

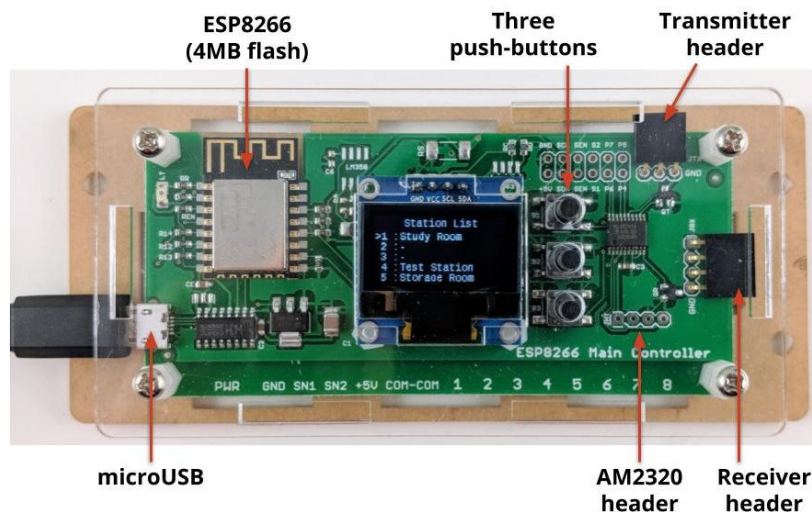
# RFToy 3.0 User Manual

(Dec 16, 2017)

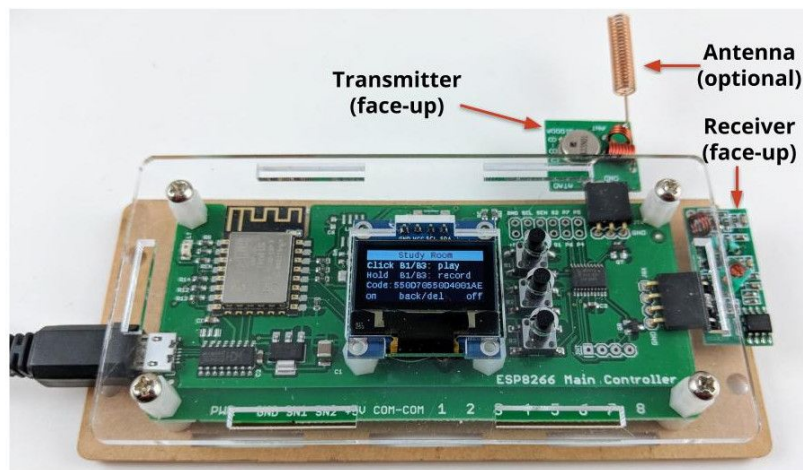
## Specifications

- A WiFi-enabled gadget for interfacing with RF wireless power sockets and RF remote sensors.
- **ESP8266 WiFi chip (4MB flash)** with CH340G USB-serial converter, Arduino-compatible (with ESP8266 core).
- One **128×64 OLED** display, three tactile buttons, one light sensor.
- Pin headers for 433/315 MHz RF transmitter and receiver modules, and MOSFET power switches for them.
- Pin header for AM2320 temperature/humidity sensor.
- Supports WiFi and web-based control; supports Over-the-Air (OTA) firmware update, and raw RF signal sampling.
- One microUSB connector for providing power (also supporting USB-based firmware update).
- Acrylic front and back panel covers. Size: 45mm x 100mm (1.8" x 3.9")
- This is a completely open-source product.

## Hardware Interface



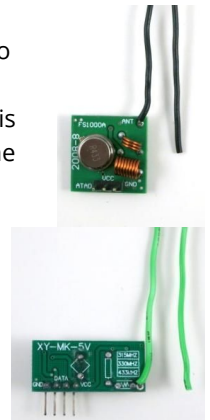
## Plugging in RF Modules



**Both the transmitter and receiver should be plugged in facing up.** If you use your own transmitter and receiver, you should verify the orientation by checking that the GND pin on the transmitter and receiver match the GND pin on the silkscreen printed next to the pin headers.

## Antenna and Transmission Range

- The **transmitter** module does NOT have built-in antenna. It's recommended that you solder a wire to the antenna pin (marked **ANT**) to extend the transmission range. The recommended wire length is 17cm (6.7 inch) for 433MHz transmitter, and 24cm (9.4inch) for 315MHz transmitter. The wire length is calculated based on 1/4 of the wave length. As the antenna is quite long, if it's not possible to keep the antenna straight, you can fold it down in half or curl it. See pictures on the right for example.
- The **receiver** module also does NOT have built-in antenna. Generally you do not need to solder a wire antenna since the receiver is only used for decoding RF signal, and you can place it next to your RF remote so even without antenna it receives the signal well. However, if you plan to use the receiver for longer distance reception, such as sniffing signals from RF wireless temperature and weather sensors, adding an antenna (same length as transmitter) is highly recommended.



## Acrylic Cover Assembly

RFToy 3.0 comes with a set of acrylic front and back covers together with screws and support pillars. To assemble:

1. Take the **back panel** (the longer one, with four wall-mount holes on the sides), and insert the **four long screws** down into the inner four screw holes. Use your fingers to hold the four screws in place, and then flip the panel so that the screws face up, and place the assembly on a flat surface.
2. Insert the four **6mm long separation pillars** (these have smooth interiors) into the screws. Then place the RFToy circuit board through the screws onto the separation pillars.
3. Screw and tighten the four **7mm long support pillars** (these have screw threads in the interior) onto the screws.
4. Take the **front panel**, carefully place it on the top and align the screw holes with the support pillars. Note the front panel has **only one correct orientation** such that the button holes align well with the pushbuttons. Finally, insert the **four short screws** into the screw hole and tighten the screws to secure the front panel.
5. (Optional) If you would like the front panel to be transparent, you can peel off the protective paper cover. To do so, use a coin or your fingernail to carefully lift the paper from a corner, then slowly peel off the protective paper.

## Power Up and Manual Mode

To power RFToy 3.0, insert a microUSB cable into the microUSB jack on RFToy, and the other end to a USB power source, such as your computer's USB port, or a USB power adapter, or a USB power bank. Upon powering up, RFToy will prompt you to select whether you want to enter WiFi mode or Manual Mode. **Click B3 will enter Manual Mode**, where WiFi functions are disabled. If you don't click any button, it will time out in 15 seconds and enter WiFi mode automatically.

Next, you will see a page of brief instructions about how to use buttons. Basically:

- Click B1 or B3 to navigate up/down the station list. Click B2 to enter the selected station.
- Long press B1 to reset everything to factory default.
- Long press B2 to reset WiFi settings.
- Long press B3 to display device IP address and MAC (IP address will be 0 in manual mode).

After you have entered a selected station, you will see further instructions:

- Long press B1 or B3 to record ON/OFF signals (this requires the RF receiver to be plugged in, and you should click a button on your RF remote within 10 seconds, otherwise it will time out and go back to the station screen).
- Long press B2 to confirm and delete the station data.
- Click B1 or B3 to play back the ON/OFF signals (this requires the RF transmitter to be plugged in).
- Click B2 to go back to the station list

The default firmware has a capacity of 50 stations, which can be extended by modifying the firmware.

## WiFi Mode

When powering up RFToy, if you **click B1** or wait till the initial screen times out, it will enter WiFi mode. The WiFi mode provides additional functionality, including web-based control, the ability to change station names, and Over-the-Air Firmware Update (i.e. update firmware through WiFi). If this is the first time it's placed in WiFi mode:

- The device will start in AP mode, presenting a WiFi AP with SSID (ESP\_XXXXXX) printed on the LCD screen.
- Use your phone or computer to connect to this SSID (it's an open WiFi so no password). Then follow the sign-in instructions (or if you don't see any sign-in instructions, open a browser and manually type 192.168.4.1, then you will see instructions to further input your home router's SSID and password, in order to allow RFToy to log on to your home WiFi). Once this is completed, the **IP address** assigned to RFToy will be briefly display on the LCD. RFToy will then record the settings in non-volatile memory and reboot to connect to your home WiFi.

Following the WiFi setup, every time RFToy is booted into WiFi mode, it will log on to your home WiFi, and obtain an IP address. To find out the IP address, in the station list screen, long press B3.

Using a smartphone or computer that's connected to the same home WiFi, open a browser and type in RFToy's IP address. You will see a web UI with station list and a menu icon on the top-right. Click a station, you can change its name, and also trigger record or playback of station signal (these are the same as using buttons to record and playback signals).

In WiFi mode, all buttons still function exactly the same way as in manual mode.

## Raw Sampling

In WiFi mode, RFToy 3.0 supports raw signal sampling. This is implemented by capturing the raw RF signal into its internal memory, then displaying the signal onto the web UI. You can also trigger RFToy to playback the most recently captured raw signal. The sampling frequency is **100KHz** (i.e. one sample per 10 microseconds), and the duration is **1 second** by default. Raw sampling is useful for visualizing RF signals or analyzing ad-hoc signals that are not well documented.

To start, at the homepage click the upper-right corner menu icon, then choose 'Raw Sampling'. Once you click 'Record' the device will listen to RF signal for 1 second. Since this is very short, it's recommended that you start sending RF signal shortly before the recording starts, in order to make sure it captures the signal.

Once the signal is captured, it will be displayed in the web UI. You can zoom in to check the signal in details. You can also download the signal as raw byte data for further analysis.

To change the sampling frequency and duration, you can modify the RFToy source code. Note that the total memory required by raw sampling is **(duration\*frequency/8) bytes**. In the default setting, it's  $1*100,000/8 = 12.5\text{KB}$ . This is a considerable amount of RAM for ESP8266 (which only has a few tens of KB free RAM at run-time). So to increase duration, you should suitably reduce frequency to make sure the RAM usage doesn't increase dramatically.

## Update Firmware

### OTA Firmware Update

RFToy 3.0 supports Over-the-Air (OTA) firmware update. In WiFi mode, click the upper-right corner menu icon, and select Firmware. You can then choose a firmware file (.bin) and upload it over the web.

### USB Firmware Update

If for any reason the OTA update doesn't work, you can optionally update firmware using a microUSB cable. RFToy has built-in CH340G USB-serial chip. To begin, you may need to install CH340G driver as explained below:

- **Windows 7/8 and Linux:** no driver is needed.
- **Windows XP:** download and install driver <http://raysfiles.com/drivers/ch341ser.exe>
- **Mac OS:** download and install driver [http://raysfiles.com/drivers/ch341ser\\_mac.zip](http://raysfiles.com/drivers/ch341ser_mac.zip)

On Windows, the Serial Port name is **COM?** where ? is a number assigned to the USB-serial chip. On Linux, the Serial Port name is **/dev/ttyUSB?** where ? is a number. On Mac, the Serial Port name is **tty.wch ch341g xxx**.

After the driver is installed, you can use the open-source [esptool](#) to upload a new firmware to RFToy. Instructions are the same as any standard ESP8266 development board. Keep in mind that every time you upload a new firmware, you need to place RFToy in **bootloading mode**: to do so, unplug USB cable, then press and hold the second pushbutton (B2) while plugging the USB cable back in, then release the button. The LCD screen should remain blank, indicating it's in bootloading mode waiting for new firmware to be uploaded.

## RFToy 3.0 Hardware and Software

RFToy 3.0 is a completely open-source product. You can download its circuit design (including schematic and PCB design) and source code from:

- <https://github.com/openthingsio/rftoy>

The schematic and PCB design can be viewed and edited in [EagleCAD](#).

The source code can be compiled using [Arduino](#) installed with [ESP8266 core](#). Please follow the instructions in Github.

### Extending RFToy's Hardware Capability

You can extend or modify RFToy 3.0's Hardware Capability. For example:

- It has a built-in light sensor connected to analog pin A0, which can be used for sensing ambient light (at the time of this writing, this feature has not been integrated into the firmware yet).
- It has on-board pin header that fits an AM2320 (or similar I2C-type) Temperature / Humidity sensor (at the time of this writing, this feature has not been integrated into the firmware yet).
- There are spare GPIO pins on board to connect further sensors and actuators. For example:
  - ESP8266 GPIO12 (SEN1) and GPIO13 (SEN2)
  - P4, P5, P6, P7 are four GPIO pins on the PCF8574 IO expander (bits 4, 5, 6, 7).

## Links and Resources

- RFToy Homepage: <https://openthings.io/product/rftoy>
- RFToy Github: <https://github.com/openthingsio/rftoy>