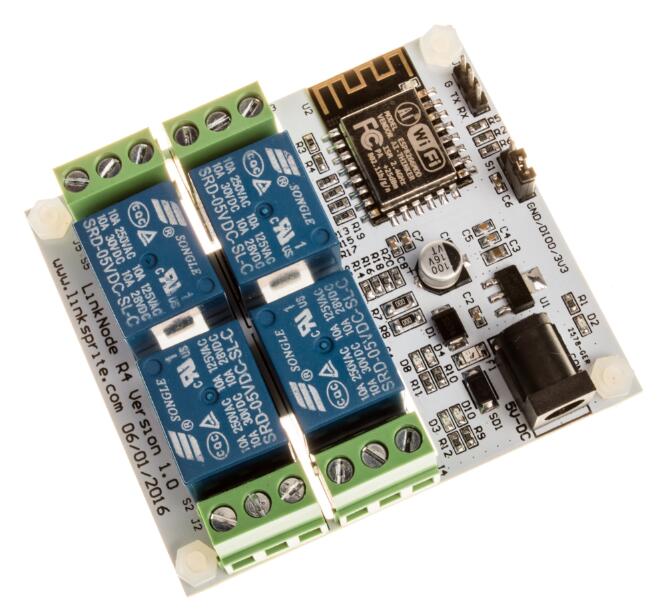
LinkNode ESP8266 4 Port Relay



1st some links to some useful information:

<http://linksprite.com/wiki/index.php5?title=LinkNode_R4:_Arduino-compatible_WiFi_relay_controller#Introduction>

Possible a slightly different model (or design)? (older model?)

<http://learn.linksprite.com/linknode/linknode-r4-arduino-compatible-wifi-relay-controller/>

Use Amazon Alexa to control linknode r4

<http://www.themakersworkbench.com/tutorial/how-use-amazon-alexa-control-linknode-r4-esp8266-4-channel-relay-board>

General ESP8266 Information (Good information)

<http://circuits4you.com/2016/12/14/esp8266-pin-diagram/>

<http://esp8266.github.io/Arduino/versions/2.0.0/doc/reference.html>

My Thoughts:

I pretty much have bad luck with just about all the Linksprite software. This was no execption.

It comes pre-loaded with software to communicate with [www.Linksprite.io](http://www.linksprite.io) which is a cloud based service like most IoT services.

The site seems fine, but isn’t clear about adding new devices. The android app doesn’t seem to talk to the site, and update new devices. The app keep crashing on me when I tried to add the device.

Which leaves only one option - install my own software - here comes the fun.

Following the wiki did not work for me.

The setting I ended up using (that work most of the time)

Generic ESP8266, 80 Mhz, 40Mhz, **DIO, 921600,** 512K (64K SPIFFS), ck, Disabled, None

The two things that changed were the upload baud rate, and DIO from QIO (DIO is the default by the way)

To get this working - I have to not only move the jumper to flash mode, but also plug the power in at just the right time. Otherwise, the programs get corrupted and the relay doesn’t work at all.

And even cycling power with setting that work doesn’t always work, some sketches (no changes make) I have to upload 3 or 4 times before they start to work. (You will know it doesn’t work, if the board keeps resetting or if you just see garbage in the serial monitor)

While we are on topic, the serial programmer you use must be 3.3v - (I wasn’t thinking, and used a 5v programmer, and may have done some damage, causing the random sketch upload fails, and the baud rate to change???)

I was unable to get either of the sketches Linksprite to work - they would compile, and upload, but appeared to never connect to [www.linksprite.io](http://www.linksprite.io)

I ended up writing my own software, I have two different ones -

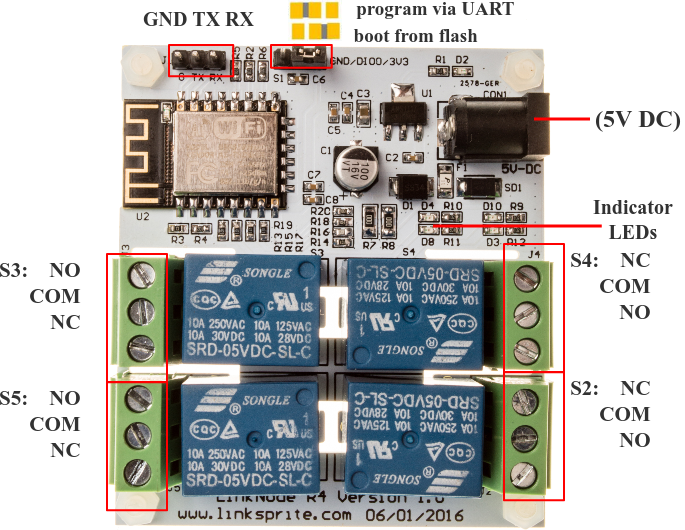
1st is a simple Website (no webpage so you have to know how to use it). This sketch will connect to your router (behind your firewall) and will run a very simple server that is waiting for the numbers 1, 2, 3, 4 (This will toggle the relays) - There is no security.

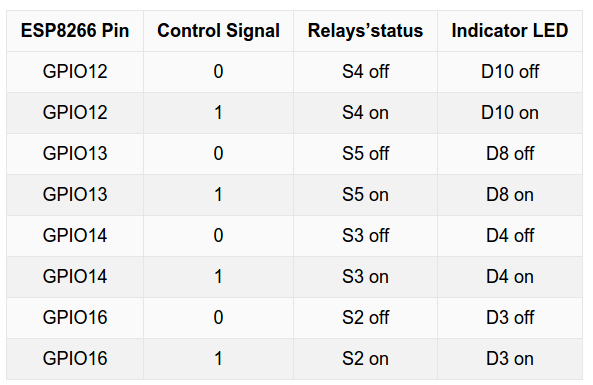
<http://wifirelay.local/1> will trigger the relay connected to D10 (S4) in the picture below.

<http://wifirelay.local/2> will trigger the relay connected to D8 (S5)

<http://wifirelay.local/3> will trigger the relay connected to D4 (S3)

<http://wifirelay.local/4> will trigger the relay connected to D3 (S2)





The simple site will respond with the status of that relay like:

Wifi Relay v1.0

Relay One: On

Sending a 1 followed by a 1 will toggle the relay on and off (or toggle the state).

My 2nd sketch uses MQTT (again there is no security is use it with caution - it open to the public) I am using a public broker (test.mosquitto.org)

Just publish the number 1, 2, 3, 4 again, and they will toggle the relay. MQTT appears to be slower as well.

Again I am calling relay 1 (GPIO12, labeled D10, and S4), 2 (GPIO13, labeled D8, and S5), 3 (GPIO14, labeled D4, and S3), and 4 (GPIO16, labeled D3, and S2)

To see the publish use something like this:

mosquitto\_pub -h test.mosquitto.org -t "relayin" -m "1"

To see status subscribe to:

mosquitto\_sub -h test.mosquitto.org -t relayout

At this point there is no command for a status of the relays - at some point I would like to add that feature.

Both sketches use the ESP8266 Wifimanager library which makes it easy to move devices from one location to another. (Where the WIFI changes) - If the device doesn’t connect to it’s current known wifi location - it will create a simple access point that allows someone to change to a different wifi station. It appears to work very well. The ESP8266 stores WIFI information for the next time it’s booted up.

Other notes:

The barrel connection is 5v 1a from what I can tell (from the website wiki) It is only 5v - I am using a 2a supply.

Github:

<https://github.com/automation-technology-club/LinkNode-R4-ESP8266-WIFI-Relay-sketches>