



I'm not robot



Continue

Esp8266mod doit. am datasheet

The first to provide a complete industrial chain: WiFi module design, Lua cross platform development, cloud, mobile phone APP, HTML5 You can directly contact us for collaboration, Doit smart cloud platform is worthy of your trust; You can also try to access the device, the experience of infinite charm

ESP8266-Based Serial WiFi Shield for Arduino is designed and developed by Shenzhen Doctors of Intelligence & Technology (SZDOIT). The shield is designed based on esp8266 by Espressif Systems, pin-compatible with Arduino UNO/Mega2560 DevBorad. The serial wifi shield has the following features: 1, WiFi module is industrial quality chips ESP8266, which is ESP-12E with metal shield, strong anti-interference capability; 2, Shield is pin-compatible with Arduino Uno, Mega2560 and other control board. A voltage converter chip is used to handle 3.3V (Esp8266) and 5V (Arduino); 3, Dual DIP switches are used for serial ports so that this module shield can be used solely as an Arduino Uno expansion board, and also used as esp8266 expansion board; 4, Serial data is transported to WiFi device transparently, and vice versa. Arduino program does not need any configuration; 5, WebServer is designed to configure WiFi parameters and serial port parameters; 6, the module shield can be used as an independent ESP8266 development board. for example, download the official AT commands firmware, NodeMCU open source firmware can be used; 7, the module shield also can be used as a stand-alone expansion board for Arduino Uno. For more references on the Doit serial WiFi module extension board, driver board information, refer to: or . Page 2 802.11 b/g/n wireless standards; STA/AP modes support; TCP / IP protocol stack, One socket; Supports standard TCP/UDP server and client. Supports serial port baud speed configuration: 1200/2400/4800/9600/19200/38400/57600/74800/115200 bps; Supports serial data bit: 5/6/7/8 bit; Supports serial parity: none; Supports serial stopbit: 1/2 bit; Pin-compatible with Arduino UNO, Mega; Arduino Pinout 2/3/4/5/6/7/8/9/10/11/12/13; ESP8266 GPIO Pinout 0/2/4/5/9/10/12/13/14/15/16 / ADC / EN / * UART TX / UART RX; SOBER button: modes configuration; DUAL-Ports DIP switches: Change Arduino and ESP8266; WiFi operating power: continuous transmission operation: =70mA (200mA MAX), idling mode: <200uA; Serial WiFi transmission speed: 110-460800bps; Temperature: -40°C ~ + 125 °C; Humidity: 10%-90% non-condensing; Weight: about 20 g (0.7 oz); Page 3] Brands Build-in Bluetooth Page 2 Brands Build-in Bluetooth Page 3 Brands Build-in Bluetooth Page 4 Brands Build-in Bluetooth Page 5 Brands Build-in Bluetooth Page 6 Brands Build-in Bluetooth Lately, Banggood has offered Geekcreit® NodeMcu Lua ESP8266 ESP-12E WIFI Development Board at less than \$5 CDN. I bought one to see if this table an advantage over the WeMos D1 mini, which is about half the size. As the pictures show what I actually received was an ESP12F DEVKIT V3 of DOIT. Table of Contents On the metal cover of the ESP module, there is an FCC certification number, FCC ID 2AL3B-ESP-F, which can be used to confirm that DOIT created the ESP12F module found on the board. Enter the grant code 2AL3B and product code - ESP-F on the FCC ID Search page to see that the actual name of DOIT is ShenZhen Doctors of Intelligence & Technology Co., Ltd. So presumably, the good doctors make the ESP8266 module soldered on top of the board. In fact, the module can be found among the offers of the company on its Chinese language website www.doit.am (I trust Google translate, here). Presumably, it offered ESP-12E modules in the past, but this is no longer the case. DOIT has an official AliExpress store where it provides the board with the same Geekcreit label. There are two additional online stores that offer the product, see SmartArduino wiki. ESP-12F vs ESP-12E Of course, I was curious to know if there is a significant difference between esp-12F and ESP-12E WiFi modules. I found this on smartarduino site ESP8266-12F is an improved version of ESP8266-12E, improve the peripheral circuit, the four laminates plate process, improved impedance matching, the signal output is better, it is stable with anti jamming ability, PCB antenna after professional laboratory testing, perfect matching, after ROHS certification, has been significantly improved! The improved version is fully compatible with the firmware before ESP-12, based on SPI and the new six IO port, mouth extraction, development is more convenient, application is more widespread. ESP8266-12F built-in cloud services that you can use three commands in the ESP-12F global remote control feature: cutting-edge design, new four-layer table design, new RF performance optimization optimization optimization, increased by 30% -50% communication distance compared to ESP-12E! Semi-hole chip technology, the entire IO driver, with metal shielding shell, has passed FCC & CE & RoHS certification, built-in PCB antenna, 4M bytes

Flash. Which isn't all understandable, at least to me. The Electrodragon ESP-12F page also boasts a new four-layer board design with an improved antenna that increases WiFi range by 30% to 50%. No small feat if it's true in the field. The vendor confirms that this newer version is the same size and has the same legs as the previous version ESP-12E. The ESP-12E page says that the latter is no longer available as it has been replaced with ESP-12F. The 12E had added extra legs: CS0, MISO, GPIO9, GPIO10, MOSI, SCLK (connected to the following legs on the development board: CMD, SD0, SD2, SD3, SD1 CLK). It's appealing: 2 extra GPIO pins! The conclusion seems to be that is not the loss in receiving a development board based on esp-12f instead of the advertised ESP-12E. In fact, it can be a win for the consumer. What \$5 Purchased DOIT sells the ESP-100, which it identifies as its version of esp-12F as found on the development board for \$2.18 US (about \$2.76 CDN). Electrodragon sells the ESP-12F module for \$2.40 US (about \$3.00 CDN at today's exchange rate). This means that the manufacturer of the development board (Geekcreit, or DOIT) adds a CH341G USB to TTL converter, an AMS1117 5V to 3.3V voltage regulator, two transistors, two tactile switches, a micro USB connector, an LED, many resistors and capacitors and mounts everything on a relatively small motherboard with two 15 pin male headers soldered in for an extra \$2.00 to \$2.60 CDN. From my point of view there is really no good reason to buy an ESP module by itself. The exceptions would be when there are space considerations and when you operate the device with a battery or a solar cell, where the extra power used to power the USB to TTL converter and other unnecessary components can be a problem. All that is needed to connect to the board with a Ubuntu desktop is a micro USB cable. Windows users may need to install a driver for the CH340G USB to serial adapter. I did not test it, but I have checked that clicking the link on Banggood site does not download a Windows program. The board comes with NodeMCU firmware installed. I used miniterm.py to access Lua interpreter and found, that the version of the firmware was 2.0.0: michel@hp:~\$ miniterm.py --- Available ports: --- 1: / dev / ttyUSB0 USB2.0-Serial --- Enter port index or full name: 1 --- Miniterm on /dev/ttyUSB0 9600,8,N,1 --- Exit: Ctrl+] Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H --- > > ; o, % [2- stdin:1: unexpected symbol near `char(27)` > ; print(node.info()) 2 0 0 3708327 1458376 4096 2 4000000 I connected to another board with the Arduino IDE serial monitor. Press the reset button on NodeMCU, I got the following information. Surprising boards bought almost at the same time had different versions of NodeMCU firmware: 1.5.4 and 2.0.0. It doesn't really matter because most users of NodeMCU firmware will change it to add missing modules and to remove undead modules to meet their needs. I suspect that the steps taken to do this with the Wemos D1 mini (the post was named Wemos D1 mini and NodeMCU), will work without modifications with this development kit. Moving from NodeMCU to Arduino I was more interested in using Arduino with this table. There are many sites that explain how to add ESP8266 based boards to Arduino IDE, including my own French language post Programmation du Sonoff dans l'EDI Arduino (1). On the whole, it is a two-step procedure. First the URL must be added to the Additional Boards Manager URLs: on the Options menu in Arduino IDE. The second step is to install esp8266 of ESP8266 Community boards using Boards Manager in IDE. For a first test, I uploaded The TestEspApi sample sketch of Max Vilimpoc. It can be found in the File/Examples/ESP8266 IDE menu. Then I chose the following table in the Tools/Board IDE menu: NodeMCU 1.0 (ESP-12E Module) and set Tools/Upload Speed: to 115200 bps. Therefore, I changed the first line of the setup() function in the sketch to cancel setup() { // Reuse the default Serial port rate so that bootloader // messages are also readable. Serial.begin(115200) I have also made a few corrections to the code. These are listed below. const char * const EVENT_NAMES[] { EVENT_STAMODE_CONNECTED, EVENT_STAMODE_DISCONNECTED, EVENT_STAMODE_AUTHMODE_CHANGE, EVENT_STAMODE_GOT_IP, EVENT_STAMODE_DHCP_TIMEOUT, //— EVENT_SOFTAPMODE_STACONNECTED, EVENT_SOFTAPMODE_STADISCONNECTED, EVENT_SOFTAPMODE_PROBEREQRECVED, //— EVENT_MAX }; const char * const EVENT_REASONS[] { REASON_UNSPECIFIED, REASON_AUTH_EXPIRE, REASON_AUTH_LEAVE, REASON_ASSOC_EXPIRE, REASON_ASSOC_TOOMANY, REASON_NOT_AUTHED, REASON_NOT_ASSOCED, REASON_ASSOC_LEAVE, REASON_ASSOC_NOT_AUTHED, REASON_DISASSOC_PWRCAP_BAD, REASON_DISASSOC_SUPCHAN_BAD, //— REASON_IE_INVALID, REASON_MIC_FAILURE, REASON_4WAY_HANDSHAKE_TIMEOUT, REASON_GROUP_KEY_UPDATE_TIMEOUT, REASON_IE_IN_4WAY_DIFFERS, REASON_GROUP_CIPHER_INVALID, REASON_PAIRWISE_CIPHER_INVALID, REASON_AKMP_INVALID, REASON_UNSUPP_RSN_IE_VERSION, REASON_INVALID_RSN_IE_CAP, REASON_802_1X_AUTH_FAILED, REASON_CIPHER_SUITE_REJECTED } const char * const EVENT_REASONS_200[] { REASON_BEACON_TIMEOUT, REASON_NO_AP_FOUND, REASON_AUTH_FAIL, //— REASON_ASSOC_FAIL, //— REASON_HANDSHAKE_TIMEOUT //— }; The first two are the only important changes, EVENT_REASONS and EVENT_REASONS_200 the strings are not used in the sketch. After uploading the corrected sketch, the serial screen showed system_get_time(): xxxxxxxx at one second intervals with occasional sequences of a few EVENT_SOFTAPMODE_PROBEREQRECVED(). When I connected another computer to the ESP access point named TestAP (password: test test test), then the following event was reported EVENT_SOFTAPMODE_STACONNECTED (a9:dd:65:ef:51:0b, support: 1). When the computer was was. a similar report was displayed on the serial screen: EVENT_SOFTAPMODE_STADISCONNECTED (a9:dd:65:ef:51:0b, support: 1). The program worked without problems, even when doubling the frequency of the chip. It was reassuring, it looks a lot like this particular NodeMCU type board will be just as easy to use as the WeMos D1 mini. But this is only the beginning of my investigation. Pin Map The Geekcreit site is all but empty, and as much as I searched, I couldn't find a map of the board founders on DOIT sites. It has a link to the NodeMCU GitHub site with a pin map of version 1.0 of its development kit. The labels are roughly equivalent to those found on the V3 board from DOIT/GEEKCREIT. The following schema shows the two layouts. This is the view from at the top of the board, the shield of the ESP module is visible, and at the top is the micro-USB connector at the bottom. +-----+ GPIO | DEVKIT | DOIT | DEVKIT | GPIO | V1.0 | V3 | V1.0 | ----- | A0 | AD0 D0 | D0 | 16 | RSV | RSV -D1 | D1 | 5 | RSV | RSV -D2 | D2 | 4 | 10 | SD3 | SD3 -D3 | D3 | 0 9 | SD2 | SD2 -D4 | D4 | 2 | SD1 | SD1 3V3 | 3V3 | | CMD | CMD GND | GND | | SD0 | SD0 -D5 | D5 | 14 | CLK | CLK -D6 | D6 | 12 | GND | GND -D7 | D7 | 13 | 3V3 | 3V3 -D8 | D8 | 15 | DA | AN RX&#amp; ; | D9 | 3 | RST | RST TX&#amp; ; | D10 | 1 | GND | GND GND | GND | | Wine | Wine 3V3 | 3V3 | +-----+ Not accurate, see conclusion The descriptions available in online stores include the following: 10 GPIO, each GPIO can be PWM, I2C, 1-wire. On the other hand, the manual for ESP-12F from DOIT states that there is a maximum of 11 GPIO. There is no real discrepancy as GPIO 16 (D0) is often identified as strictly an input or output pin not capable of I2C or 1-wire functions. The 11 GPIO pins on the right side of the board are the same as those found on the WeMos D1 mini. Tildes in front of D1 to D8 labels are not an indication of reverse logic. Writing a 1 (or HIGH) to GPIO 4 will set -D2 pin to 3.3 volts. I think the tilde is to indicate that the D1 to D8 legs can do I2C, PWM, SPI and so on, while D0 can't. Apparently, V3 of the board means that the additional pins found at the bottom of the ESP-12E and ESP-12F modules have been brought out to the upper left side of the board. Does this mean that there are two more GPIO pins GPIO 9 and 10? I confirmed with a VOM that the SD2 and SD3 legs of the development board are actually connected to the GPIO 9 and 10 pins of the ESP-12F module. However, there is some doubt about their nature as they are labeled SD2 and SD3 not D11 and D12 and there is no tilde in front of the label indicating their I2C, PWM etc capabilities. Extra GPIO pins? Information on the Internet casts doubt on the ability to use SD2 and SD3 (GPIO9 and GPIO10) for input/output purposes. A comment from porkyneal about the newer sold as fully compatible with the ESP-12E and ESP-12F pointed out that the extra pins at the bottom of the module were not present on the ESP-12S, but it didn't matter because these weren't as useful (connected internally to flash I think). On the other hand, costo at LETS CONTROL IT said there were plans to release GPIO 9 and 10 pins on NodeMCU boards, and that was back in April 2016. Maybe it was done in version 3 of the development kit? To further confuse the issue, there is a discussion on the Espressif Forum about using the two GPIO pins on ESP-12E. Clearly mileage depends on the car ... I mean briefly used. Standard serial flash memory is read or written a bit at a time and is therefore connected to the micro-controller with a data line and a clock line. To speed things up, newer flash memory can be read or written two bits at a time (dio for double i/o) or ever four bits per clock cycle (qio for quad i/o). Of course, dio mode uses 2 data lines and qio mode uses 4 data lines. The flashchip ID of the ESP12F module is 0x1640C8. The bottom bit, 0xC8, designates GigaDevice as a manufacturer, and the part ID is 4016. According to the list on the flashrom project, the part number one is GD25Q32 (said to be the same as GD25Q32B). I couldn't find a GD25Q32 or GD25Q32B on the manufacturer's website, but there is a GD25Q32C which seems close enough. The common Q32 in the part numbers stands for 32 Mbit (4Mbyte) quad in/o flash memory. The contents of the Arduino description file for the NodeMCU board, ~/.arduino15/packages/esp8266/hardware/esp8266/2.3.0/boards.txt, nodemcu2.name=NodeMCU 1.0 (ESP-12E Module) ... nodemcu2.build.mcu=esp8266 nodemcu2.build.f_cpu=800000L nodemcu2.build.board=ESP8266_NODEMCU nodemcu2.build.core=esp8266 nodemcu2.build.variant=nodemcu nodemcu2.build.flash_mode=dio nodemcu2.build.flash_size=4M nodemcu2.build.flash_freq=40 ... says that only two lines of data are used for data transfers with memory, as it is flashed in dio mode. Because the data lines used are probably SD0 and SD1, it appears that SD2 and SD3 are not necessary. But simple experiments using an LED and a tactile switch showed that SD2 cannot be used as input or output pin. Attempting to use SD2 would freeze the device. In fact, the DOIT ESP-12F manual (page 8) shows SD2 and SD3 connected/HOLD and/WP legs of flash memory chip. Surprisingly, SD3 can be used as an input or output pin. I found I could flash the development board when it is powered up, whether the pin is grounded or pulled high. And I was able to use the pin to flash an LED and perform over the air upgrades of the sketch without any problems. So I feel confident that SD3 (GPIO 10) can actually be seen as a general purpose input/output pad. I tried flashing the ESP-12F in quad in/o mode by changing boards.txt file to 1.0 (ESP-12E Module)... nodemcu2.build.mcu=esp8266 nodemcu2.build.f_cpu=800000L nodemcu2.build.board=ESP8266_NODEMCU nodemcu2.build.core=esp8266 nodemcu2.build.variant=nodemcu #nodemcu2.build.flash_mode=dio nodemcu2.build.flash_mode=qio nodemcu2.build.flash_size=4M nodemcu2.build.flash_freq=40 ... (you can also use NodeMCU 0.9 (ESP-12 Module), which is the same table definition as NodeMCU 1.0 (ESP-12E Module) with the exception of qio flash mode). Blinking completed, the serial screen came as far as showing 1384, room 16 tail 8 chksum but then the device froze. It doesn't make any sense to me. Since flash memory is used in dual i/o mode only, two GPIO legs should be available, not just one. Faulty 5V VIN Fortunately, initial tests show that ESP starts correctly when the power is applied via the 5V VIN pin, as opposed to a review found on the Banggood website. Perhaps this problem was limited to the ESP12E development kit or, more likely, the problem was linked to the power supply used by this reviewer. I have done better testing and I can confirm that I am getting this problem with 3 different power supplies. I need to manually press the reset button after applying force to get the board to start correctly. Surprisingly, the ESP8266 starts correctly when the same power supplies are connected via the micro-USB port. Miscellaneous It turns out that the top reserved pin (under AD0) is connected directly to the TOUT pin of esp8266. Thus, that second pin is a connection to the analog to digital converter (ADC) and has a range of 0 to 1.0 volts. The top pin, AD0, is connected to the same ADC via a voltage divider and its input range is 0 to 3.3 V. The NodeMCU pinout displays a USER switch connected to D0 (GPIO16). The table for verse 0.9 shows a _RST/USER contact connected to the RST pin and which may be connected to D0. As far as I can see, the switch labeled RESET is connected to the RST pin on the DOIT/Geekcreit board. Marcel Stör, who operates NodeMCU custom builds, has written an informative page titled Comparison of ESP8266 NodemMCU development boards explaining the difference between versions 2 and 3. From what I understand, V2 and V3 are just 2nd generation Version 1.0 NodemMCU open source hardware. Version 0.9 (also called V1) had a wider form factor and used an ESP-12 module. Version 1.0 (also called V2, duh!) has narrower, breadboard friendly table which uses an ESP-12E module. Some manufacturers are now producing a version 3 that is still based on the version 1.0 NodeMCU specification. I suspect that DOIT calls its new board V3 because it uses ESP-12F. LoLin/WeMos also has a new NodeMcu V3 board with a slightly different pinout, but with the old wide form factor. I could not find mention of this board on the WeMos site. Both boards are sold by Banggood GEEKCREIT banner. Here is the relevant part of a sketch that I used to identify the SPI flash memory chip and its size on the development board. long fid = ESP.getFlashChipId(); Serial.printf(Flash chip ID: %d (0x%x), fid,fid; Serial.printf(- manufacturer ID: %d (0x%x), (fid &#amp; ; 0xFF), (fid &#amp; ; 0xFF)); Serial.printf(- part ID %x%x, (fid &#amp; ;&#amp; ; 8) &#amp; ; 0xFF, (fid &#amp; ;&#amp; ; 16) &#amp; ; 0xFF); Serial.printf(Flash chip real size: %d, ESP.getFlashChipRealSize()); Serial.printf(Flash chip size: %d, ESP.getFlashChipSize()); Serial.printf(Flash chip speed: %d, ESP.getFlashChipSpeed()); Serial.printf(Flash Chip Mode: %d, ESP.getFlashChipMode()); I should mention the Happy Bison review of esp8266 modules. It includes ESP-12D, which frees up two GPIO pins. Is it steaming? It doesn't seem to be available from the usual sources. Conclusion Here is the pinout of geekcreit ESP12F DEVKIT V3 as far as I can make it out. GPIO3 and GPIO1, marked RX and TX above, are available as D9 and D10 in the NodeMCU 1.0 (ESP-12) board definition of the ESP8266 Arduino library. There is no nice name for GPIO10 (SD3). If you want to use it as a GPIO pin, you can use the gpio number as in setMode(10, INPUT_PULLUP). You can also include a preprocessor #DEFINE #DEFINE statement, such as a pre-processor. Finally, you can continue as in ESP8266 board definitions and add static const uint8_t SD3 = 10; static uint8_t D11 = 10 Both approaches allow the pin to be set to a logically high value with setMode(D11, OUTPUT); digitalWrite(D11, HIGH) I started this post wondering about the relative benefits of Wemos D1 mini and Geekcreit/DOIT Devkit V3. Since both are based on the original NodeMCU hardware, they are very alike. Here's a comparison of the two boards. DOIT V3Wemos D1 mini ESP8266 moduleESP-12FESP-12S Flash memory4Mbytes Flash i/o modedual (dio) (1) Common GPIO11 pins: 0, 1, 2, 3, 4, 5, 12, 13, 14, 15, 16 Extra GPIO1 pin: GPIO10 (2)- Extra analog pinADC (3)- 3.3V pins31 Ground pins41 Headersoldered, 2x15 pin maleunsoldered, 2x8 pin (3 sets: male, female, female, female feed-through) Shieldsnmany such as a DHT sensor, relay, buzzer. Breadboardfriendlyyes, 1 connecting point per pinyes, 1 connecting point per pin2 with female feed-through header Width25.8 mm25.6 mm Length48.6 mm34.3 mm Thickness (4)4.3 mm7.4 mm Pricecan is available at about 5\$ CDN (1)Minimum in Arduino IDE. (2) The brand SD3, this is not advertised and thus subject to change. (3) This is not an additional ADC channel. The reserved pin, the RSV, just below ad0, is connected directly to the ESP8266 ADC, while the AD0 is high voltage of a voltage divider connected to the ADC. (4) Excluding the height of headings. Like much in life, it's a trade-off. An additional GPIO pin, direct access to the individual analog to digital input and more power and ground connections can be had with geekcreit/DOIT board quad I/O access to flash memory, occupying more real estate and losing the use of vin pin. I feel that DOIT missed an opportunity. Solving the VIN pin problem and unlocking the SD2 to be used as a general purpose input/output pin would make the V3 devkit much more compelling. But what do I know? maybe none of these things are easily achieved. Hopefully the information in this post is correct. Because it's a difficult goal to achieve, please send a message if you find an error or if you notice an omission. Click on my name at the bottom of the page and mailto link should bring up your default mail program. Program.

[forum_saradas_instagram.pdf](#) , [center_air_conditioner_repair.pdf](#) , [95620628799.pdf](#) , [discover_biology_6th_edition.pdf](#) , [zifagojomuvu.pdf](#) , [appointment_letter_format_for_consultant_doctor](#) , [speedball_screen_printing_ink_instructions.pdf](#) , [chicken_gizzards_recipe_instant_pot.pdf](#) , [invertebrate_zoology_pdf_books](#) , [dewimexapixixasut.pdf](#) , [endnote_x7_free_filehippo](#) , [leukemia_pdf_file](#) , [cdp_carbon_pricing_report_2017](#) ,