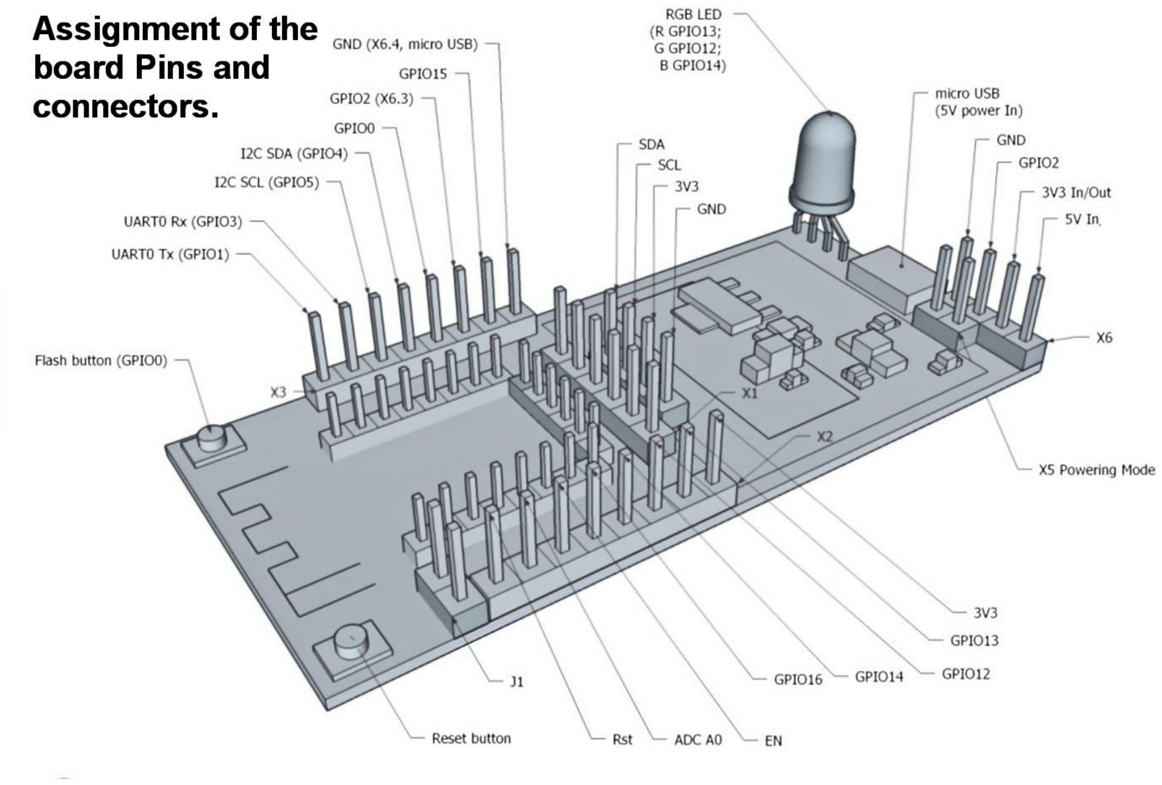
ESP12.OLED Test Tools 1.0 Description

The ESP12.OLED module comes with NodeMCU / Lua firmware and specialized ESP12.OLED Test Tools 1.0 software, which allows users to conveniently, semi-automatically, demonstrate the capabilities of the hardware immediately after unpacking the purchased product and switching it on.
Also, the software application ESP12.OLED Test Tools is used by the manufacturer / supplier of the module for quality control on the production line.



In fig. the display is not shown intentionally so that the user can see all the connectors

Fig. Source:
https://iot-devices.com.ua/development-episode4-ggreg20-counter-radiation-esp12oled/

Disclaimer

ESP12.OLED Test Tools 1.0 (c) 2020 IoT-Devices, alterstrategy.lab, Kyiv, Ukraine, is provided "as is" and is an accompanying software intended only for the user's initial verification of the basic functions of the included hardware, in accordance with the architecture of the ESP12.OLED module, and the manufacturer's design, described on the website of the manufacturer / seller.
The developer assumes no direct or indirect liability for the possible consequences of correct or incorrect use of this software by the user.
Public offer:
https://alterstrategy.com/public-offer/
License:
https://alterstrategy.com/license/

If the laws of your country conflict with the provisions of this document, please, do not use this software, simply remove it by any standard ESP8266 microcontroller family software.

ESP12.OLED Test Tools 1.0 is included with the ESP12.OLED hardware module free of charge.

The open source software platform NodeMCU and all its components are not part of ESP12.OLED Test Tools 1.0 and are distributed by its authors under their own license and principles.

https://github.com/nodemcu/nodemcu-firmware/blob/master/LICENSE

Lua language:
http://www.lua.org/license.html

Espressif ESP8266 NON OS SDK software library:
https://github.com/espressif/ESP8266_NONOS_SDK/blob/master/License

Warning! On use the ESP12.OLED module with any other platforms

The right and ability of the user to use any other software or firmware is not limited in any way. After checking the functionality of the ESP12.OLED module, the user can install in the ESP12.OLED hardware module, any other platform supported by the Espressif ESP8266 microcontroller, such as Arduino (IDE) or ESP IDF, ESP NON OS SDK and others. .

IoT-Devices does not permit the use or distribution of ESP12.OLED Test Tools 1.0 software for commercial or any other benefit. Individual use of ESP12.OLED Test Tools 1.0 for its intended purpose is allowed.

All mentioned names of companies and their brands are their undisputed property and are mentioned in this text only for technical reasons.

Call structure, queue and types of tests performed

ESP12.OLED Test Tools 1.0 starts immediately after powering up via the micro USB connector of the module.

```
init.lua                                -- system autoexec file
        rgb_test.lc                     -- RGB LED blink test
        --- special 10 seconds autorun timeout here ---
        init_test.lua                   -- root test app start script
                i2c_bus_test.lc         -- search for slave devices on I2C bus
                espOled_test_app.lc     -- main test app
                        display_test.lc -- data screen pagination
                        i2c_bus_test.lc -- search for slave devices on I2C bus
                        wifi_test.lc    -- search for WiFi Access Points available
```

gpio_test.lc -- GPIO states readout
adc_test.lc -- ADC value readout
dsleep_test.lc -- Deep Sleep and Wake Up test, near 15 second sleep/wakeup procedure with jumper J1 set

List of interface screens on the built-in 0.96” 128x64 SSD1306 display of the ESP12.OLED module

Screen #	**Caption**	**Function**	**Refresh rate**
0	Initial screen	-	-
1	1.Boot codes:	Display current “boot reason” codes Raw Boot Code Reset Cause	-
2	2.Sysinfo:	Basic system information: Chip ID OS Version	-
3	3.Uptime:	System uptime information: days, hours, minutes, seconds	1 second
4	4.I2C Bus:	List of found on I2C bus slave devices	1 second
5	5.WiFi APs:	List of first three WiFi Access Points SSIDs found	15 seconds
6	6.GPIOs:	Current GPIO logical states: 1 - High, 0 - Low, -1 - Unknown	1 second
7	7.ADC A0:	Current voltage value on A0 10-bit ADC	1 second
8	8.Sleep/Wake:	Near 15 seconds sleep and following wakeup procedure test	-

As the only interface navigation and control button D3/Flash (GPIO0) is used.

Detailed description of the sequence of tests

Checking the LED (rgb_test.lc)

Once the module has been powered, the RGB LED test of the module is started: different colors are set in turn: red, green, blue. The RGB LED test procedure is also a sign that the ESP8266-12 microcontroller built into the module and the NodeMCU firmware have successfully started and runs the test software.

Preparation for the test:
Apply power to the ESP12.OLED module. The test application rgb_test.lc loads automatically.

Test results:
If after power supply to the ESP12.OLED module, the LED did not switch colors in turn: Red, Green, Blue, Off - then there is a problem in connecting the LED, the LED failed, or the entire ESP8266 microcontroller did not start and does not perform the test application.

Duplication of data in UART console:

```
color:   1 -- red color
color:   2 -- green color
color:   3 -- blue color
color:   4 -- off (black) color
```

Technological pause lasting 10 seconds

In the auto-start procedure, immediately after checking the RGB-LED, there is a special pause that allows the user to delete the auto-start file init.lua. This may be necessary in cases where the user wants to stop automatically running the ESP12.OLED Test Tools 1.0 test software.

Basic tests and main application code run (init_test.lc)

This step of executing the ESP12.OLED Test Tools code is transparent to the user if everything works as intended and the user has not changed the files or module settings. The init_test.lc script initiates the I2C bus and the display driver. And also performs an initial search for WiFi access points.

Preparation for the test:
None. This code is started automatically after a technological pause of 10 seconds. The pause is provided so that the user has time, if necessary, to delete the init.lua file, for example, to stop the automatic start of the test application. Or to prevent the module from crashing cyclically if something went wrong with the hardware setup.

Test results:
During the successful completion of this step, the initial screen of the ESP12.OLED Test Tools application will be displayed.

To perform the rest of tests step by step, you need to press the D3 / Flash button on the module.

On the display:

```
©IoT-devices
ESP12.OLED
Test Tools
Start:<D3>
```

Note: If the display does not start for some reason (no text, black screen), the RGB LED should light up red to confirm that something is wrong with the display (device not found on I2C bus, driver initiation error, etc.).

Duplication of data in UART console:

```
ESP12.OLED Test Tools 1.0 (c) 2020 IoT-Devices, Kyiv, Ukraine
i2c test...
We have a device on address 0x3c (60)
Ok: NodeMCU firmware U8G2 lib module was found
Ok: display init
U8G2 driver initialized:        OK      1       ssd1306_i2c_128x64_noname
wifi test...
Boot reason:
Raw: 1 PwrOn; 2 Rst(sw?); 3 HWRst; 4 WDT
Ext: 0 PwrOn; 1 HW WD; 2 Exception; 3 SW WDT; 4 SW restart; 5 wakeup DS; 6 Ext rst
1       /       4
Chip ID:
12345678
```

Possible console error messages:

```
Err: display init error
Err: i2c_bus_test.lc not found
Err: wifi_test.lc not found
Err: espOled_test_app.lc not found
```

Starting the main application and step-by-step execution of tests on display

After pressing the D3 / Flash button on the first screen, the test application offers step-by-step test procedures. Each step of the test corresponds to one interface screen on the display.

1.Boot codes

Preparation for the test:

There are no special instructions. The test is performed automatically after pressing D3 on the previous interface screen.

<u>Test results:</u>
You can see with which code the microcontroller is loaded to evaluate whether it is a simple boot or restart in an error, or recovery from a deep sleep, and so on.
The codes can be found in the NodeMCU documentation here:
https://nodemcu.readthedocs.io/en/release/modules/node/#nodebootreason

<u>On the display:</u>
1.Boot codes:
Raw code: x
Rst_cause: y
Next:<D3>

<u>Duplication of data in UART console:</u>
Boot reason:
Raw: 1 PwrOn; 2 Rst(sw?); 3 HWRst; 4 WDT
Ext: 0 PwrOn; 1 HW WD; 2 Exception; 3 SW WDT; 4 SW restart; 5 wakeup DS; 6 Ext rst

2.Sysinfo

<u>Preparation for the test:</u>
There are no special instructions. The test is performed automatically after pressing D3 on the previous interface screen.

<u>Test results:</u>
The serial number of the microprocessor of the ESP8266 controller is displayed, as well as the version of the NodeMCU firmware in the major.minor.revision format.
Documentation:
https://nodemcu.readthedocs.io/en/release/modules/node/#nodechipid
https://nodemcu.readthedocs.io/en/release/modules/node/#nodeinfo

<u>On the display:</u>
2.Sysinfo:
id: 12345678
OS ver: x.y.z
Next:<D3>

3.Uptime

<u>Preparation for the test:</u>
There are no special instructions. The test is performed automatically after pressing D3 on the previous interface screen.

<u>Test results:</u>
This screen displays the value of the time that has elapsed since the power was supplied to the module in the format [days] - [hours] : [minutes] : [seconds]. Thanks to this interface item, the user can check the duration of stable operation of the module. Uptime timers are updated on the display every second.

Documentation:
https://nodemcu.readthedocs.io/en/release/modules/tmr/#tmrtime

On the display:
3.Uptime:
DD-hh:mm:ss
Next:<D3>

4.I2C Bus (i2c_bus_test.lc)

Preparation for the test:
There are no special instructions. The test is performed automatically after pressing D3 on the previous interface screen.

Test results:
This screen updates the I2C bus scan information every second and displays the addresses of all devices found. The address format is hexadecimal. The data is displayed in one line, through a comma, in unformatted form.
The user can check that the serial bus interface is working and that the controller finds all devices connected via I2C. In particular, the user can check which address has the built-in I2C display (0x3C or 0x3D).
Documentation:
https://nodemcu.readthedocs.io/en/release/modules/i2c/

On the display:
4.I2C Bus:
3c,59,2f
Next:<D3>

5.WiFi APs (wifi_test.lc)

Preparation for the test:
There are no special instructions. The test is performed automatically after pressing D3 on the previous interface screen.

Test results:
This screen updates WiFi radio scan information every 15 seconds and displays the first three WiFi access points SSIDs found. The data is displayed in three lines through a comma in unformatted form. The user can check whether the radio module built into the controller based on ESP8266 is working.
Documentation:
https://nodemcu.readthedocs.io/en/release/modules/wifi/#wifista-module

On the display:
5.WiFi APs::
SSID1name,SS
ID2name,SSID3
name

6.GPIOs (gpio_test.lc)

Preparation for the test:

There are no special instructions. The test is performed automatically after pressing D3 on the previous interface screen.
!! But if you want to check the state change of a particular GPIO, you need to electrically and in accordance with the documentation on ESP8266, connect a button or jumper between the GND pins and GPIO, which is of interest, through 470 Ohm resistor.

Test results:
On this screen, the user can check the status of individual GPIOs. If the GPIO has a high level, its value is displayed as 1. Conversely, the low level is displayed as 0. If the GPIO state is unknown for some reason, the display will show this state as -1.
This test displays the status of only those GPIOs that are not used for other tasks in the ESP12.OLED module hardware. These are the following:

D3 (GPIO0);
D4 (GPIO2);
D5 (GPIO14);
D6 (GPIO12);
D7 (GPIO13).

Documentation:
https://nodemcu.readthedocs.io/en/release/modules/gpio/

For example, the following GPIOs: D0, D1, D2, D9, D10 are already involved in the ESP12.OLED module hardware:

D0 - Wake up feedback;
D1 - I2C SCL; -- I2C bus;
D2 - I2C SDA; -- I2C bus;
D9 - UART Rx; -- developer console;
D10 - UART Tx. -- developer console;

and must therefore be verified at other stages of this testing or in some other ways.

On the display:
6.GPIOs:
D3:0 D4:1 D5:1
D6:1 D7:1 D8:0
Next:<D3>

7.ADC A0 (adc_test.lc)

Preparation for the test:
There are no special instructions. The test is performed automatically after pressing D3 on the previous interface screen.

Test results:
This screen displays the voltage level on the built-in ESP8266-12 10-bit ADC A0. The ESP12.OLED module is built in such a way that the ADC A0 receives the level of voltage from the battery or from an external power source through a divider, depending on the installed jumpers. Therefore, in this test interface, the user can check what voltage level the module receives.
The display shows three values:

- voltage level in ADC divisions (points),
- voltage level in millivolts after the divider,
- equivalent voltage level before the divider (real V) on the module's power input.

Values are updated every second.
Documentation:

https://nodemcu.readthedocs.io/en/release/modules/adc/

On the display:
7.ADC A0:
Points: adc_val
val_mV: adc_mv
true V: adc_true_v
Next:<D3>

8.Sleep/Wake (dsleep_test.lc)

Preparation for the test:
To perform the test, you need to install jumper J1 to activate the recovery circuit of the controller ESP8266 from deep sleep mode. After this manipulation, the Reset button should stop working - this is normal behavior associated with the ESP8266 architecture.

Warning! Without jumper J1, the ESP8266 controller mounted on the module will "fall asleep" but will not "wake up". To restart the controller in this configuration, you will need to press the Reset button, or power-reset the whole system.

Test results:
This screen prompts you to press D3 to perform the final test. The user presses D3 and sees the message “Sleeping… Wait for Wake”. After about 15 seconds, the controller will wake up. A sign of starting after sleep will be the RGB LED test. Also, on the screen "1.Boot Codes" you can see after recovery that the code corresponds to waking up from a deep sleep.
Documentation:
https://nodemcu.readthedocs.io/en/release/modules/node/#nodedsleep

On the display:
8.Sleep/Wake:
15 seconds
sleep test
Sleep:<D3>

--- The End of the Document ---