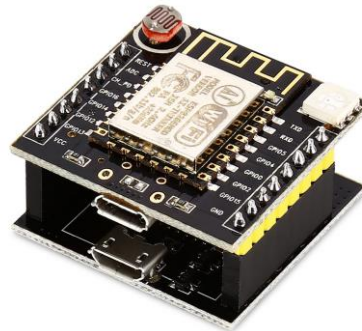




# Witty Cloud / GizWits - ESP8266 Development Board

## 1 Introduction

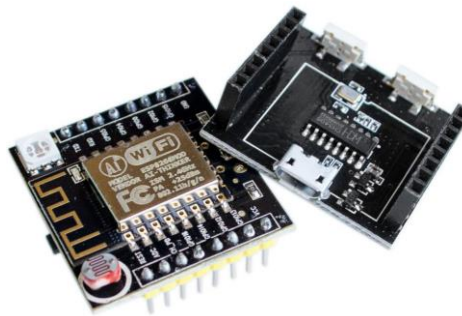
The *Witty Cloud* (also named *GizWits*) development board



Witty cloud development board

is interesting in the fact that it consists by design of two separate modules, with each module having a USB connector. Once programmed, only the upper part may be used fully functionally as programmed by you.

If you tear those modules apart



Witty Cloud modules torn apart

you end up with

- the *upper module*



Upper Witty Cloud module

holding on the *upper side*

- a photo resistor
- a WIFI antenna
- a multi-color LED
- a blue LED (between the multi-color LED and the corner of the ESP8266 module)
- the *ESP-12 module*

holding on the *lower side*

- a button
  - a voltage regulator
  - a micro USB port - with its only function to accept a power source
  - and the pins to plug into a breadboard
- The *lower module* with components on the upper side only



Lower *Witty Cloud* module

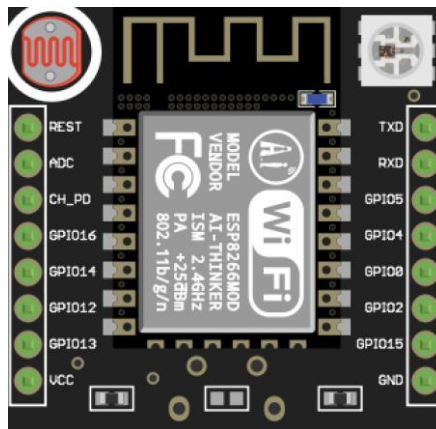
holding

- two buttons - the left one to set the system into flash mode, the right one to reset the system (Obviously both modules must then be assembled).
- the CH340G UART chip
- a micro USB port accepting a power source as well as data exchange with another device
- two female header connectors to connect with the second Witty Cloud module

## 2 The pinout

The pinout description is important - we want to know what pins we must consider to build our prototypes. Normally it is the Board developer/designer that will publish the corresponding

documentation, but in this case help comes from Mike Causer. He published a *Fritzing Part* describing the Witty Cloud board on *GitHub*.



Witty Cloud Fritzing Part  
by Mike Causer

### 3 About the factory installed firmware

The *firmware* is the operating system that is closely tied to hardware it has been developed for.

The board has been developed by *AI-Thinker*. The technical specifications can be found [here](#). The page is currently in Chinese language only - but once 'auto translated' you will get a comprehensive English translation. Under frequently asked questions you'll find the mention that Shun technology is protecting their intellectual property rights, that the SDK is encrypted and that the device is *factory activated*. Currently there is no solution to reactivate your device again once overwritten with a different firmware.

As the factory installed firmware is currently poorly understood or documented, I propose to install a different firmware to start playing around with our Witty Cloud board.

### 4 Firmware alternatives

Here the choices you have:

- *GizWits* - The original firmware provider of the Witty Cloud board. They are only referenced so you can follow up what's new. As of the writing of this paper, it is difficult to find proper documentation. Further reading on [GizWits's forum](#) (in Chinese) and [GizWits's Club](#).

*Espressif* (The creators of the ESP8266 microchip) is also referring to the firmware as *special firmware release (for China users)*.

- *ESP8266Basic* - Basic is a language beloved by millions of people. It's what many of us started out with and what the old timers used on their Commodore 64s. It's how Microsoft got started and one of the reasons for the explosive growth of computers in the 80s. Basic is a simple but powerful language that lets you do amazing things without needing a degree in computer science.



- *ESPEasy* is a firmware that is *IoT* (Internet Of Things) centric - controlling sensors, processing their data, publishing, accepting control commands via a simple web frontend - all this right out of the box. Further reading about supported sensors [here](#). The latest firmware is currently Firmware image R78 and the download link is [here](#).
- *MicroPython* is a lean and fast implementation of the Python 3 programming language that is optimised to run on a microcontroller. The *MicroPython* board is a small electronic circuit board that runs *MicroPython* on the bare metal, and gives you a low-level Python operating system that can be used to control all kinds of electronic projects. Documentation can be found [here](#). Further reading about MicroPython on this [Wiki](#). The forum can be found [here](#). Ready made firmware files can be downloaded [here](#). Firmware still in early stages!
- *NodeMCU* is an open source IoT platform using the *LUA* scripting language. The latest firmware can be found [here](#). Some code samples can be found [here](#).

There are other customized firmwares available, but without a readymade solution or a complete documentation about its possibilities, this is hardly an option for you.

Some reading of the sources provided is recommended before picking the firmware that best fits your project.

Just to mention it, firmware can be *flashed* more than once - so switching after some real life testing to an alternate firmware is no problem.