# LinkNode R4: Arduino-compatible WiFi relay controller

## Contents

Introduction
Product UPC
Features
LinkNode R4 Diagram
Control logic
How to use Android APP to control LinkNode R4 Steps
Tutorial <ol> <li>Get started in Arduino programming</li> <li>Remotely control LinkNode R4 with LinkSprite.IO platform</li> </ol>
Schematic
Dimension
3D printing enclosure
Video

# Introduction

LinkNode R4 is a WiFi relay controller and it is powered by ESP-12f ESP8266 WiFi module which is comptiable with Arduino programming. There are 4 relay channels and each channel allows you to control high-power devices (up to 10 A) via the on-board relay. LinkNode R4 can be used to remotely turn lights, fans and other devices on/off. The WiFi interface will allow you to associate the board with your existing WiFi network and send the commands over the network.





# **Product UPC**

Product UPC: 713382671484

### **Features**

- ESP-12f ESP8266 WiFi module
- 4 Channel relays, supporting:
  - 277V AC, 10A
  - 125V AC, 12A
- 5V DC power
- Two work modes:
  - Program via UART
  - Boot from flash
- 4 indiator LEDs

# LinkNode R4 Diagram



# **Control logic**

ESP8266 Pin	Control Signal	Relays'status	Indicator LED
GPIO12	0	S4 off	D10 off
GPIO12	1	S4 on	D10 on
GPIO13	0	S5 off	D8 off
GPIO13	1	S5 on	D8 on
GPIO14	0	S3 off	D4 off
GPIO14	1	S3 on	D4 on
GPIO16	0	S2 off	D3 off
GPIO16	1	S2 on	D3 on

# How to use Android APP to control LinkNode R4

#### LinkNode R4: Arduino-compatible WiFi relay controller - LinkSprite Playgound

LinkNode R4 is an open source 4-Channel relay controller which is powered by the ESP8266 WiFi SoC.



The following will introduce how to use Android APP to control the LinkNode R4.

#### Steps

#### 1. Install Android and iOS APP

- Download the APK file from the LinkNode-Rx-2016-12.apk (https://github.com/delongqilinksprite/LinkNodeR4-app/blob/master/LinkNode-Rx-2016-12.apk) and install it.
- if your equipment is iphone/ipad, you can get the app form appstore and name is "LinkNode Relay".
- iOS App download link : LinkNode Relay (https://itunes.apple.com/cn/app/linknode-relay/id1163468201?mt=8)

#### 2.Login LinkSpriteIO

- If you never register an account on LinkSpriteIO, please enter Email and your password, then click the REGISTER button
- If you have an account on LinkSpriteIO, please enter Email and your password, then click the SIGN IN button

# **K LinkSprite**

Email
Password (optional)
SIGN IN
REGISTER

#### 3. Go to Relay device list

Click the plus icon on the upper right quarter to add a new LinkNode R4.



4.Scan the QRcode

- Get the device information and register it to your account.



#### 5.Configure to accece WiFi AP

- Supply the power for LinkNodeR4, the LinkNode R4 will create a AP called LinkNodeAP. If it can't be detected, switch off your wifi router and try again.
- Use your mobile phone to connect this AP
- Open a browser and enter the ip address 10.0.1.1 and you will see the following website:

11:03 🚓 🎝	0.72 %	((i‡	4G ● ● ● ● ●	+
10.0.1.1				:

# LinkNodeAP

#### WiFiManager

Configure WiFi
Configure WiFi (No Scan)
Info
Reset

- Click the button configure WiFi
- Select your WiFi AP which you want to connect and enter your wifi password.
- If connecting failed, you can go to the same website to configure it again.
- After that, LinkNode R4 will connect to Linksprite IO via the internet.

#### 6.Control your 4-channel relay

• control the button on the right side to turn on or off the relays, and you can click the relay's name to change it.

BACK	LinkNode R4	$\oplus$				
			٩	R1		OP
Q R2			٩	R2		-0
R3			Р	ease input:		
Q 84			(	R	1	_
			1	CANCEL	ОК	
Q R4				CANCEL	1 ОК	ľ

## Tutorial

#### 1. Get started in Arduino programming

a. Requirements

Software

<ul> <li>Arduino IDE (https://www.arduino.cc/en/Main/Software), (the latest version is 1.6.8)</li> <li>The Arduino core (https://github.com/esp8266/Arduino) for LinkNode R4</li> </ul>	
** Hardware**	
<ul> <li>5V/1A DC power is recommanded.</li> </ul>	
http://linksprite.com/wiki/index.php?title=LinkNode_R4:_Arduino-compatible_WiFi_relay_controller	6/14

- USB TTL UART cable
- b. Install Arduino core for ESP8266
  - Install Arduino 1.6.8 from the <u>Arduino website (http://www.arduino.cc/en/main/software)</u>.
  - Start Arduino and go to File --> Preferences .
  - Enter <a href="http://arduino.esp8266.com/stable/package\_esp8266com\_index.json">http://arduino.esp8266.com/stable/package\_esp8266com\_index.json</a> into Additional Board Manager URLs field. You can add multiple URLs, separating them with commas.

Preferences	
Settings Network	
Sketchbook location:	
/home/yao/Arduino	Browse
Editor language:	System Default v (requires restart of Arduino)
Editor font size:	12
Interface scale:	Automatic 100 ‡ % (requires restart of Arduino)
Show verbose output during:	compilation upload
Compiler warnings:	None 💌
Display line numbers	
🗌 Enable Code Folding	
🗹 Verify code after upload	
Use external editor	
Check for updates on start	up
🗹 Update sketch files to new	extension on save (.pde -> .ino)
🛃 Save when verifying or uplo	pading
Additional Boards Manager URL	.s: http://arduino.esp8266.com/staging/package_esp8266com_index.json
More preferences can be edite	d directly in the file
/home/yao/.arduino15/prefer	ences.txt
(edit only when Arduino is not	running)
	OK Cancel

#### Open Boards Manager from Tools --> Board menu --> Boards Manager.

<u>File</u> Edit Sketch	Tools Help		
	Auto Format	Ctrl+T	
	Archive Sketch		
sketch_apr05a	Fix Encoding & Reload		
<pre>void setup() {</pre>	Serial Monitor	Ctrl+Shift+M	
// put your se	Serial Plotter	Ctrl+Shift+L	
}	Board: "Beta ESP318 Module"	•	Δ
woid loop() {	Flash Mode: "QIO"	÷	Boards Manager
// put your ma:	Flash Frequency: "40MHz"		Arduino AVR Boards
	CPU Frequency: "80 MHz"	÷	Arduino Yún
3	Flash Size: "2M (1M SPIFFS)"		Arduino/Genuino Uno
	Upload Speed: "115200"		Smart World
	Port		Arduino Duemilanove or Diecimila
	Programmer: "AVRISP mkII"	÷	Arduino Nano
	Burn Bootloader		Arduino/Genuino Mega or Mega 2560
			Arduino Mega ADK

Search and install esp8266 platform (and don't forget to select your ESP8266 board from Tools --> Board menu after installation).

\varTheta 🤭 Boards Manager	
Type All v esp8266	
esp8266 by ESP8266 Community Boards included in this package: Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 M Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, SparkFun Ti mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifInfo, ESPDuino. Online help More info	Aodule), NodeMCU 1.0 (ESP-12E Module), hing, SweetPea ESP-210, WeMos D1, WeMos D1

c. Check the configuration of Board

#### LinkNode R4: Arduino-compatible WiFi relay controller - LinkSprite Playgound

Because the LinkNode R4 has not been added into the official ESP8266 Arduino core repository yet, so you can't find this board on the boards list, but you can use the Generic ESP8266 Module, and select Flash Mode as QIO.

Auto FormatCtrl+TArchive SketchFix Encoding & ReloadSerial MonitorCtrl+Shift+MSerial PlotterCtrl+Shift+LBoard: "Generic ESP8266 Module">Flash Mode: "QIO">Flash Frequency: "40MHz">CPU Frequency: "80 MHz">Flash Size: "512K (64K SPIFFS)">Debug port: "Disabled">Debug Level: "None">Reset Method: "ck">Upload Speed: "115200">Programmer: "AVRISP mkII">	<u>Tools</u> <u>H</u> elp	
Archive Sketch Fix Encoding & Reload Serial Monitor Ctrl+Shift+M Serial Plotter Ctrl+Shift+L Board: "Generic ESP8266 Module" Flash Mode: "QIO" Flash Frequency: "40MHz" CPU Frequency: "80 MHz" Flash Size: "512K (64K SPIFFS)" Debug port: "Disabled" Debug Level: "None" Reset Method: "ck" Upload Speed: "115200" Port Programmer: "AVRISP mkII"	Auto Format	Ctrl+T
Fix Encoding & Reload Serial Monitor Ctrl+Shift+M Serial Plotter Ctrl+Shift+L Board: "Generic ESP8266 Module" Flash Mode: "QIO" Flash Frequency: "40MHz" CPU Frequency: "40MHz" Flash Size: "512K (64K SPIFFS)" Debug port: "Disabled" Debug Level: "None" Reset Method: "ck" Upload Speed: "115200" Port Programmer: "AVRISP mkII"	Archive Sketch	
Serial Monitor Ctrl+Shift+M Serial Plotter Ctrl+Shift+L Board: "Generic ESP8266 Module" > Flash Mode: "QIO" > Flash Frequency: "40MHz" > CPU Frequency: "80 MHz" > Flash Size: "512K (64K SPIFFS)" > Debug port: "Disabled" > Debug Level: "None" > Reset Method: "ck" > Upload Speed: "115200" > Port >	Fix Encoding & Reload	
Serial Plotter       Ctrl+Shift+L         Board: "Generic ESP8266 Module"       >         Flash Mode: "QIO"       >         Flash Frequency: "40MHz"       >         CPU Frequency: "80 MHz"       >         Flash Size: "512K (64K SPIFFS)"       >         Debug port: "Disabled"       >         Debug Level: "None"       >         Reset Method: "ck"       >         Upload Speed: "115200"       >         Programmer: "AVRISP mkII"       >	Serial Monitor	Ctrl+Shift+M
Board: "Generic ESP8266 Module"  Flash Mode: "QIO" Flash Frequency: "40MHz" CPU Frequency: "80 MHz" Flash Size: "512K (64K SPIFFS)" Debug port: "Disabled" Debug Level: "None" Reset Method: "ck" Upload Speed: "115200" Port Programmer: "AVRISP mkII"	Serial Plotter	Ctrl+Shift+L
Flash Mode: "QIO"       >         Flash Frequency: "40MHz"       >         CPU Frequency: "80 MHz"       >         Flash Size: "512K (64K SPIFFS)"       >         Debug port: "Disabled"       >         Debug Level: "None"       >         Reset Method: "ck"       >         Upload Speed: "115200"       >         Programmer: "AVRISP mkII"       >	Board: "Generic ESP8266 Module"	Þ
Flash Frequency: "40MHz"       >         CPU Frequency: "80 MHz"       >         Flash Size: "512K (64K SPIFFS)"       >         Debug port: "Disabled"       >         Debug Level: "None"       >         Reset Method: "ck"       >         Upload Speed: "115200"       >         Port       >	Flash Mode: "QIO"	•
CPU Frequency: "80 MHz"  Flash Size: "512K (64K SPIFFS)" Debug port: "Disabled" Debug Level: "None" Reset Method: "ck" Upload Speed: "115200" Port Programmer: "AVRISP mkII"	Flash Frequen <b>cy</b> : "40MHz" 🚿	Þ
Flash Size: "512K (64K SPIFFS)"       >         Debug port: "Disabled"       >         Debug Level: "None"       >         Reset Method: "ck"       >         Upload Speed: "115200"       >         Port       >	CPU Frequency: "80 MHz"	Þ
Debug port: "Disabled" > Debug Level: "None" > Reset Method: "ck" > Upload Speed: "115200" > Port > Programmer: "AVRISP mkII" >	Flash Size: "512K (64K SPIFFS)"	Þ
Debug Level: "None"  Reset Method: "ck"  Upload Speed: "115200"  Port  Programmer: "AVRISP mkII"	Debug port: "Disabled"	Þ
Reset Method: "ck" > Upload Speed: "115200" > Port Programmer: "AVRISP mkII" >	Debug Level: "None"	Þ
Upload Speed: "115200" > Port Programmer: "AVRISP mkII" >	Reset Method: "ck"	Þ
Port  Programmer: "AVRISP mkll"	Upload Speed: "115200"	Þ
Programmer: "AVRISP mkil"	Port	Þ
	Programmer: "AVRISP mkII"	Þ
Burn Bootloader	Burn Bootloader	

d. Create a Arduino Project

#### Enter the following source code and compile

/*Turn on and off the S3 relay in every second */	
void setun()	
Serial.begin(9600);	
}	
void loop()	
digitalWrite(12 HIGH).	
Sapid ninth//Palay (N/n"):	
delau, pintun (keidy Ok(h.)),	
delay(1000);	
digitalWrite(12,LOW);	1
Serial.println("Relay OFF\n");	
delay(1000);	





- Jump out the S5 on LinkNode R4 and select program via UART
- Connect DC power to LinkNode R4
- Connect USB TTL UART cable to UART port of LinkNode R4
- Connnect the other side to PC
- Check your serial port which your PC recognize
- Click the \*\*Upload\* on Arduino IDE
- After finished, jump out the S5 on LinkNode R4 and select boot from flash

#### 2. Remotely control LinkNode R4 with LinkSprite.IO platform

LinkSprite IO (https://github.com/YaoQ/pcduino-doc/blob/master/LinkNodeR4/www.linksprite.io) is an IoT platform which supports RESTful API and WebSocket. These make the mobile APP, website application or device connect it very easily. The following I will introduce is about how to use LinkNode R4 to communicate with LinkSprite IO platform.

a. Create a new account and device on LinkSprite.io

Go to <u>www.linksprite.io (http://www.linksprite.io/)</u> and sign up Enter your Email and password to create a new account Go to My Account to get your own API Key. The API Key is fatal because only add the Key in your codes, can the data sync to your IoTgo account.

×

My Profile       Email       test@gmail.com         My Devices       API Key       285316a1-873b-4a03-b5da-6a0072ee6554         Old Password       Old Password         New Password       New Password	My Account	My Profile	
My Devices       API Key       285316a1-873b-4a03-b5da-6a0072ee6554         Old Password       Old Password         New Password       New Password	My Profile	Email	test@gmail.com
Old Password Old Password New Password	My Devices	API Key	285316a1-873b-4a03-b5da-6a0072ee6554
New Password New Password		Old Password	Old Password
		New Password	New Password

#### Create New Device

Device Name	light					
Device Type	02 (Simple light)	T				
Group Name	LinkNode					
	Create	Close				
Click the created device icon nodeMCU	and get the DeviceID.					
	Light: OFF					
Light On Light Off						
		Î				
Action: ON OFF		Add Time				
Device ID	020000001					
Device Name	nodeMCU					
Group Name	LinkNode					
	Save Delete	Close				

b. Query the source code using your own apikey and device ID

Device API which the LinkSprite IO support is JSON-based, which means all request and response data is enclosed in JSON format. Currently it supports 3 kind of request.

- Update: Update device status to LinkSprite IO
- Query: Get device status from LinkSprite IO

This demo will send http POST request to query the param--light on linksprite.io, if the light is on, then turn one relay on, if off, then turn off the relay.

c. Install WiFi Manager library

To make it more convinent, we add WiFi manager library in this demo.

The ESP8266 WiFi Connection manager with web captive portal, this Arduino library can make configure AP's SSID and password via web page when you want LinkNode R4 to connect to AP.

- Open Arduino IDE and go to Sketch --> Include Library --> Manage Libraries
- Search the wifimanager and install it

d. Develope source code

Enter the following source code

```
- Configure your apikey and deviceID in the source code
```

<pre>#include <esp8266wifi.h> #include <usstring.h> //the library are needed for autoconfig WiFi #include <onsserver.h> #include <esp8266webserver.h> #include <unsserver.h> #include &lt;</unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></unsserver.h></esp8266webserver.h></onsserver.h></usstring.h></esp8266wifi.h></pre>	
<pre>wiriManager.setAPStatilPContig(IPAddress(10,0,1,1), IPAddress(10,0,1,1), IPAddress(255,255,255,0)); wiriManager.autoConnect("LinkNodeAP"); Serial.print("WiFi Connected\n"); Serial.println("WiFi connected"); </pre>	
<pre>void loop() {     if (client.connect(server,80)) {         String postStr =="{;             postStr +="\"action\":\"query\",";             postStr +="pikey\":\";             postStr += apikey;             postStr +="\"action\":\";             client.print("foot:\");             client.print("footnet-t-!spe: application/json\n");             client.print("content-tegth: ");             client.print(postStr);             client.print(postStr);             client.print(postStr);             postStr =="\";             while (client.actionlebe()) {             client.println("Store response");             String request =="\";             while (client.actionbel()) {             chart = client.read();             request +=c;             }             client.read();             request +=c;             }             for action = not post             tere:             tere:</pre>	
<pre>if (request!= NULL) {     int index1 = request.indexOf(":{");     int index2 = request.indexOf("},");     String param = request.substring(index1, index2 + 1);     Serial.print("The param is ");     Serial.println(param); }</pre>	
<pre>if(param.indexOf("off")&gt;0){     digitalWrite(12, LOW);     Serial.println("OFF"); } else if(param.indexOf("on")&gt;0){     digitalWrite(12, HIGH);     Serial.println("ON"); }</pre>	
<pre>client.stop(); Serial.println("Waiting"); delay(2000); } }</pre>	
2. Configure to accece WiFi AP	

- Take the step above to upload the program, the LinkNode R4 will create a AP called LinkNodeAP
- Use your mobile phone to connect this AP

<sup>•</sup> Open a browser and enter the ip address 10.0.1.1 and you will see the following website:



# LinkNodeAP

WiFiN	lanager
-------	---------

Configure WiFi

Configure WiFi (No Scan)

Info

Reset

- Click the button configure WiFi
- Select your WiFi AP which you want to connect and enter your wifi password.
- If connecting failed, you can go to the same website to configure it again.
- Also, you can use Serial Monitor in Arduino IDE to check the status.
- After that, LinkNode R4 will connect to Linksprite IO via the internet.

f. Test

Open the serial monitor to check the status

×

nodeMCU

Light On Light Off						
Action: ON OFF	(	Add Timer				
Device ID	020000001					
Device Name	nodeMCU					
Group Name	LinkNode					
	Save Delete Close					
<ul> <li>Open your light device which is create on linksprite.io</li> </ul>						

- Click the button ON and OFF
- Check the status of relay on LinkNode R4, is it following your control?

# Schematic

LinkNode R4 Schematic (https://s3.amazonaws.com/linksprite/LinkNode+R4/LinkSprite+R4+V1.0.pdf)

# Dimension

LinkNode R4: Arduino-compatible WiFi relay controller - LinkSprite Playgound



# 3D printing enclosure





Linknode R4 3D Builder a (https://s3.amazonaws.com/cutedigi/linknode+r4/LinkNode+R4a.STL)

Linknode R4 3D Builder b (https://s3.amazonaws.com/cutedigi/linknode+r4/LinkNode+R4b.stl)

# Video

How to use Android APP to control Linknode R4 (https://www.youtube.com/watch?v=Tj6FJDmlhXY&feature=youtu.be)

How to use IOS APP to control Linknode R4 (https://youtu.be/OO\_Q2HpZP5I)

Retrieved from "http://linksprite.com/wiki/index.php?title=LinkNode\_R4:\_Arduino-compatible\_WiFi\_relay\_controller&oldid=12475"

This page was last edited on 20 June 2017, at 05:52.