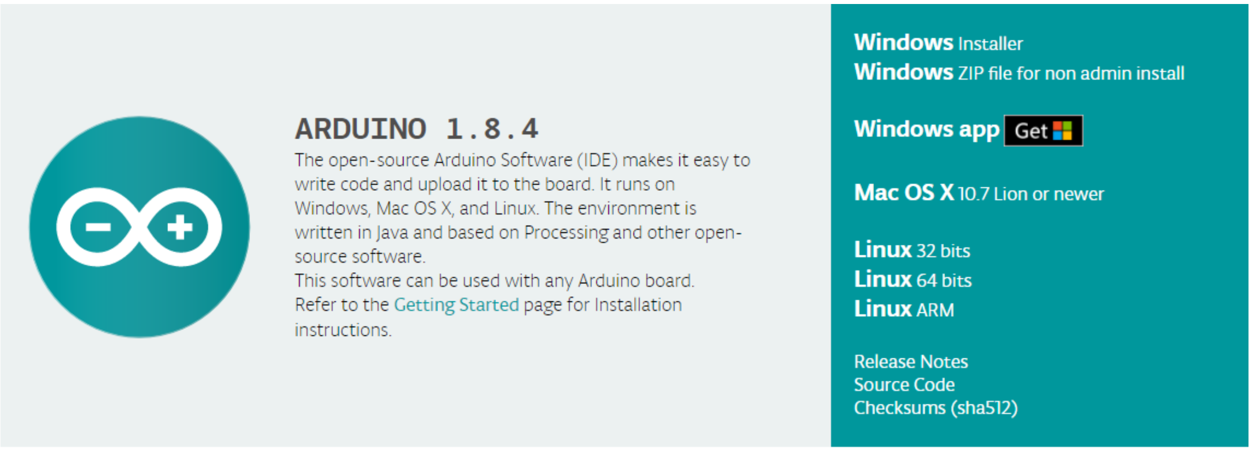
HARDWARE REQUIRED

- NodeMCU ESP8266-12E

- 0.96" I2C IIC SPI Serial 128X64 White OLED LCD LED Display Module
- DHT22 AM2302 Temperature And Humidity Sensor
- Mini Breadboard
- Male-Female DuPont Cables
- External 5V power Supply or battery

SOFTWARE REQUIRED

Arduino IDE ((Programmable platform for Arduino boards) you can download it from this link: https://www.arduino.cc/en/Main/Software



ARDUINO 1.8.4

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

Windows app Get

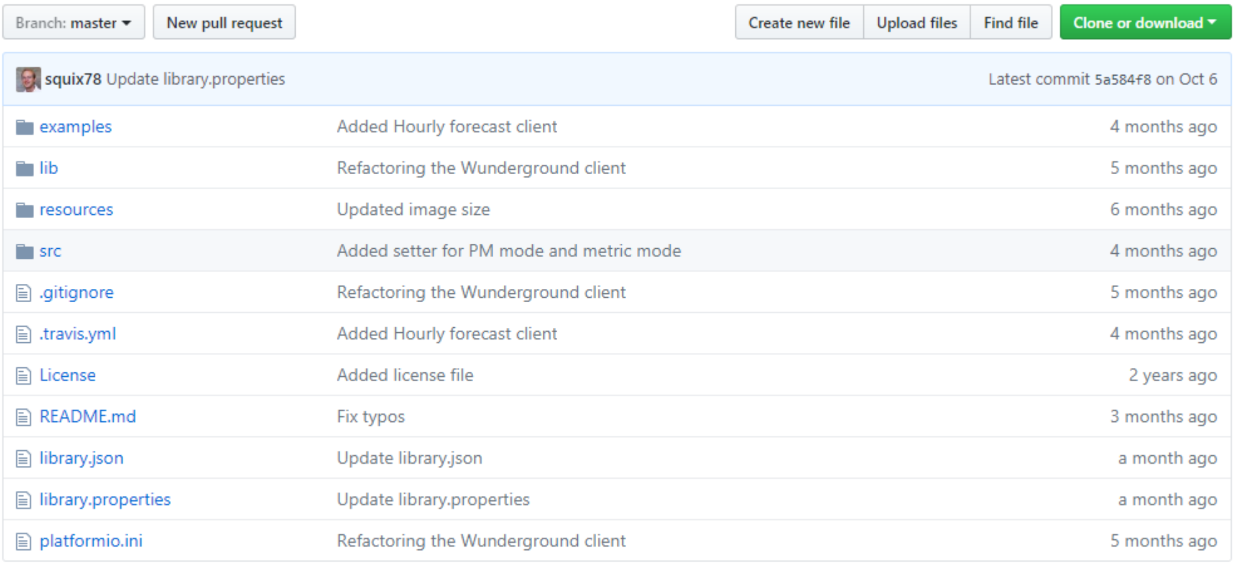
Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

Release Notes
Source Code
Checksums (sha512)

INSTALL LIBRARY

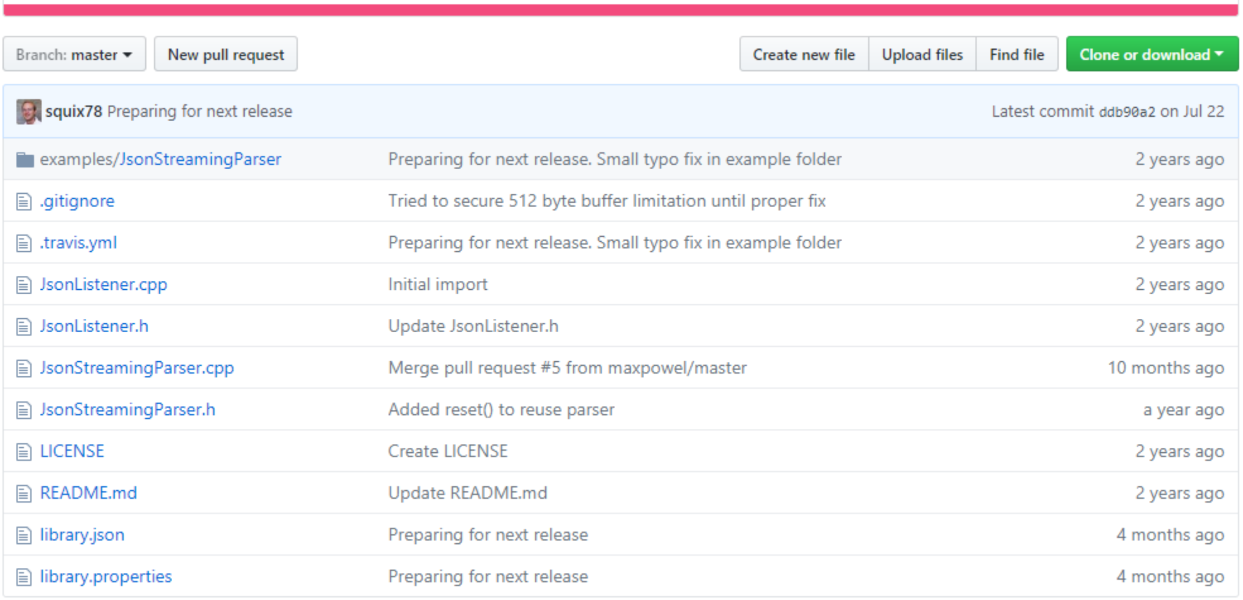
Weather Station by Daniel Eichhorn: https://github.com/squix78/esp8266-weather-station



squix78 Update library.properties — Latest commit 5a584f8 on Oct 6

examples	Added Hourly forecast client	4 months ago
lib	Refactoring the Wunderground client	5 months ago
resources	Updated image size	6 months ago
src	Added setter for PM mode and metric mode	4 months ago
.gitignore	Refactoring the Wunderground client	5 months ago
.travis.yml	Added Hourly forecast client	4 months ago
License	Added license file	2 years ago
README.md	Fix typos	3 months ago
library.json	Update library.json	a month ago
library.properties	Update library.properties	a month ago
platformio.ini	Refactoring the Wunderground client	5 months ago

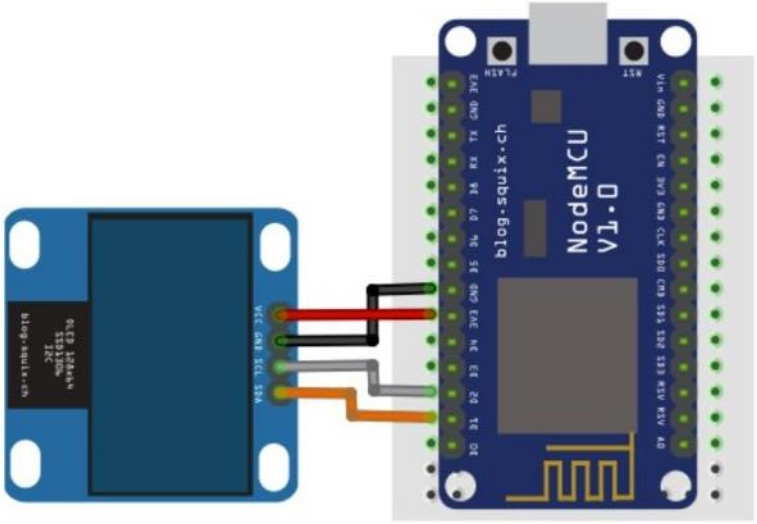
Json Streaming Parser by Daniel Eichhorn: https://github.com/squix78/json-streaming-parser



squix78 Preparing for next release — Latest commit ddb90a2 on Jul 22

examples/JsonStreamingParser	Preparing for next release. Small typo fix in example folder	2 years ago
.gitignore	Tried to secure 512 byte buffer limitation until proper fix	2 years ago
.travis.yml	Preparing for next release. Small typo fix in example folder	2 years ago
JsonListener.cpp	Initial import	2 years ago
JsonListener.h	Update JsonListener.h	2 years ago
JsonStreamingParser.cpp	Merge pull request #5 from maxpowel/master	10 months ago
JsonStreamingParser.h	Added reset() to reuse parser	a year ago
LICENSE	Create LICENSE	2 years ago
README.md	Update README.md	2 years ago
library.json	Preparing for next release	4 months ago
library.properties	Preparing for next release	4 months ago

Installing the OLED on NODEMCU

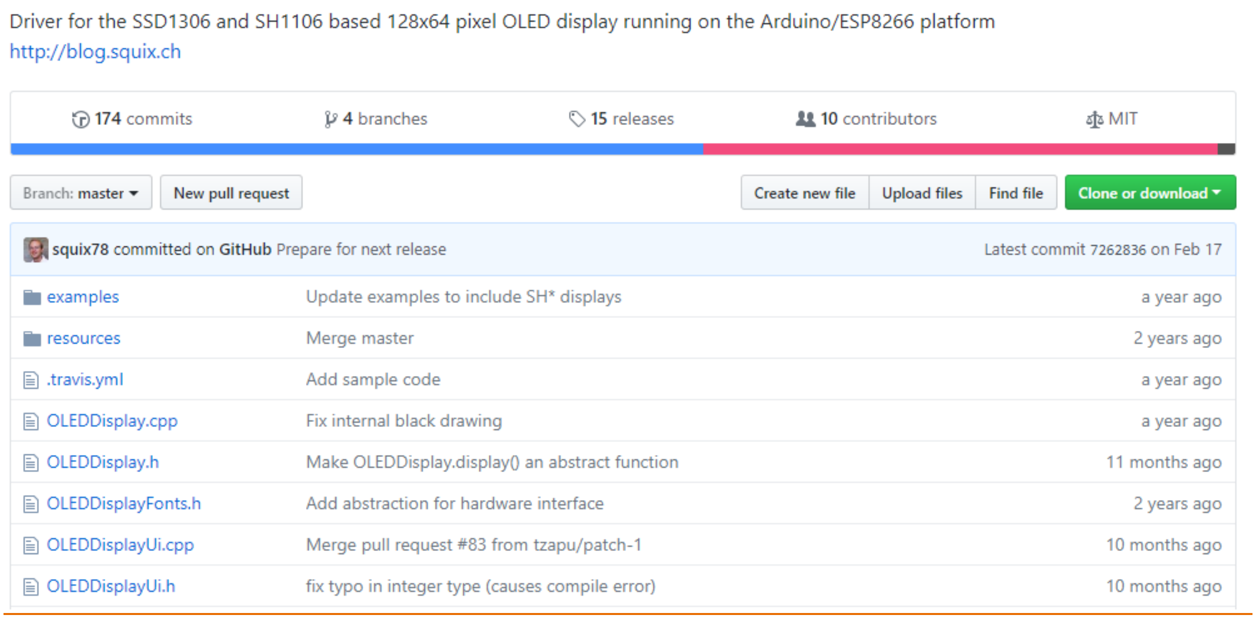


- Connect the OLED pins to the NodeMCU, as described below and shown at above electrical diagram:

  - SDA ==> D1 (5)
  - SCL* ==> D2 (4) * Also you can find "SDC" in the text
  - VCC ==> The SSD1306 can be powered with 5V (external) or 3.3V directly from the NodeMCU module.
  - GND ==> GND

- Once we have connected the display, let's download and install its library on our Arduino IDE.
- Below the library that must be downloaded and installed on your Arduino IDE:
-

https://github.com/squix78/esp8266-oled-ssd1306



Driver for the SSD1306 and SH1106 based 128x64 pixel OLED display running on the Arduino/ESP8266 platform
http://blog.squix.ch

174 commits · 4 branches · 15 releases · 10 contributors · MIT

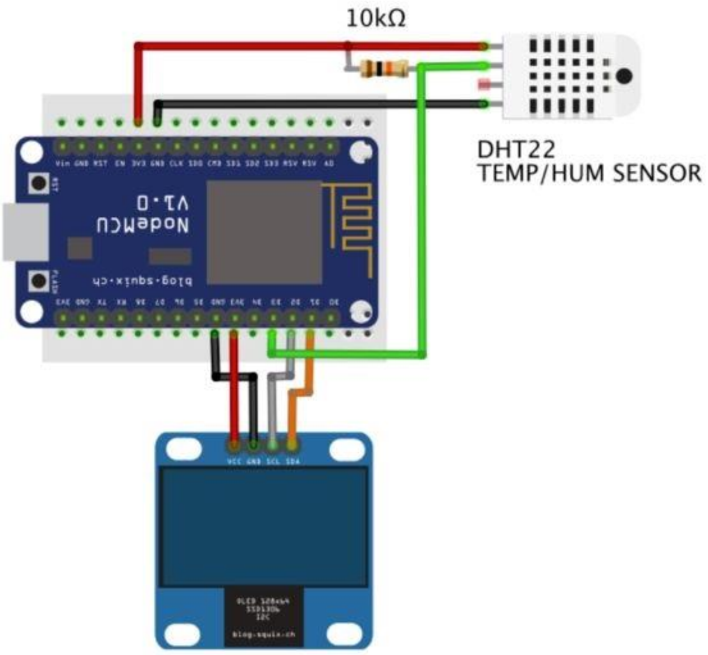
Branch: master ▾ | New pull request | Create new file | Upload files | Find file | Clone or download ▾

squix78 committed on GitHub Prepare for next release — Latest commit 7262836 on Feb 17

examples	Update examples to include SH* displays	a year ago
resources	Merge master	2 years ago
.travis.yml	Add sample code	a year ago
OLEDDisplay.cpp	Fix internal black drawing	a year ago
OLEDDisplay.h	Make OLEDDisplay.display() an abstract function	11 months ago
OLEDDisplayFonts.h	Add abstraction for hardware interface	2 years ago
OLEDDisplayUi.cpp	Merge pull request #83 from tzapu/patch-1	10 months ago
OLEDDisplayUi.h	fix typo in integer type (causes compile error)	10 months ago

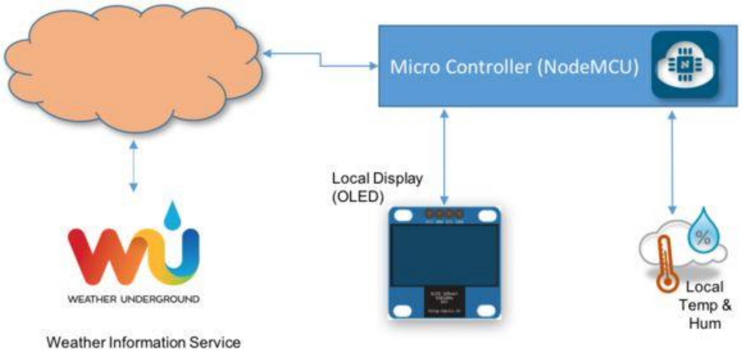
- Once you re-started the IDE, the library should be already installed.
- The library supports I2C protocol to access the OLED display using the built in Wire.h library:

Step 3: Getting Indoor Data

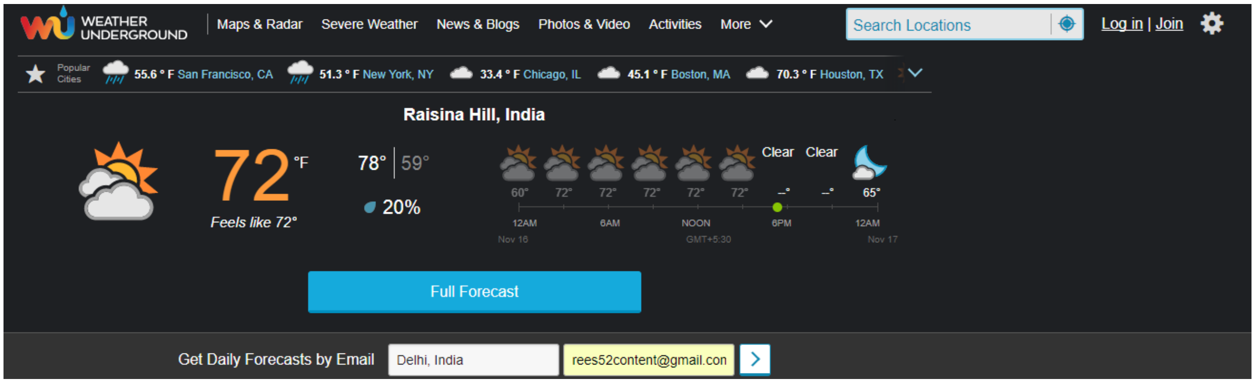


- Once usually you will use the sensor on distances less than 20m, a 10K ohm resistor should be connected between Data and VCC pins. The Output pin will be connected to NodeMCU pin D3 (see the diagram above).

- Once the sensor is installed at our module, download the DHT library from [Adafruit github repository]() and install it in your Arduino's Library file. Once you reload your Arduino IDE, the "DHT sensor library" should be installed.



- Click to download Outdoor weather data by clicking on this [Weather Underground]().



Get Daily Forecasts by Email: Delhi, India | rees52content@gmail.con

- You will need to create an account on their website and get an Weather API key. Do it following the instructions at bellow link:

  [https://www.wunderground.com/weather/api]()

A WEATHER API DESIGNED FOR DEVELOPERS

Reliable data, accurate forecast, & global coverage in 80 languages. About Our API

Key Features

- JSON or XML formatted
- Easy key management
- Error-logging tools
- Rate-monitoring tools
- Complete geo location service
- Autocomplete API to power geo-search
- Worldwide weather stations network

WunderMap Layers

CODE

Modified by

TECHNICAL TEAM,REES52

```
/* Hello World OLED Test */


#include <Wire.h>                  // Only needed for Arduino 1.6.5 and earlier

#include "SSD1306.h"                // alias for `#include "SSD1306Wire.h"`

SSD1306  display(0x3c, 5, 4);      // Initialize the OLED display using Wire library


/* DHT22 */

#include "DHT.h"

#define DHTPIN D3

#define DHTTYPE DHT22
```

```
DHT dht(DHTPIN, DHTTYPE);
int localHum = 0;
int localTemp = 0;
void setup()
{
 Serial.begin(115200);
 display.init();              // Initialising the UI will init the display too.
 display.flipScreenVertically();
}void loop()
{
 getDHT();
 display.clear();
 drawDHT();
 display.display();
 delay (2000);
}
/**************************************************
 Get indoor Temp/Hum data
***************************************************/
void getDHT()
{
 float tempIni = localTemp;
 float humIni = localHum;
 localTemp = dht.readTemperature();
 localHum = dht.readHumidity();
 if (isnan(localHum) || isnan(localTemp))   // Check if any reads failed and exit early (to try again).
 {
  Serial.println("Failed to read from DHT sensor!");
```

```
    localTemp = tempIni;
    localHum = humIni;
    return;
  }}
/***************************************************
  Draw Indoor Page
****************************************************/
void drawDHT()
{
  int x = 0;
  int y = 0;
  display.setFont(ArialMT_Plain_10);
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.drawString(0 + x, 5 + y, "Hum");
 display.setFont(ArialMT_Plain_10);
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.drawString(43 + x, y, "INDOOR");
display.setFont(ArialMT_Plain_24);
  String hum = String(localHum) + "%";
  display.drawString(0 + x, 15 + y, hum);
  int humWidth = display.getStringWidth(hum);
   display.setFont(ArialMT_Plain_10);
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.drawString(95 + x, 5 + y, "Temp");
  display.setFont(ArialMT_Plain_24);
  String temp = String(localTemp) + "°C";
  display.drawString(70 + x, 15 + y, temp);
  int tempWidth = display.getStringWidth(temp);}
```

Rees52

Planting The Seed of Innovation

EVERYTHING You NEED
for your first ARDUINO project

Rees52

G-9 Om Vihar, Phase 5, Block-G, Uttam nagar,
New delhi 110059

Mobile: 011-65355152, +919599594524