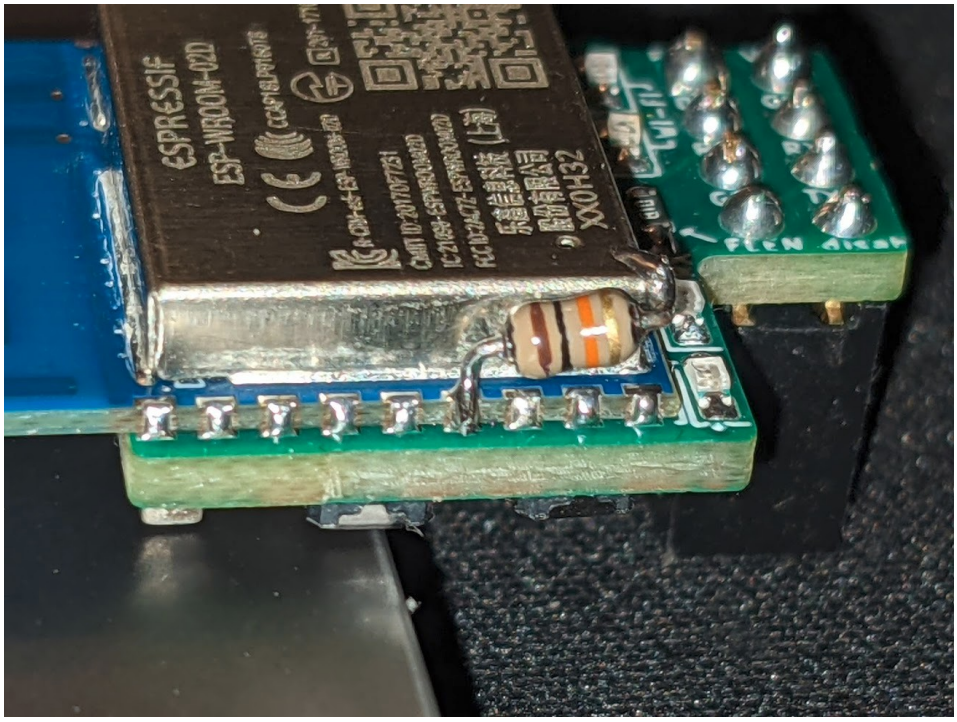
Assembly instructions

Errata (all board types – v0.1 only)

A pull-down resistor is missing on GPIO15 to ground on v0.1 boards. Solder a 10 Kohm resistor from GPIO15 to ground or to the ESP8266 shield.

v0.2 boards have this pull-down resistor built-in so an external resistor is not required.



Wi-Se rewirable

1. The board is pre-wired for 3.3V VCC input. If you want to use 5V, on the bottom side cut the left trace between the two right pads of the 3-way solder jumper on the left, put a solder blob on the 2-way solder jumper on the right.
2. Solder the two buttons on both sides
3. Solder the LEDs on the top side – power them externally to determine the polarity. The cathode is close to the edge of the board, the anode points inside.
   From right to left, looking at the board with the pin headers on top: blue, yellow, red, red
4. Solder the ESP8266.
5. **(v0.1 only)** Solder the 10K resistor from GPIO15 to ground, as shown in the picture in the Errata section
6. Put 4 solder blobs on the solder jumper matrix to connect the pin header pins to the preferred nets.

7. Solder the pin header. The DTR/RTS header is optional. You can also flash the board by holding down the BOOT0 button while you reset the board. If you solder it you should connect it to the same-named pins on your UART adapter, or to GPIOs on your target board.

Wi-Se raspi header

1. Solder the two buttons on the bottom side
2. Solder the LEDs on the top side – power them externally to determine the polarity. The cathode points to the bottom of the board, when looking at it with the raspi header on the right.
   From the top down: blue, yellow, red, red. Note that the footprint in the middle is not for LEDs.
3. If you want to be able to flash it from the SBC without pressing buttons, solder a 0 ohm resistor in the middle footprint.
4. Solder the ESP8266.
5. **(v0.1 only)** Solder the 10K resistor from GPIO15 to ground, as shown in the picture in the Errata section
6. Solder a female pin header. The header MUST STICK OUT FROM THE SIDE OPPOSITE TO THE ESP8266

Wi-Se Orange Pi 4

1. Solder the two buttons on both sides
2. Solder the LEDs on the top side – power them externally to determine the polarity. The cathode has to go on the side where the silkscreen has a [marking.
   From the top down, looking at the board with the pin headers labels in the readable orientation: blue, yellow, red, red. Note that the big footprint in the middle is not for LEDs.
3. Solder the 10uF capacitor on the bigger footprint, in the middle on the right side.
4. Solder the ESP8266
5. Solder two 1Kohm resistors on the pads on the bottom left
6. **(v0.1 only)** Solder the 10K resistor from GPIO15 to ground, as shown in the picture in the Errata section
7. Solder a female pin header on the top side. The header MUST STICK OUT FROM THE SIDE OPPOSITE TO THE ESP8266
8. Optionally solder a 90° header on the bottom side, or a straight header sticking out from the top side. You can also flash the board by holding down the BOOT0 button while you reset the board. If you solder it you should connect it to the same-named pins on your UART adapter, or to GPIOs on your target board

9. If you don’t want to solder an entire header you still need to connect VCC to 5V. You can solder a wire directly to the VCC pin of the bottom header.

Testing and troubleshooting

The RPI and OPI boards must be powered with 5V. The rewirable board must be powered with 3.3V unless you change the solder blob configuration as described above.

First of all, measure how much current the board draws. The RPI and OPI boards should draw around 40 mA. The rewirable board on 3.3V should draw around 30 mA. All boards should draw a very small amount of current when you hold down RESET, and the current amount should not change if you press the BOOT0/IO0 button or when you touch them.

If the current value does not make sense, there’s probably leftover conductive flux on the board. Clean it thoroughly. A ultrasonic cleaner may be required.

Once the board is clean, connect it to a UART adapter and run a terminal client program with UART settings: 74880n1.
The board should report “boot mode:(3,6)” when the reset button is pressed (embedded flash boot mode), “boot mode:(1,6)” when the BOOT0/IO0 button is held down while the reset button is pressed (esptool bootloader mode).

The second value of “boot mode” can be anything but if it isn’t 6 there may be other issues.

Meaning of the first value:

Value of “m”	**GPIO 15**	**GPIO 0**	**GPIO 2**	**Mode**	Notes
0	0	0	0	Invalid	
1	0	0	1	UART	Esptool mode
2	0	1	0	Invalid	
3	0	1	1	Flash	Normal boot mode
4	1	0	0	SDIO	
5	1	0	1	SDIO	
6	1	1	0	SDIO	
7	1	1	1	SDIO	

When the first boot mode number is ≥ 4, the GPIO15 pull down is not working and the board is booting in SDIO Wi-Fi adapter mode. Reflow the pull-down resistor.

Reset causes:

0 Unidentified
1 Power reboot or normal boot
2 External reset using reset pin or wake up from deep sleep
3 Software reset

4 Hardware watchdog (WDT) reset

You should normally see rst reason 2. If you see other reset reasons, there might be issues with the board.

Once the board is booting correctly when the strapping buttons are or are not pressed, you can proceed to flashing the board.

See firmware-specific instructions in the firmware GitHub page: https://github.com/Depau/wi-se-sw

To flash the board, connect it to a USB UART adapter and power it. Make sure you use a 3.3V UART adapter. Use a terminal client program to ensure the UART lines aren't swapped: after pressing reset you should receive a boot message from the bootrom at 74880n1.

Hold down the GPIO0/IO0 button; while still holding press RESET once, then release the GPIO0 button. While in bootloader mode, some LEDs will be dimly lit. This will also happen when the ESP8266 SoM is brand-new and the SPI flash doesn't contain any program. In this case, ensure the the boot mode reports "boot mode:(1,#)".
Once this is done you can run the Wi-Se builder script – ensure it is set to "serial" flashing and that the serial port is correct.

After flashing, press RESET once to reboot into the new firmware.

After flashing the boards, the blue LED will blink smoothly and the yellow LED will stay as long as the board is not connected to a network or as long as the soft access-point isn't ready. After connection, or after the access-point is up and running, the blue LED will stay on and the yellow LED will turn off.

During normal operation, the RX and TX LEDs will blink when data is received from/transmitted to UART. The yellow LED will blink when software flow control was triggered.

The RX/TX LEDs will not blink if data is received while no client is connected. When no client is connected the firmware will ignore all data coming from the UART.