

Photos: <https://photos.app.goo.gl/JBW4wv51Eb6RX1ej8>

PLD-PZ005-V1.9 [0, 0]

Contents: [Dobrica PavlinuÅ;jiÄ 's random unstructured stuff]

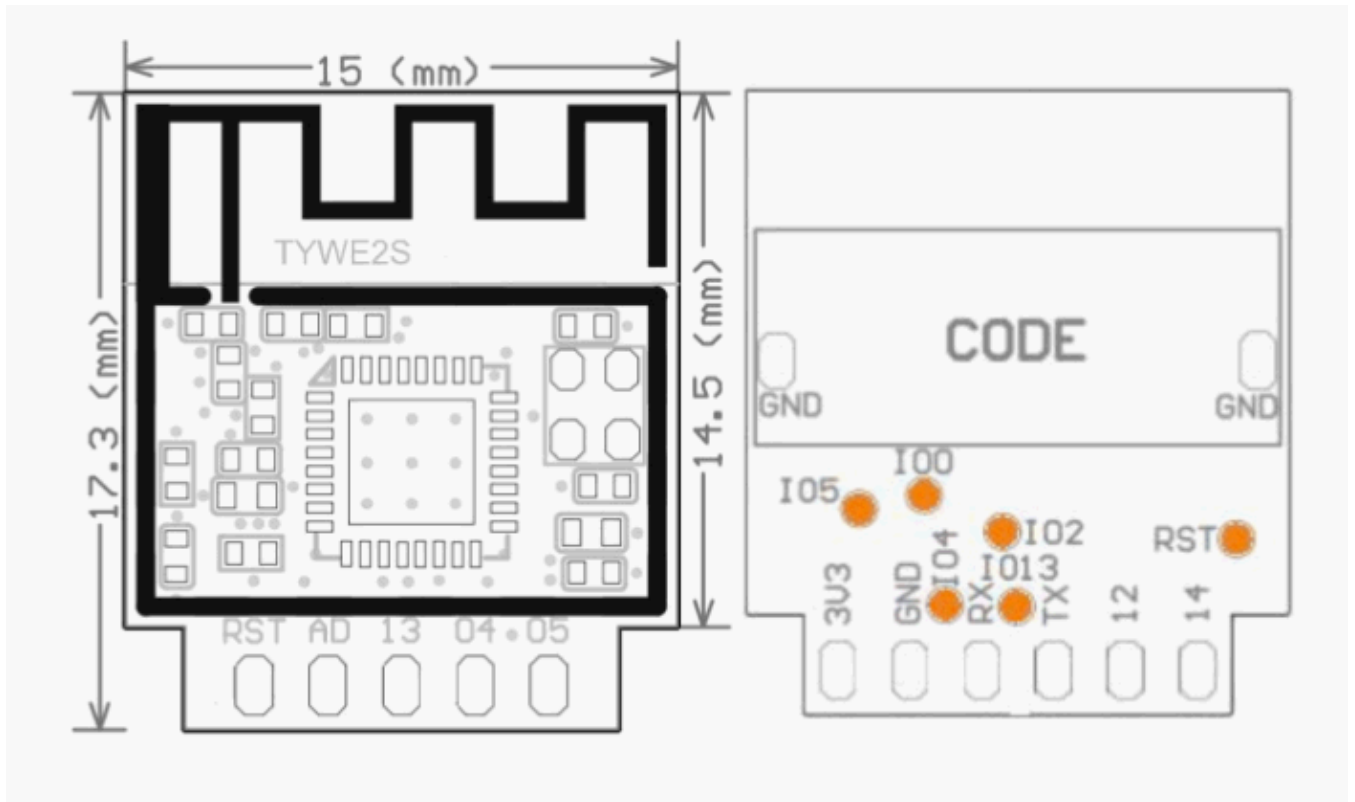
- Dobrica PavlinuÅ;jiÄ 's random unstructured stuff (Board markings)
- Dobrica PavlinuÅ;jiÄ 's random unstructured stuff (wifi module)
- Dobrica PavlinuÅ;jiÄ 's random unstructured stuff (sonoff tasmota)
 - ◆ Dobrica PavlinuÅ;jiÄ 's random unstructured stuff (enable user config)
 - ◆ Dobrica PavlinuÅ;jiÄ 's random unstructured stuff (build)
 - ◆ Dobrica PavlinuÅ;jiÄ 's random unstructured stuff (flashing)
 - ◇ Dobrica PavlinuÅ;jiÄ 's random unstructured stuff (erase flash)
 - ◇ Dobrica PavlinuÅ;jiÄ 's random unstructured stuff (write flash)
- Dobrica PavlinuÅ;jiÄ 's random unstructured stuff (first powerup)

Board markings

PLD-PZ005-V1.9.

12/13/2018

wifi module



TYWE2S

pins on flat side (bottom)

3v3
 gnd
 rx
 tx
 12
 14

pins on side with components

rst
 ad
 13
 04
 05

sonoff tasmota

<https://github.com/arendst/Sonoff-Tasmota/issues/3950>

```
{ "NAME": "Teckin", "GPIO": [0, 17, 0, 53, 134, 132, 0, 0, 131, 56, 21, 0, 0], "FLAG": 0, "BASE": 52 }
```

enable user config

```
dpavlin@nuc:/nuc/esp8266/Sonoff-Tasmota$ git diff
diff --git a/platformio.ini b/platformio.ini
index 59fab723..98d285fc 100755
--- a/platformio.ini
+++ b/platformio.ini
@@ -193,7 +193,7 @@ build_flags = ${core_active.build_flags}
;                                     -DFIRMWARE_BASIC
;                                     -DFIRMWARE_KNX_NO_EMULATION
;                                     -DFIRMWARE_DISPLAYS
-;                                     -DUSE_CONFIG_OVERRIDE
+                                     -DUSE_CONFIG_OVERRIDE

; *** Fix espressif8266@1.7.0 induced undesired all warnings
build_unflags = -Wall

dpavlin@nuc:/nuc/esp8266/Sonoff-Tasmota$ vi sonoff/user_config_override.h
```

Configure defaults for your device and build only sonoff firmware (we don't need all variants)

build

```
dpavlin@nuc:/nuc/esp8266/Sonoff-Tasmota$ platformio run -e sonoff
```

flashing

connect IO0 to GND to get in bootloader

erase flash

Flashing sonoff doesn't work for me reliably if I don't erase flash first. I suspect that config area is corrupted or something similar.

```
dpavlin@x200:/mnt/nuc/esp8266/esptool$ ./esptool.py --port /dev/ttyUSB2 erase_flash
esptool.py v2.8-dev
Serial port /dev/ttyUSB2
Connecting....
Detecting chip type... ESP8266
Chip is ESP8285
Features: WiFi, Embedded Flash
Crystal is 26MHz
MAC: c4:4f:33:87:b1:bd
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 6.6s
Hard resetting via RTS pin...
```

write_flash

```
dpavlin@x200:/mnt/nuc/esp8266/esptool$ ./esptool.py --port /dev/ttyUSB2 write_flash 0x0 ../Sonoff
esptool.py v2.8-dev
Serial port /dev/ttyUSB2
Connecting....
Detecting chip type... ESP8266
Chip is ESP8285
Features: WiFi, Embedded Flash
Crystal is 26MHz
MAC: dc:4f:22:ee:a2:4a
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 1MB
Compressed 565472 bytes to 388329...
Wrote 565472 bytes (388329 compressed) at 0x00000000 in 34.3 seconds (effective 131.8 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Please note that 565472 is bigger than 512Kb which means that OTA update won't fit if firmware size isn't reduced.

```
dpavlin@x200:/mnt/nuc/esp8266/esptool$ ./esptool.py --port /dev/ttyUSB2 write_flash 0x0 ../Sonoff
esptool.py v2.8-dev
Serial port /dev/ttyUSB2
Connecting....
Detecting chip type... ESP8266
Chip is ESP8285
Features: WiFi, Embedded Flash
Crystal is 26MHz
MAC: c4:4f:33:87:b1:bd
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 1MB
Compressed 491040 bytes to 339235...
Wrote 491040 bytes (339235 compressed) at 0x00000000 in 30.0 seconds (effective 131.0 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Remove bridge from IO0 to GND and test module.

first powerup

open web, toggle relay

verify that voltage seems sane

Wifi > Hostname

Mqtt > Topic

other > friendly name (displayed on web UI)

Logging parameters > Telemetry period - 10 # report sensor status every 10 sec

open console, and read voltage at output:

```
18:09:28 CMD: VoltageSet 233.9
```

```
18:09:28 MQT: stat/lemilica/RESULT = {"VoltageSetCal":1724}
```