# esp8266hints

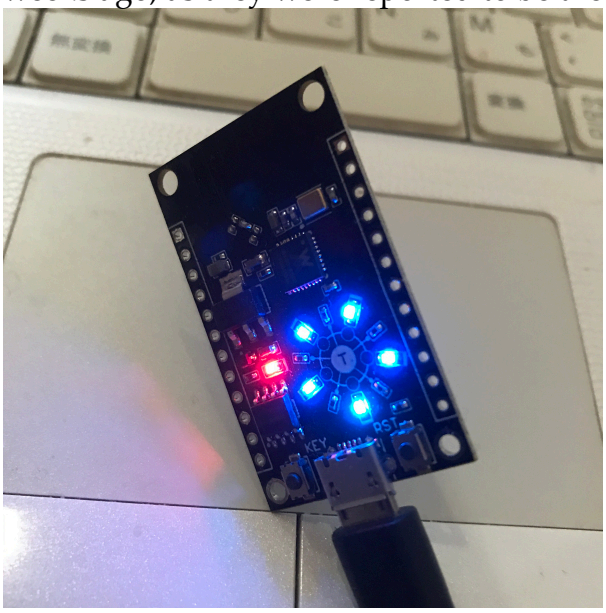
Just trying to untangle the threads

# W600

## First steps with the W600 (on Linux) (https://esp8266hints.wordpress.com/2019/08/31/first-steps-with-the-w600-on-linux/)

August 31, 2019September 5, 2019 PuceBaboon Hardware, How-To, MicroPython, Software, W600 Getting Started, Minicom, miniterm, Python, screen, Tps & Tricks

I ordered a couple of these W600 dev boards (https://www.aliexpress.com/item/32978979389.html) a few weeks ago, as they were reported to be the easiest ones to use when introducing yourself to the



(https://esp8266hints.wordpress.com/wp-content/uploads/2019/08/30-e1567239875594.jpg)W600 chip (these "ThingsTurn" boards don't have any second processor to grapple with, but do have a UART chip and USB connector for easy power/comms).

It turned out that while the power connection was easy, the comms were anything but and I wasted far too much time messing around trying to get it to accept an upload or even just a terminal connection. So today's post mainly consists of a series of short tips on how to get started with the W600 "ThingsTurn" dev board without pulling out all of your hair. Tips 1 through 4 deal with getting MicroPython onto your W600, while those from #5 onwards are examples of what you can do with the board from the MicroPython REPL (command line).

For my purposes, I just wanted to upload MicroPython as a quick, functional test, without having to download and install a toolchain and/or SDK, or really do anything at all that took more than a couple of lines of typing or 5 minutes of my time. Two days later, I was -still- getting nowhere very fast; my blood pressure was rising and I was itching to use my magic, solves-everything tool …the 1.5kg lump hammer.

So, tip #1 is — Don't believe everything you read (including this!). The W600 firmware and applications are still under heavy development. Most (but not all, see tip #2) of that work is taking place in China. The translation and maintenance of the manuals seems to be quite well down the TODO stack (something that most of us are guilty of). Specifically, don't bother trying to use their "download.py" (if you can find it) for anything, most especially not for uploading to ThingsTurn boards.

Tip #2 — Volodymyr Shymanskyy (https://github.com/vshymanskyy) has recently posted a great tool for uploading to the W600. It knows about the two available bootloader methods and will try to do the right thing (https://github.com/vshymanskyy/w600tool), depending upon which type of file you're attempting to upload. Don't bother with anything else; go straight to his repository and start using it now.

Tip #3 — Don't assume that your favourite terminal program(s) will work with the ThingsTurn boards, even if it has worked with everything else you've ever tried it with. It turns out the the the ThingsTurn boards (note that I'm being very specific here, as I don't believe that any of the other W600 modules suffer from this issue) have a requirement for the RTS signal to be held at "0" when connecting, otherwise the W600 will be forced into a permanent reset condition. I was unable to get either "screen" or "minicom" (my usual goto choices for comms terminals) to play with the ThingsTurn boards.

By this time I knew that the boards worked, as they communicated with "w600tool" easily and loaded MicroPython (to displace the simple, one-second flash on/off program which comes as default on the boards). Connecting with either screen or minicom would very occasionally produce a random, brief flash from one of the blue LEDs, but there was never any further indication of life and definitely nothing coming from the boards. A major step forward was finding that sending a "Hang-Up" request (CTRL-A H) from Minicom produced the MicroPython header and a REPL prompt from the board, before it lapsed back into its previous, comatose state. This confirmed that the upload had indeed worked and that, given the right conditions, MicroPython would run. At this point I reverted back to my 1980's RS232 self and started writing a simple C program to toggle the control lines… before remembering that this was the glorious future and somebody else on the 'net had undoubtedly already done it (and undoubtedly made a far better job of it than I ever did). So I knuckled myself on the side of the head a couple of times and turned back to Volodymyr Shymanskyy, as he'd already helped me out once and seemed to have already cracked this particular problem. That particular rabbit-hole led me, in very short order, to Python.

Tip #4 — Use the force, Luke! Or at least, don't be as stupid as me and refuse to see what's right under your nose. You wanted to get running with something quickly:- MicroPython. MicroPython =~ Python. W600tool =~ Python. There's a bit of a pattern emerging here. Yup, those clever Python dev guys have already written the solution to our problem and it is called "miniterm". A quick check of the on-line manual pages showed that it was quite capable of toggling selected hardware control lines and holding them in the selected state, as well as all of the normal port and baud selection stuff. So here's the command line to get you connected, up and running using miniterm:-

```
python3 -m serial.tools.miniterm --rts 0 /dev/ttyUSB0 115200
```

[…and just a reminder here that this is from Linux, you'll likely have to change at least the device name if you're using a different OS].

Once you've typed that in you'll see something like:-

— forcing RTS inactive
— Miniterm on /dev/ttyUSB0 115200,8,N,1 —
— Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H —

…followed by an ASCII-art representation of the Winner-Micro (stacked "W" and "M") logo (which really confuses WordPress formatting, so is not included here), followed by:-

WinnerMicro W600
MicroPython v1.10-279-g3c60627-dirty on 2019-05-24; WinnerMicro module with W600
Type "help()" for more information.
>>>

Yay! We're in business!

---

Tip #5 — Import the machine-specific hardware module. Configure a W600 pin (PB14 in this case) as an output (with no pull-up or pull-down) and assign name "led14" to it. Use the assigned name to toggle the value of the W600 pin (note that the blue LEDs on the ThingsTurn boards have their anodes connected to VCC via a 1k resistor and the cathodes are connected to the GPIO pins, so the output value of "1" turns the LED off and "0" turns it on:-

```
>>> import machine
>>> led14=machine.Pin(machine.Pin.PB_14,machine.Pin.OUT,machine.Pin.PULL_FLOATING)
>>> led14.value(1)
>>> led14.value(0)
```

---

Tip #6 — Connect to an existing Access Point. To do this, we first import the network module and then initialize as a station and scan for available networks. We connect to an available network (SSID "BigHouse") with whatever the password is ("Th3Pa55word" in this example), check for a valid, live connection and finally, print out the IP address.

```
>>> import network
>>> sta_if = network.WLAN(network.STA_IF)
>>> sta_if.active(True)
True
>>> sta_if.scan()
[(b'BigHouse', b'0Z:\xc7\x82\x10', 6, -67, 32, False)]
>>> sta_if.connect("BigHouse", "Th3Pa55word")
>>> sta_if.isconnected()
True
print("Connected, ip is: " + sta_if.ifconfig()[0])
Connected, ip is: 192.168.4.3
```

Tip #7 — Create an access-point on the W600 …and then shut it down again. First we need to load the easyw600 module, which contains Winner Micro's own helper functions to make this easy. There's just a single call to easyw600.createap(), with your choice of SSID. You'll find that the new access point pops up immediately and is visible and accessible (but not very useful) from any nearby device equipped with WiFi:-

```
>>> import easyw600
>>> easyw600.createap(ssid="W600_softAP")
softap working, ip is 192.168.43.1
```

When you're finished, there's just a single call to easyw600.closeap() to shut it all down again:-

```
>>> easyw600.closeap()
```

---

Tip #8 – Enable an FTP server process. This will allow us to manipulate files on the W600, so that longer MicroPython programs can be written elsewhere and then debugged on the device itself. Before you start with the FTP server, you need to make sure that your W600 is already on-line and associated with an existing AP (see tip #6, above).

The FTP server is included in yet another module, this time named simply "w600", so to begin with, we need to make sure that it is imported. After that, there's just a one line command to start the FTP server with a given username and password (note that in the given example, the username and password are both set literally to the word "None"):-

```
>>> import w600
>>> w600.run_ftpserver(port=21,username="None",password="None")
ftpserver is running.
```

[*After writing this tip, I found that the "easyw600" module has a wrapper for run_ftpserver() which is even easier to use, just:-* "easyw600.ftpserver()" ]

You can now access your W600 from any local machine and you'll see something like this:-

```
Connected to 192.168.4.3.
220-= welcome on W600 FTP server =-
220
Name (192.168.4.3:guest): None
331 Password required for None
Password:
230 User logged in
Remote system type is UNIX.
ftp> ls
200 Port Command Successful.
150 Opening Binary mode connection for file list.
drwxrwxrwx 0 root root 0 Jan 1 2018 sys
drwxrwxrwx 0 root root 0 Jan 1 2018 lib
drwxrwxrwx 0 root root 0 Jan 1 2018 cert
-rwxrwxrwx 0 root root 139 Jan 1 2018 boot.py
-rwxrwxrwx 0 root root 34 Jan 1 2018 main.py
-rwxrwxrwx 0 root root 40 Jan 1 2018 easyw600.py
226 Transfert Complete.
ftp>
```

So, we can now create a file, “fc.py” containing Python code on our local Linux machine:-

```
def wibble(FC="dog"):
    print("Furry creature: " + FC)

wibble("cat")
wibble()
wibble("ferret")
```

Now upload it to the W600 using FTP’s “put” command and then go back to the W600 console and type:-

```
>>> import fc
```

…(note:- there’s no “.py” on the end) and you’ll see the output from your new code.

If you repeat the “import fc” command, there’s no further output, but you can now call the “wibble” function from the REPL command line. Try:-

```
>>> fc.wibble()
```

and

```
>>> fc.wibble("rabbit")
```

Note that it isn’t really possible to un-import a file (or module) once it has been cached, so -don’t- edit your file remotely and then expect to see the changes reflected when it is re-uploaded. MicroPython generally ignores an import request for something which it already has marked as existing in cache. The only bullet-proof way around this is to reboot the W600.

---

Tip #9 – How do I reboot the W600 (without pulling the power cord)?

```
>>> import machine

>>> machine.reset()
```

---

Tip #10 – And finally, here's a replacement main.py file which you can now upload to your W600 to have it automatically connect to your WiFi network and start the FTP server (both as described above) on reboot. You -will- need to set the SSID and password for your access point before installing it.

```
##
## main.py file content. Used to automatically start
## the network, join an access-point and then start
## an FTP server on the W600 at initial boot.
##
## *** You must set an SSID and password ***
##
import network
import w600
import time

print("Starting network and FTP server…")

##
## Join Existing Access-Point (Local Wireless Network).
##
sta_if = network.WLAN(network.STA_IF)
sta_if.active(True)
sta_if.scan()
sta_if.connect("MY_SSID", "MY_PASSWORD")

##
## Start built-in FTP Server.
##
w600.run_ftpserver(port=21,username="None",password="None")

##
## …and let the user know what our address is.
##
print("Waiting for DHCP…")
time.sleep(5)
sta_if.isconnected()
print("Connected, ip is: " + sta_if.ifconfig()[0])
```

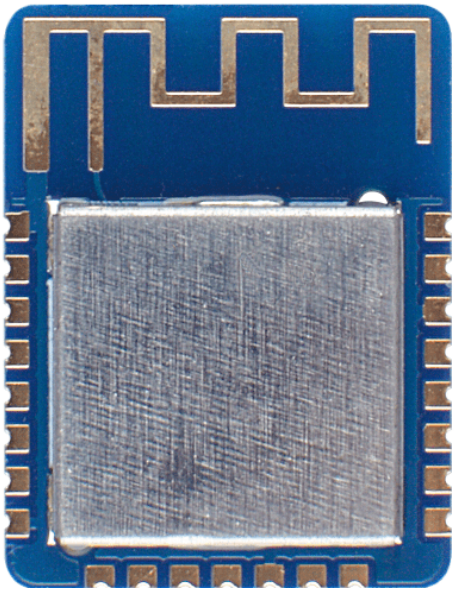
---

3 Comments

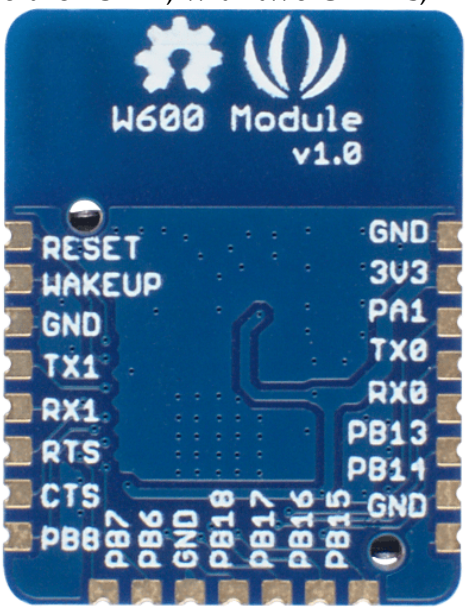
# Son of YAESPK (Yet Another ESP Killer) (https://esp8266hints.wordpress.com/2019/04/27/son-of-yaespk-yet-another-esp-killer/)

April 27, 2019April 29, 2019 PuceBaboon Hardware, W600 ARM, Cortex M3

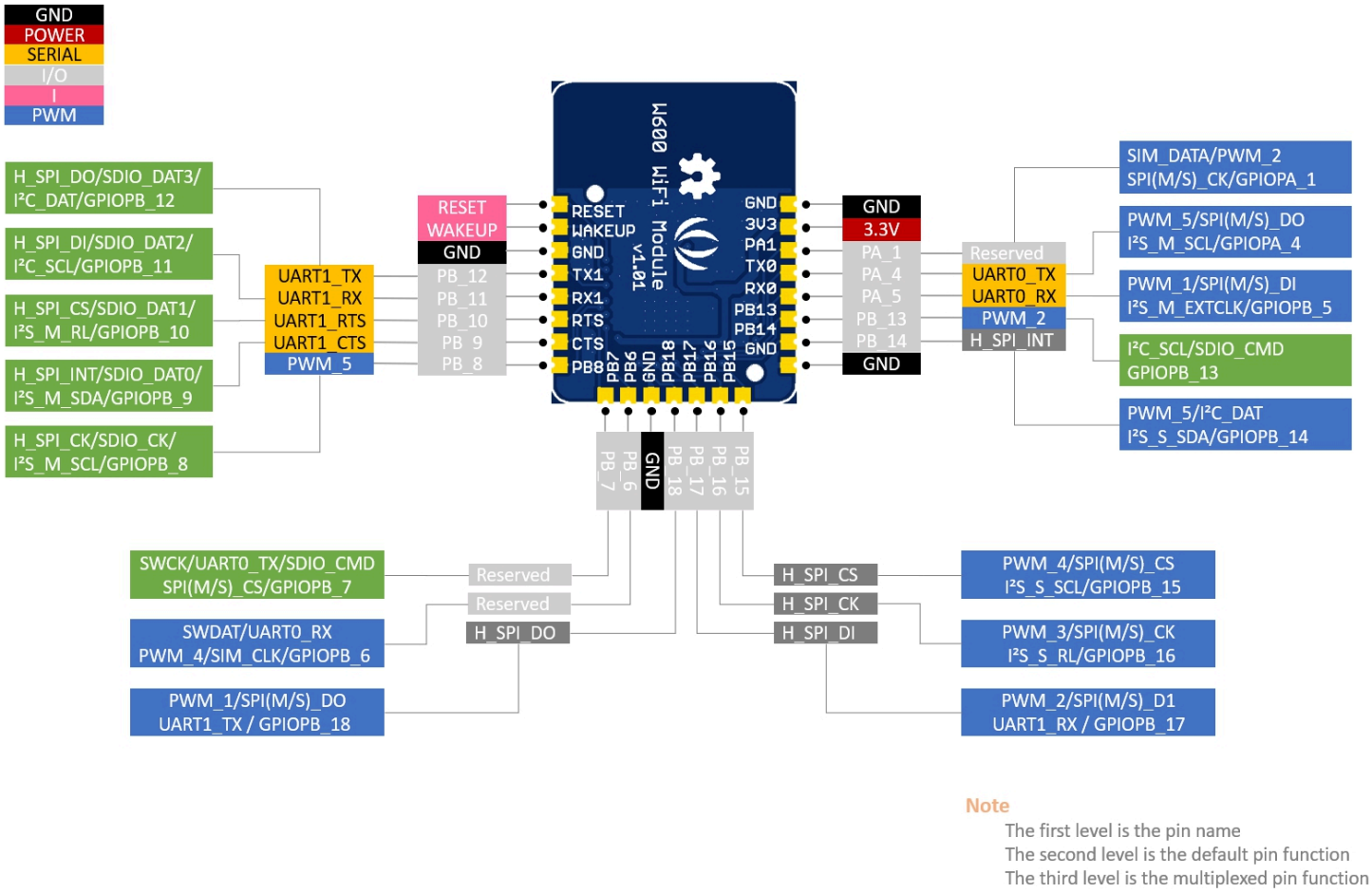
Seeed Studio have just announced another module based on the Winner Micro W600 ARM SOC. (https://www.seeedstudio.com/W600-Module-p-4020.html) This one is slightly more expensive than the previous two offerings, but does seem to be a much more sensible design for hobbyists and, with Seeed touting CE/FCC certification for it, attractive to those looking to produce WiFi enabled commercial products, too. It looks very similar to the ESP12 series modules, with an on-board, PCB antenna and castellated GPIO connectors on three sides of the board. The metal RFI shield completely covers all of the components, with only the antenna showing externally. On the version shown on the Seeed pre-order page, the silk-screen pin numbers are only on the bottom of the module and, given the size of the RFI shield, I wouldn't expect that to change.



The pin-out also looks very similar to the ESP12, with two UARTs, I2C, H-SPI and I2S interfaces. Most of the pins share multiple functions and can also be used as standard GPIO or PWM outputs. As noted with the previous offerings based on the W600, the processor is a single-core Cortex-M3 with 288KB of RAM and 1MB of on-board flash. The SOC also features an on-board hardware cryptographic accelerator and an ISO/IEC "Smart-Card" 7816 interface.




In yet another echo of an ESP-based past, there is mention of an AT interface mode for communication between an external microprocessor and the W600. The documentation available on the Winner Micro site (http://www.winnermicro.com/en/html/1/156/158/497.html)does include a Linux based install guide for the GCC toolchain and an SDK User's guide, along with several other useful PDF manuals, though.




(https://esp8266hints.wordpress.com/wp-content/uploads/2019/04/w600_module_pinout.jpg)

---

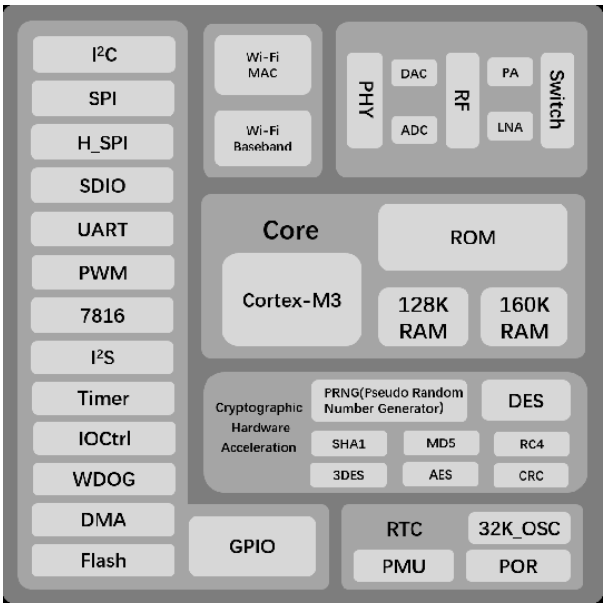
The module is scheduled to be available for shipping from Seeed on May 22nd at $3.79 per unit, or $3.59 in quantities larger than 20.

Leave a comment

# YAESPK (Yet Another ESP Killer) (https://esp8266hints.wordpress.com/2018/09/26/yaespk-yet-another-esp-killer/)

September 26, 2018 PuceBaboon Hardware, W600 ARM, Cortex M3, WinnerMicro

Seeed Studios is currently advertising the Air602 WiFi module at a price of $1.90 (https://www.seeedstudio.com/Air602-WiFi-Module-p-3139.html) for single unit quantities. The Air602 is tiny (smaller than a postage stamp and smaller than the ESP8266), but before you whip out your credit card to order half a dozen, you need to realize that one reason it's so small is that there is no on-board antenna and no antenna socket, just a normal, edge connect pin on the board.




(https://esp8266hints.wordpress.com/wp-content/uploads/2018/09/winnermicro_w600_block-diagram.jpg)This thing in not an ESP of any variety. It is based on the WinnerMicro W600 SOC, which has a Cortex M3 core. The block diagram looks very impressive (although at the moment, the WinnerMicro web site is much less impressive and is doing an excellent impression of a honey-pot/tar-pit trap for unwary browsers), with dual UARTs, I2C, SPI and I2S interfaces, an RTC and hardware crypto all baked in.

The Air602 module has a limited number of pinouts (more than the ESP-01, though), so the available interfaces are fairly limited. There are 12 pins available, but with two ground pins, one 3v3, one antenna and one (input only) reset, that leaves only 7 pins available for I/O and those seem to be split between the two UARTs as primary (with the SPI bus as alternate use) and the single GPIO-8 available as unassigned. I'm guessing that the reason for the two pins being assigned to GND is an attempt to make it easier to add a micro antenna socket (the grounds are either side of the antenna pin).

The SDK zip is available from Seeed's site and, given the current, sloooo-o-o-w state of the WinnerMicro web site, that's where I'd advise you to go for more info. The module itself seems to ship with LUA and an AT command set implementation as standard (again, shades of the original ESP-01). Seeed also have a WiFi development board, based on the Air602 module, for only $2.90 (https://www.seeedstudio.com/Air602-WiFi-Development-Board-p-3140.html?utm_source=mailchimp&utm_medium=edm&utm_campaign=bazaar_0925), which is a give and take proposition. It has an antenna (big plus) and the PCB is made to plug directly into a USB socket, but only 5 of the W600's IO pins are available as through-hole connections.

5 Comments