# N0DY Wi-Fi Relay Hardware User’s Guide

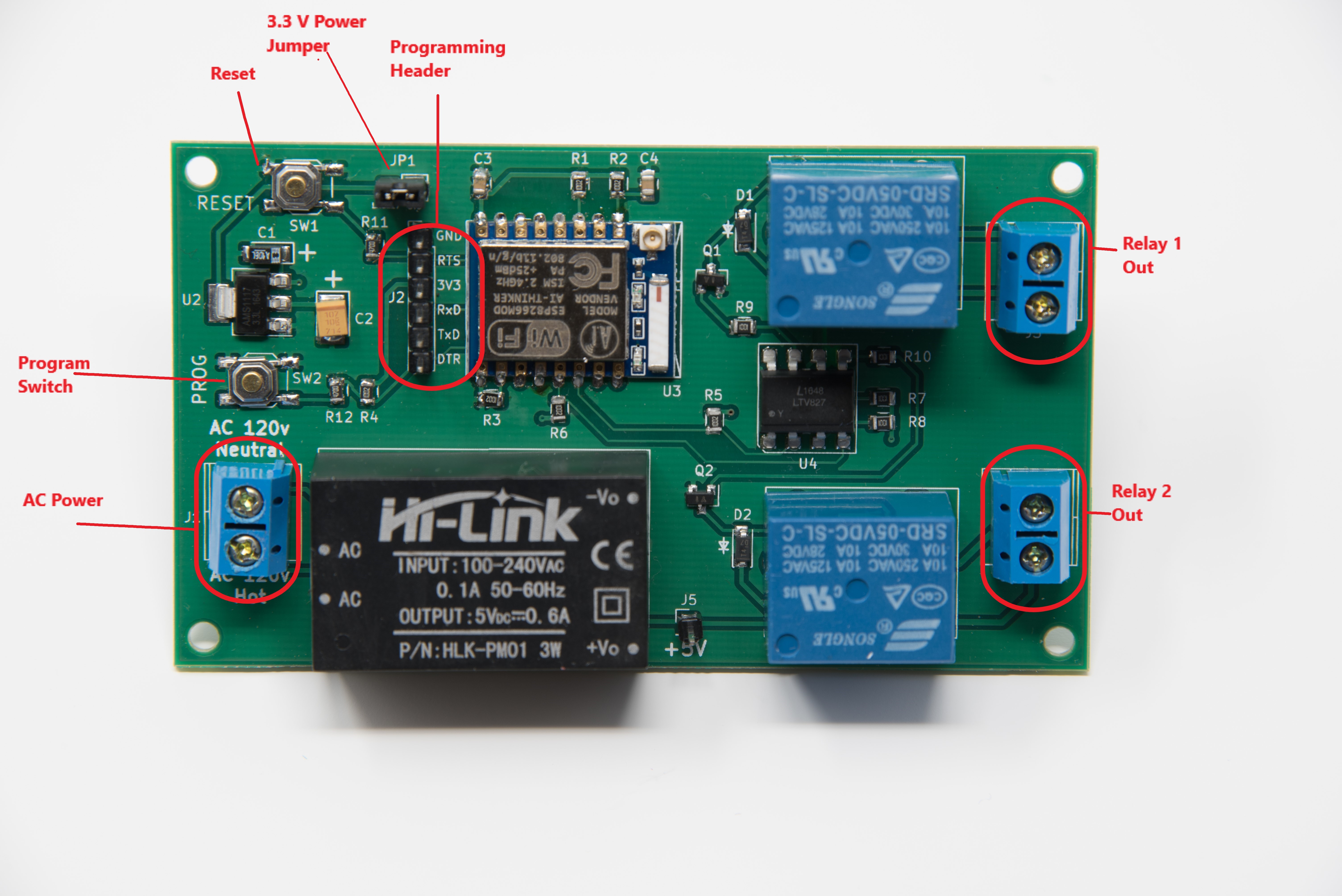


Figure 1-Hardware Breakout

**AC Power**

Input connector for primary AC power. The DEVICE\_NAME works with alternating current 100 – 240 volts, 50-60 Hz. Exercise caution as the pins for the connector and the power module are exposed on the bottom side of the board, creating the potential for shock and short circuits.

**Relay 1 Out, Relay 2 Out**

Output connectors for the two controlled relays. The relay contacts are rated for 10A non inductive loads up to 240 volts AC.

**3.3V Power Jumper**

This jumper allows the processor and 3.3v circuits to be powered either from the AC / DC converter (normal case), or through the 3.3v pin on the programming header (used when flashing the processor). If your FTDI device supplies 3.3v power, move the jumper to connect the pins labelled “3V3” and “USB” and disconnect the DEVICE\_NAME from AC power. For normal operation, connect the pins labelled “3V3” and “LM1117”.

**Programming Header**

This header allows access to various signals useful for flashing the ESP8266 device, communicating over serial ports, providing 3.3v power, and GPIO 0. All pins are labelled according to their function on the DEVICE\_NAME.

**GND** – Ground reference.

**RESET** – Connects to ESP8266 RESET and RESET switch. Frequently used to connect to an RTS signal for flashing from an Arduino IDE.

**3V3** – Used to supply 3.3 volt power to the board when programming. Set the 3.3v power jumper appropriately when using this pin, and disconnect AC power.

**TXD** – Connects to the ESP8266 serial transmit data pin. Connect this pin to the RxD of an FTDI device. Normally 115,200 bits/sec.

**RXD** – Connects to the ESP8266 serial receive data pin. Connect this pin to the TxD of an FTDI device. Normally 115,200 bits/sec.

**GPIO 0** – Connects to ESP8266 GPIO 0 and the PROG switch. Frequently used to connect to a DTR signal for flashing from an Arduino IDE. Can also be used as a general purpose IO pin. See “Hacking the N0DY Wi-Fi Relay” for details.

**Program Switch**

Connects to the ESP8266 GPIO 0. Used for multiple purposes.

Flashing – Holding down the PROG button while simultaneously pushing and releasing the RESET button puts the ESP8266 into programming mode. This is a manual alternative to using the RTS / DTR combination with an Arduino IDE style flashing program.

<TODO CHECK>1 short press: Toggles Relay 1. If MQTT is configured (see SOFTWARE\_GUIDE) sends a MQTT status message like stat/sonoff/LIGHT on. If cmnd/sonoff/ButtonRetain on has been used the MQTT message will also contain the MQTT retain flag.

<TODO CHECK>2 short presses: Toggles Relay 2. This will blink the LED twice and sends a MQTT status message like stat/sonoff/POWER on.

<TODO CHECK>3 short presses: Start Wifi smartconfig allowing for SSID and Password configuration using an Android mobile phone with the ESP8266 SmartConfig app. The MQTT server still needs to be configured in the user\_config.h file. The LED will blink during the config period. A single button press during this period will abort and restart sonoff.

4 short presses: Start Wifi manager providing an Access Point with IP address 192.168.4.1 and a web server allowing the configuration of both Wifi and MQTT parameters. A single button press during this period will abort and restart sonoff.

5 short presses: Start Wifi Protected Setup (WPS) allowing for SSID and Password configuration using the routers WPS button or webpage. A single button press during this period will abort and restart sonoff.

6 short presses: Will restart the module

7 short presses: Start OTA download of firmware.

Pressing the button for over four seconds: Reset settings to defaults as defined in user\_config.h and restarts the device