



Indoor Air Quality with ESP32 Board

K0068

User guide

Rev 1.0 Apr 2017

Table of contents

1. Kit Introduction.....	3
2. Assembly	5
2.1 ESP32 development board	5
2.2 IAQ TVOC Sensor Module	6
2.3 DHT11 Humidity & Temperature Sensor	9
2.4 0.96" 128X64 Yellow Blue OLED Display	10
2.5 20pin jumper wires	11
3. Install Arduino ESP32 support on Windows	11
4. Build your IAQ project.....	14
4.1 Circuit connection diagram.....	14
4.2 Program and code	14
4.3 Data Logging at Thing Speak.....	14

1. Kit Introduction

This is a complete kit for measuring the quality of indoor air. With all of the components of the kit, you will be able to start your innovations with ESP32 development board.

You can use this kit to measuring the following three parameters that determine the quality of the indoor air.

- Temperature in degree Celsius
- Humidity in Percentage
- Total Volatile Organic Compound (TVOC) in Parts per Billion ppb

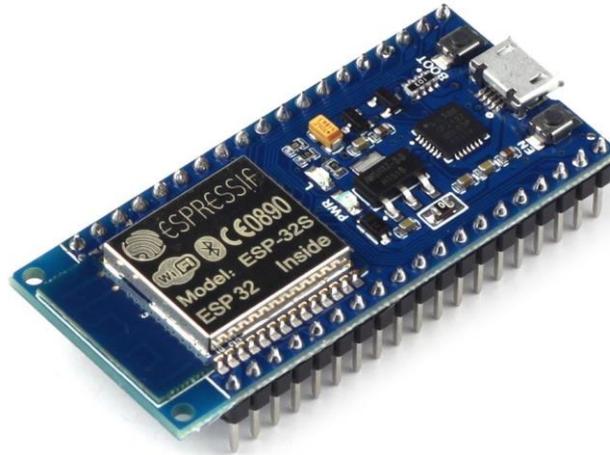
To help customers easily learn how to use the kit, we provide specific introduction of the components and code library as well as the steps of operation in this tutorial. Moreover, there are illustrations of each process for your reference.

If you need any further support, please feel free to contact us at sales@uctronics.com, our engineers will do utmost to help you.



2. Assembly

2.1 ESP32 development board



UCTRONICS ESP32 development board is the latest hybrid WIFI and Bluetooth (BLE4.2) system module. It has a 32bit dual core 240MHz processor and supports WIFI and BLE4.2 connection. The highlight of this development board is fully supporting the Arduino programming IDE, which makes it easier to develop the projects about ESP32 and do more innovations. With the low power consumption and rich connectivity peripherals, it is the ideal prototype solution for wearable and IoT applications.

Specifications of ESP32 dev board:

- Supply Voltage: 2.2V ~ 3.6V
- Operating temperature: -40°C ~ 85°C
- Operating current: 80mA
- Support LWIP protocol, Free RTOS
- Support three modes: AP, STA, and AP+STA
- Support: Lua program

- Processor: 32bit dual-core 240 MHz Tensilica LX6
- SRAM: 520 KB
- Flash: 16 M Byte
- Wi-Fi Standard: FCC/CE/TELEC/KCC
- Wi-Fi Protocol: 802.11 b/g/n/d/e/i/k/r
- Bluetooth Protocol: BLEv4.2 BR/EDR
- Bluetooth Audio: CVSD and SBC
- Peripheral: UART, SPI, I2S, ADC, DAC, I2C
- On-Chip Sensors: Hall Effect, Temperature
- Module dimension: 51.6mmx28.5mm

2.2 iAQ TVOC Sensor Module



The iAQ TVOC sensor module is used to detect the indoor air quality. It can measure TVOC levels and provide TVOC equivalent predictions. This sensor support serial communication and you can read the data via UART. So, it's very easy for you to get the

data from the sensor.

The TVOC sensor module will get into the electric preheat steady state for 3 minutes when powered on. The time of preheat will be affected by the environment and temperature. The lower temperature, the longer time of preheat. The sensor module will correct the ambient temperature and humidity to neutralize the drift error of the sensor. Besides the data shown is the largest. The sensor will output data stably after 3 minutes.

The module has two transfer modes: active and passive transfer modes.

When the module is powered on, the default mode is active transfer. When received the command: 0XF5 0XF0 0X30 0X15, it will stop active data transfer and return the TVOC data: 0X5F 0Xnn 0Xnn CHECKSUM.

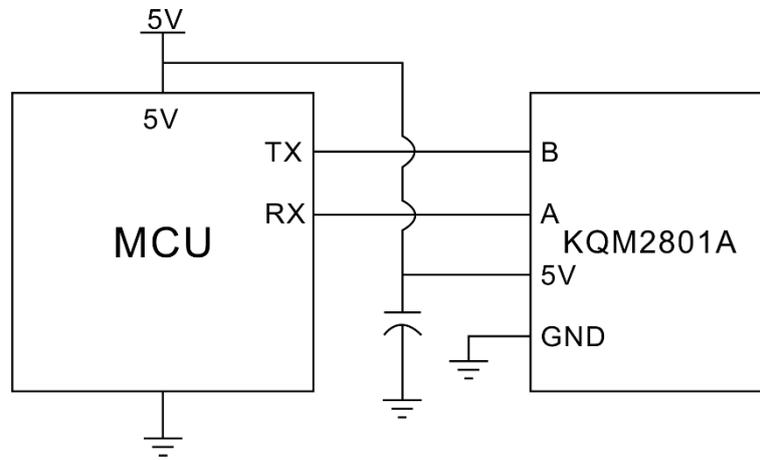
If you want to make it to active mode again, you can send the command: 0XF5 0XF0 0X31 0X16, then it will show the data: 0X5F 0XF5 0X31 0X80, which means that the module is in active transfer mode.

Note: When the detection value is higher than 30ppm, it will reach saturation and the value won't increase.

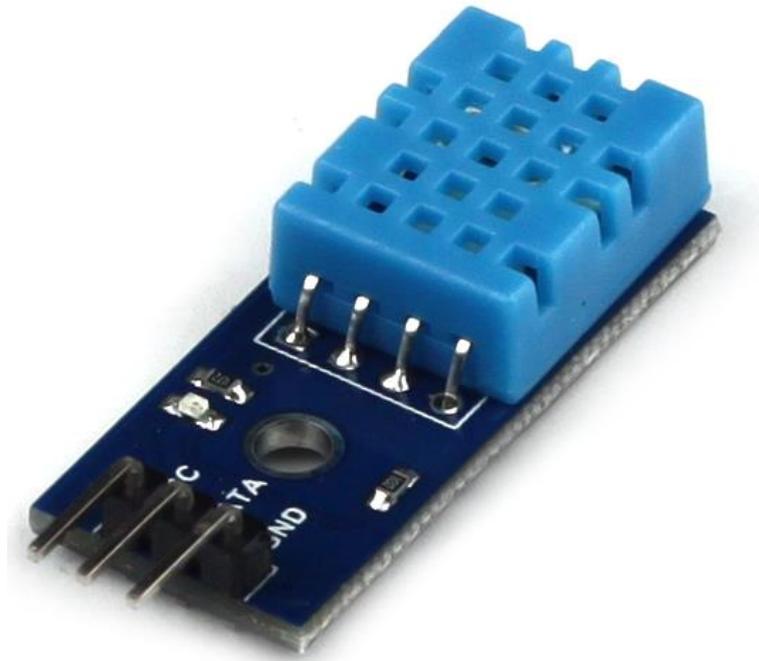
Specification of the IAQ TVOC sensor module:

- Input Voltage: DC 5V \pm 0.2V
- Working current: 80mA \pm 5mA
- Working temperature: -20°C ~ 50°C
- Output signal options: UART
- Output rate: 9600bps
- Clock speed: 100 kHz

- Dimensions: PCB 20 x 24 mm
- Sensing range: 0.1-30 ppm TVOC equivalents (relative)



2.3 DHT11 Humidity & Temperature Sensor



This DHT11 sensor module 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.

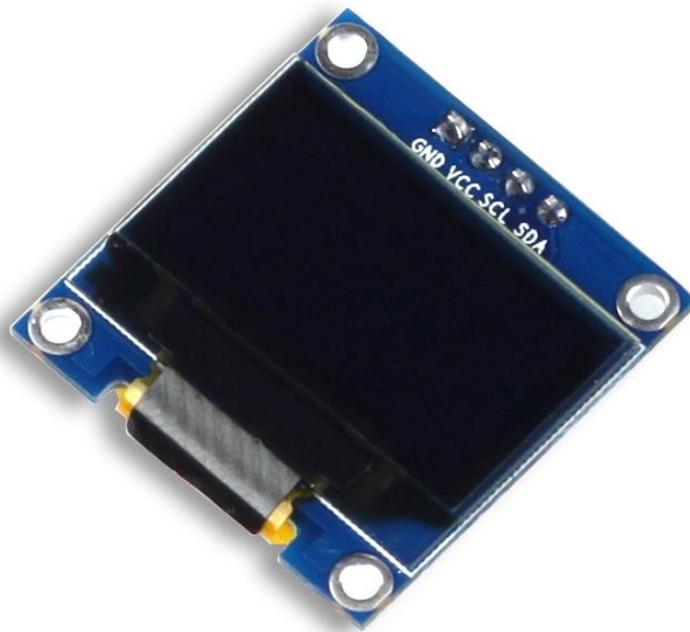
There are three pins of the sensor module: VCC, GND and DATA. Besides the VCC and GND, you just need to use one GPIO pin when wiring to make use of the DHT11 module.

Specifications of DHT11 Temp & Humi sensor module:

- Humidity measurement range: 20% ~95%
- Humidity measurement error: $\pm 5\%$

- Temperature measurement range: 0°C~50°C
- Temperature measurement error: ± 2 °C
- Operating voltage: 3.3 V~5 V
- PCB Dimension: 32 mm * 14 m

2.4 0.96" 128X64 Yellow Blue OLED Display



This OLED display module is small, only 0.96" diagonal, it is made of 128x64 individual yellow and blue OLED pixels, each one is turn on or off by the controller chip. It works without backlight. In dark environment, OLED display is higher compared to that of LCD display you will like the miniature for its crispness.

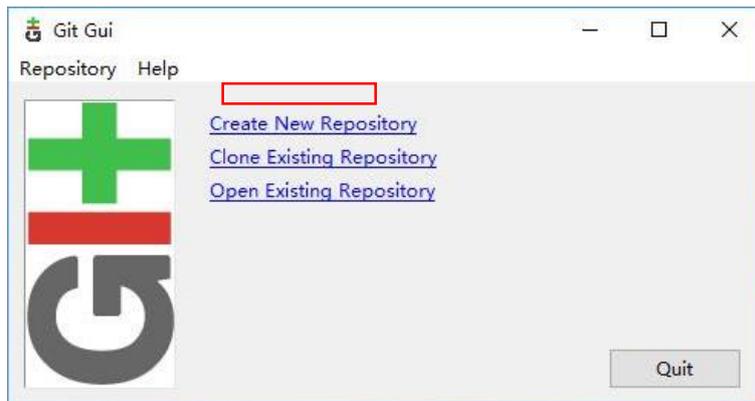
The Driver chip of this OLED is SSD1306, which is compatible with IIC communication. So this module can be controlled by I2C. Besides the VCC and GND, 2 wires would be needed when using 4-wires I2C mode. There is also a simple switch-cap charge pump that turns 5v into a low voltage drive for the OLEDs, making this module the easiest ways to get an

1) Download and install the latest Arduino IDE Windows Installer from arduino.cc

2) Download and install Git from git-scm.com

3) Start Git GUI and run through the following steps:

Select Clone Existing Repository



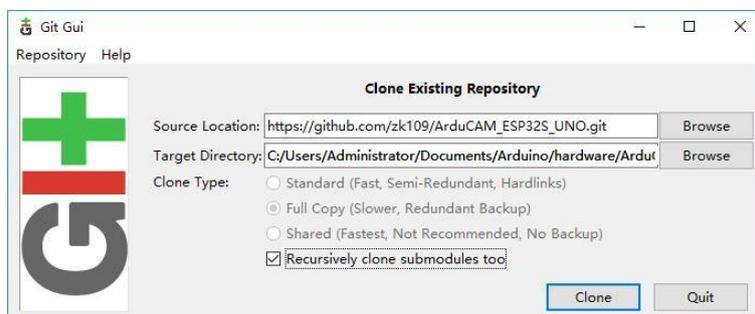
Select source and destination

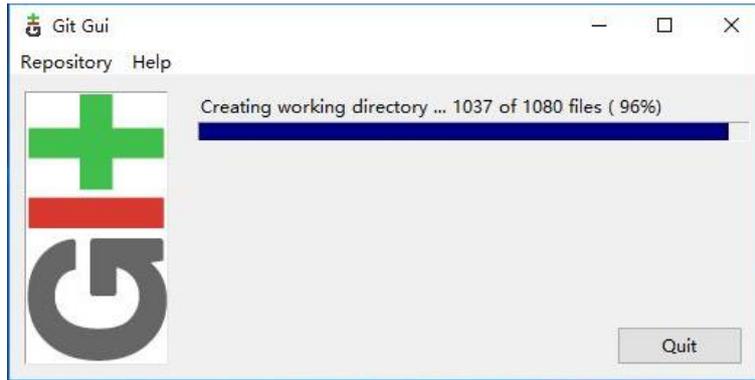
Source Location: https://github.com/ArduCAM/ArduCAM_ESP32S_UNO.git

Target Directory:

C:/Users/[YOUR_USER_NAME]/Documents/Arduino/hardware/ArduCAM/ArduCAM_ESP32S_UNO

Click Clone to start cloning the repository

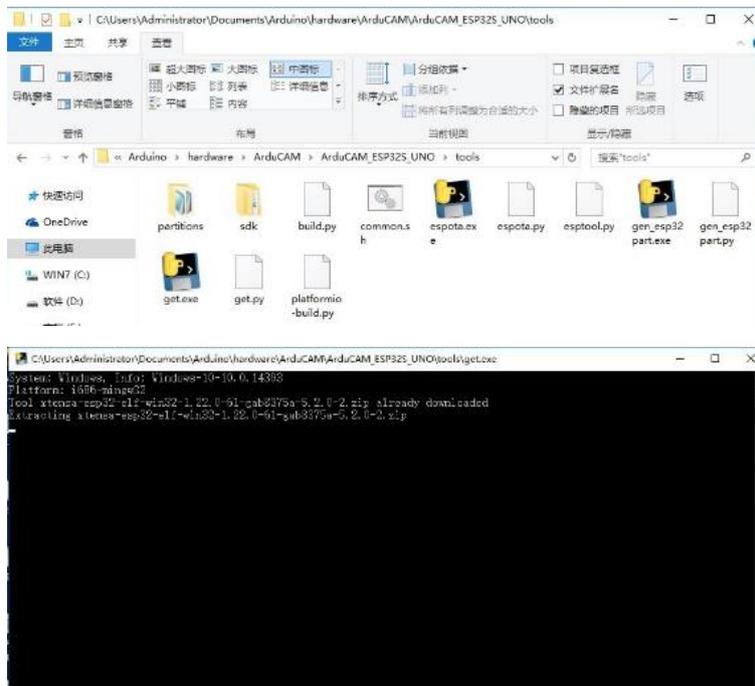




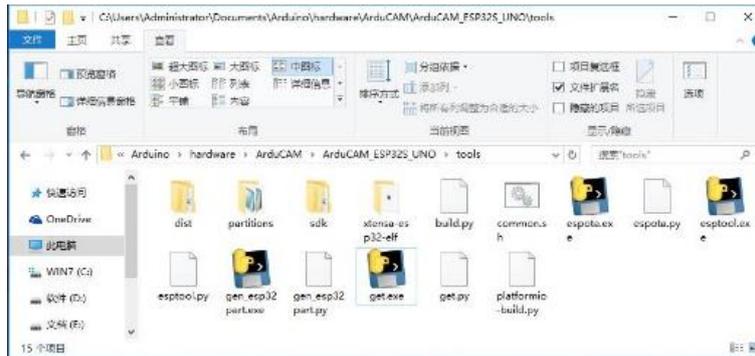
Open

C:/Users/[YOUR_USER_NAME]/Documents/Arduino/hardware/espessif/esp32/tools and

double-click get.exe



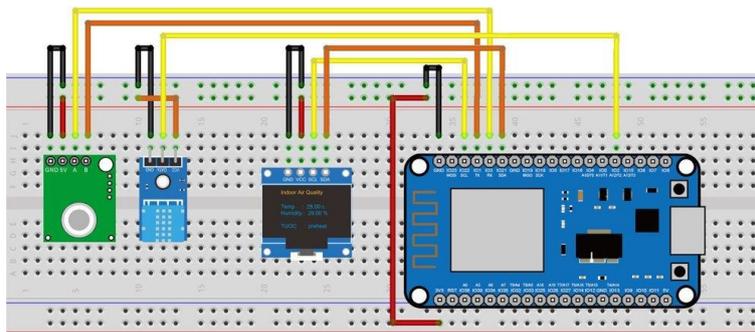
When get.exe finishes, you should see the following files in the directory



4) Plug your ESP32 board and wait for the drivers to install (or install manually any that might be required)

4. Build your IAQ project

4.1 Circuit connection diagram



4.2 Program and code

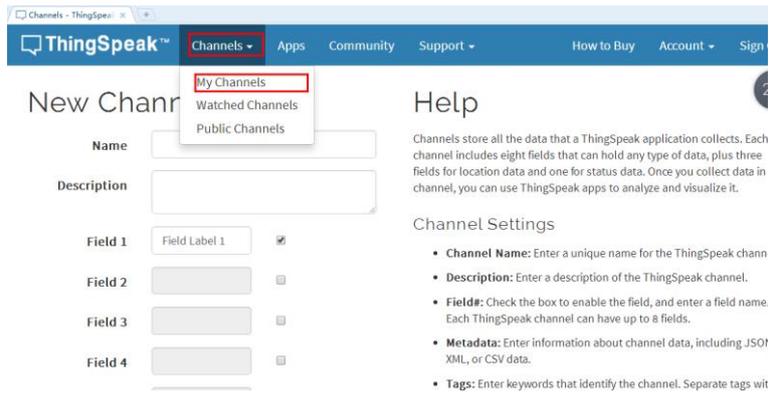
Please check the following link for Arduino code:

https://github.com/UCTRONICS/Arducam_ESP32_TVOC.git

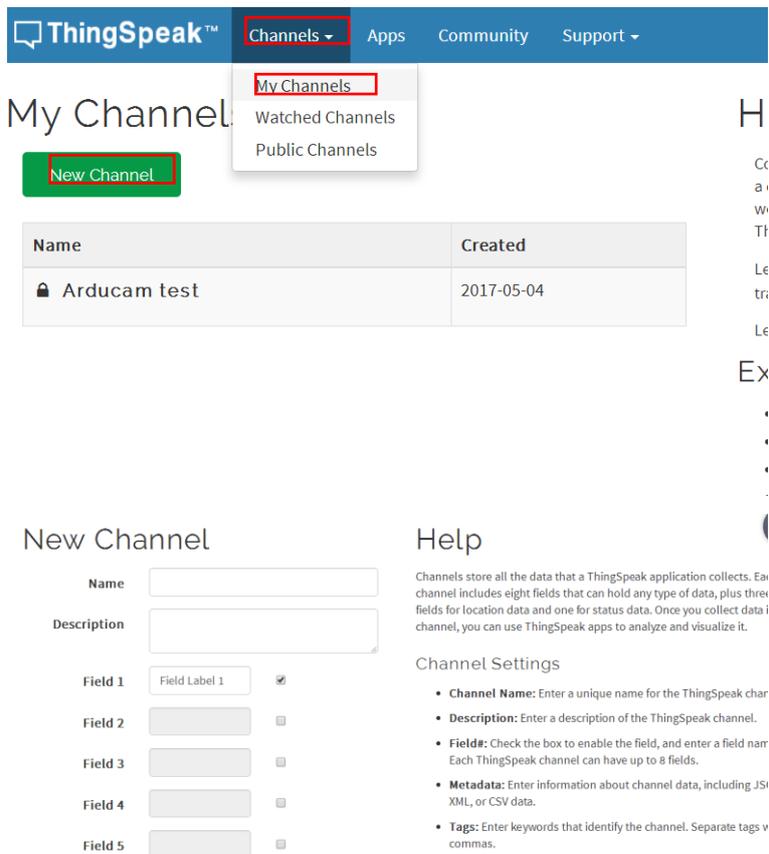
4.3 Data Logging at Thing Speak

To log data to thing speak, you'll have to create an account and additionally create 3 channels to log each of the parameters.

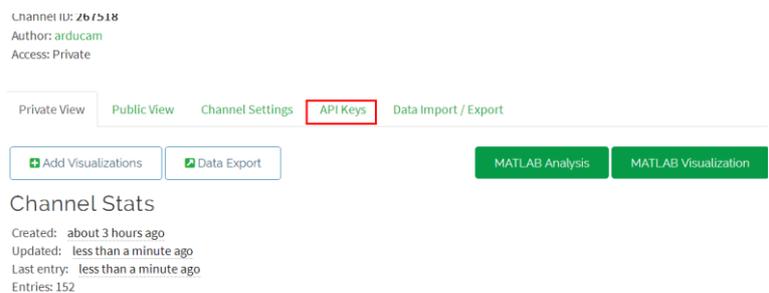
- 1) Go to https://thingspeak.com/users/sign_up and create the account.
- 2) After you completed that process, log in to your new account and go to My Channels



3) Click on the New Channel button, and fill out the form:



4) Navigate to the API Keys tab and note the two generated keys:



ThingSpeak™ Channels Apps Community Support How to Buy Account Sign

Private View Public View Channel Settings API Keys Data Import / Export

Write API Key

Key: GT8A0Q5UFOHFUHI

Generate New Write API Key

Read API Keys

Key: OGIS99QLGUTN8LDI

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key**.
- Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access.

5) Collecting and Displaying Local Data

