# Node Red Controlled Web LED on ESP32 with Raspberry Pi 4

#### Node Red Controlled Web LED on ESP32 with Raspberry Pi 4

Difficulté Moyen	Ourée 2 heure(s)	Catégories Électronique	Coût 70 USD (\$)
Commaira			
Sommane			
Introduction			
Étape 1 - Get PCBs For	Your Projects Manufactur	ed	
Étape 2 - Create a Devi	ce on Qubitro		
Étape 3 - Flash ESP32 v	vith Arduino IDE		
Étape 4 - Connect LED	to ESP32		
Étape 5 - Write Code fo	or ESP32		
Étape 6 - Create Node-	RED Flow		
Étape 7 - Conclusion			
Notes et références			
Commentaires			

#### Introduction

In this tutorial, you will learn how to use Node-RED, a visual programming tool for the Internet of Things (IoT), to control an LED on an ESP32 board with a Raspberry Pi as the MQTT broker. MQTT is a lightweight and simple messaging protocol that allows devices to communicate with each other over a network.

You will need the following components for this project:

- ESP32 development board
- USB cable to connect the ESP32 to your computer
- Raspberry Pi with Node-RED
- Computer with Arduino IDE and PubSubClient library installed

#### Matériaux

#### Étape 1 - Get PCBs For Your Projects Manufactured

You must check out PCBWAY for ordering PCBs online for cheap! You get 10 good-quality PCBs manufactured and shipped to your doorstep for cheap. You will also get a discount on shipping on your first order. Upload your Gerber files onto PCBWAY to get them manufactured with good quality and quick turnaround time. PCBWay now could provide a complete product solution, from design to enclosure production. Check out their online Gerber viewer function. With reward points, you can get free stuff from their gift shop

#### Outils



### Étape 2 - Create a Device on Qubitro

The first step is to create a device on the Qubitro platform. A device represents your physical device (Raspberry Pi) on the cloud. You need to create a device to obtain the MQTT credentials and topics for your Raspberry Pi.

To create a device on Qubitro, follow these steps:

- 1. Log in to your Qubitro account and create a new project
- 2. Then go to the Devices page, select MQTT as the communication protocol, and click Next.
- 3. Enter all the details.

4. Copy the Device ID, Device Token, Hostname, Port, Publish Topic, and Subscribe Topic. You will need these values later in the code. Click Finish.

#### You have successfully created a device on Qubitro. You can see your device on the Devices page.

					Manuel Development			
+ New Project	O Connect source		"VOC_Level": "NO", "ampere": 14412.4212.		New Project			
			"humidity": 79,					
Projects	Chirpstack		"lon": "26.76005",					
et monitoring	On open source Lo	Aalilan Network Server stack.	"voltage": 43, "Time": "26 July 2022 10:40:00			Create New Project		
at monitoring	The Trine	- Charles				Get started by creating a new project		
hargers	• A robust, f	exible LoRaWan network server.				Name		
						Description		
detection	Helium     A distributed nature	nik for LoBaltim-anabled InT devices	PROCESS DEVICE DATA			Height for teams or differentiating between onsiects with simil	ilar names	
fonitoring			Payload Formatter					
			Rule Engine					
ocumentation			Time Series Queries			Create		
redentials		× ×						
re-derrolaria	· · · · · · · · · · · · · · · · · · ·	AF 🖸 🕺	•					
lling		-	CREATE SOLUTIONS					
	()		Environmental Monitoring					
ipport			Asset Tracking					
count			New Startup Idea					
					Ballines Castadar -			
				Desires Series				
				Devices strongs				
					Browse Data Source	25		
					Integrate and standardize your device-data, and pow	er all your apps with a unified data platform.		
					Connectivity	Q Search source.		
					Featured			
					Featured			
					S)M		,t cor	net
					<i>∜</i> M	QTT 🖉 ChirpStack		net
					<i>∜</i> M	QTT ChirpStack	¢ ser	net
					ЯM	QTT ChirpStack	⊕ ser	net
					38 MO11	QTT ChirpStack	serret serret server serve	net
						ChirpStack	Senet Lutenterf Research Senee	net
						ChirpStack	Sent Lines form	net *
					Morr	Chirpstack	Sent Linester framework former	net 
					en e	CC Chirpstock	Original Sector	net 
					Image: Section of the section of t	ChirpStack	Series	net 
					€ SET Sector 21 Sector 21	CC ChirpStadd CC Chi	Sector Sector	net * *
						ChirpScace       C	C C C C C C C C C C C C C C C C C C C	net " text
Pologi (ga 📻 🖕 Juan	Menter Society -		t get kalant		events events	QCTC     C2 ChirpStack       main     C2 main	C C C C C C C C C C C C C C C C C C C	net 
Padaga Lago 📻 🛌 kanal p	Menal (Secular -		Particular ling (s	1 (B)		CC ChirpStadd CC Chi	Sectors Sectors	net
Padeg Ligo 🌄 🖕 Kont	Metaleo (Service -		Pandant Neg 1			CC ChirpSteel CC CC	Sector	net 
Padrog Ligo 👝 y Kost	You Are Almost Done		Andres ing S	1 - C C Research Control (Control (Contro) (Control (Contro) (Contro) (Control (Cont		CC ChirpStack  C	e ester e e	net 
Palega Lago 💽 🖉 . Maari	You Are Almost Done		Factors ing D	1.0 C		CCC ChirpStack	e e estat e estat	net " " "
Patroj las 💽 + Mari	You Are Almost Done		Testino Reg 10	1 (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)		CC ChirpSteed	e errer e e	net 
Aday ing 💽 + Kat	You Are Almost Done	New MQTT source	Particular sing in			CCC ChirpStack  CC Ch	<ul> <li>Constraint of the sector</li> <li>Constraint of the s</li></ul>	net
Participa ( 20 🖝 + Mari	You Are Almost Done	New WOTY Source	Feedback Heig Ch	1.0 C Total (C Total		CCC ChippStock Control	Sector	net
Particul Law Con - Market	You Are Almost Done	New MQTT source of energy and and and the source				CC ChirpStock	e estat e e	
Polarija 🕢 - Matri	Vou Are Almost Done	Net WOTT source The source of the source of		10 C		CCCharpstock       CCCharpstock       CCCharpstock       CCCCARPstock       CCCCCARPstock       CCCCARPstock	<ul> <li>Constraints</li> <li>Constraints</li> <li>Constraints</li> <li>Constraints</li> <li>Constraints</li> <li>Constraints</li> </ul>	
nytes (as 🕐 - two	Vou Are Almost Done • trans	Her MOTI Sacore Barrer and an an Million and Anna and Million and Anna and Million and Anna				CCC ChirpSteld	e e entre la compara de la compa	
	Vou Are Almost Done	Here MCTT Source The and a source and sourc				CCC Chirpstock CC Chirpstock C	texesplanet des	
	Vou Are Almost Done • trait • trait	New MOTI Suscee to more susceed and the susceed The more susceed and the sus				CC Chippeded C Chi	Sector	
	Normal control and a control a	NUM SUPT SAUCE           State Stat			Record Carlor (Control Control	CC ChippStack       State       State </td <td>e e e e e e e e e e e e e e e e e e e</td> <td></td>	e e e e e e e e e e e e e e e e e e e	
Adaptos 🗨 + tos	terrent level * - * trait *			10 C AND		CCC Chippstood CC Chippstood Martine Constraints Martine Constraints	<ul> <li>Control Control C</li></ul>	
Politic (see a second s	Vou Are Almost Done • trait • trait	New MOTT succe         Bate sources are solition and and and and and and and and and an				C: ChipSteld       Anital Cianality	es estatus	
Protection C - National Protec		New WOTT Facures           Marine Strange Str				CCC Chirpsteed and an	s of series and s	
Promptog 🗨 - Mari	Vou Are Almost Done	New MOTI Source           Maria Maria Maria Maria Maria Maria           New Moti Source				C: ChipStell       Marcine       Marcine<	s Constantination de la constantinatione de la constantinatione de la constantinatione d	

# Étape 3 - Flash ESP32 with Arduino IDE

The ESP32 is a powerful and versatile microcontroller that can run Arduino code. You will use the Arduino IDE to program the ESP32 and make it communicate with the MQTT broker using the PubSubClient library.

- To install the ESP32 board in Arduino IDE, you can follow the instructions in this tutorial or use the steps below:
  - Open the preferences window from the Arduino IDE: File > Preferences.
  - Go to the "Additional Board Manager URLs" field and enter the following URL: https://dl.espressif.com/dl/package\_esp32\_index.json.
  - Open Boards Manager (Tools > Board > Boards Manager), search for ESP32, and click the install button for the "ESP32 by Espressif Systems".
  - Select your ESP32 board from Tools > Board menu after installation.
  - Open the library manager from Sketch > Include Library > Manage Libraries.
  - Search for PubSubClient and click the install button for the "PubSubClient by Nick O'Leary".
  - Restart your Arduino IDE after installation.

Sketchbook location	:			
C:\Users\Pradeep\0	OneDrive\/	Documents\Arduino		Browse
Editor language:		System Default	(requires restart of Arduino)	
Editor font size: Interface scale:	Add	ditional Boards Manager URLs	×	
	Enter ad	ditional LIPLs, one for each row		
Theme:	ciliter du	didonal ORLS, one for each row		
Theme: Show verbose outp Compiler warnings	/espre /ardui	ssif/arduino-esp32/gh-pages/package_esp32 .no-pico/releases/download/global/package_	_index.json rp2040_index.json	
Theme: Show verbose out; Compiler warnings Display line nu	i/espre :/ardui	unonal vels, one for each row ssif/arduino-esp32/gh-pages/package_esp32 .no-pico/releases/download/global/package_	_index.json xp2040_index.json	
Theme: Show verbose outp Compiler warnings Display line nu Verify code aft Check for upda	/espre /ardui	anoma ouca, one to excit two ssif/arduino-esp32/gh-pages/package_esp32 no-pico/releases/download/global/package_ a list of unofficial boards support URLs	_index.json rp2040_index.json	
Theme: Show verbose outp Compiler warnings Display line nur Verify code aft Check for upda Use accessibilit	//espre //ardus	anoma ouca, one to excit two ssif/arduino-esp32/gh-pages/package_esp32 no-pico/releases/download/global/package_ a list of unofficial boards support URLs	_index.json rp2040_index.json	
Theme: Show verbose outp Compiler warnings Display line nur Verify code aft Check for upda Use accessibilit Additional Boards M	Click for	allolation CL2, bite for each row ssif/arduino-esp32/gh-pages/package_esp32 no-pico/releases/download/global/package_ a list of unofficial boards support URLs RLS: m/earlephilhower/arduino-pico/releases/download/	_index.; son _rp2040_index.; son OK Cancel global/package_rp2040_index.; sor	
Theme: Show verbose outp Compiler warnings Display line nu Verify code aft Check for upda Use accessibilit Additional Boards M More preferences cr	Click for	alistic duct, bie for each row saf/Arduino-esp32/gh-pages/package_esp32 no-pico/releases/download/global/package_ a list of unofficial boards support URLs R.s: [m/esrlephilhower/arduino-pico/releases/download/ el directly in the file	_index.;son rp2040_index.;son OK Cance global/package_rp2040_index.json	

ype All	esp32		
Arduino ESP32 Board	5		
by <b>Arduino</b> Boards included in thi Arduino Nano ESP32. <u>More Info</u>	is package:		
esn32			
by Espressif Systems Boards included in thi ESP32 Dev Board, ESI More Info	s version <b>2.0.9 INSTALLED</b> is package: P32-S2 Dev Board, ESP32-S3 Dev Board	l, ESP32-C3 Dev Board, Arduino Nano	ESP32.
Select version ~ Ir	nstall		Update Remove
	Downloa	ading platforms index	Cancel
aub http   Ard	Downlow	nding platforms index	Cencel
qub_http Ard	Downlos urino 1.8.19 -	ading platforms index	Cancel
qub_http   Ard	Downlow uino 1.8.19 - pols Help Auto Format Archive Sketch	- Ctrl+T	Cano
qub_http   Ard	uino 1.8.19 - ools Help Auto Format Archive Sketch Fix Encoding & Reload	- Ctrl+T	Cano
qub_http   Ard e Edit Sketch Tr qub_http include (#i	uino 1.8.19 ools Help Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries	- Ctrl+Shift+I	Cancel
qub_http   Ard e Edit Sketch To qub_http include <#i	uino 1.8.19 ools Help Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries Serial Monitor	- Ctrl+T Ctrl+Shift+I Ctrl+Shift+M	Cancel
qub_http   Ard e Edit Sketch To qub_http include <wi include <wi< td=""><td>Downlea urino 1.8.19 ools Help Auto Format Archive Sketch Fix Encoding &amp; Reload Manage Libraries Serial Monitor Serial Plotter</td><td>Ctrl+T Ctrl+Shift+I Ctrl+Shift+I Ctrl+Shift+L</td><td>Cencel</td></wi<></wi 	Downlea urino 1.8.19 ools Help Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries Serial Monitor Serial Plotter	Ctrl+T Ctrl+Shift+I Ctrl+Shift+I Ctrl+Shift+L	Cencel
qub_http   Ard e Edit Sketch To qub_http include <wi include <kt onst char* onst char*</kt </wi 	uino 1.8.19 ools Help Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries Serial Monitor Serial Plotter WiFi101 / WiFININA Firmwa	- Ctrl+Shift+I Ctrl+Shift+I Ctrl+Shift+L are Updater	Cancel
qub_http   Ard e Edit Sketch To qub_http include <wi include <wi onst char*</wi </wi 	bowner bols Help Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries Serial Plotter WiFi101 / WiFININA Firmwar Board: "Arduino Yún"	- Ctrl+T Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L xre Updater	Cance Boards Manager

# #include <#i</td> Manage Libraries... Ctrl + Shift + 1 #include <#i</td> Serial Monitor Ctrl + Shift + 1 serial Plotter Ctrl + Shift + 1 const char\* Serial Plotter Ctrl + Shift + 1 const char\* WiF101 / WiFININA Firmware Updater Board's Manager... //Your Domai Board's Arduino Yún\* Board's Manager... Port Set Board Info Arduino AVR Boards // the follo Programmer.\* AVRISP mkli\* ESP32 Arduino #manage low Bum Boottoader Raspberry Pi RP2040 Boards(3.3.0) // Timer set to 10 minutes (500000) Invalid library found in C:\Users\Pradeep\OneDrive\ Invalid library found in C:\Users\Pradeep\OneDrive\ Invalid library found in C:\Users\Pradeep\OneDrive\ 6 Arduino Yún on COM4 Invalid library found in C:\Users\Pradeep\OneDrive\

# Étape 4 - Connect LED to ESP32

The LED is a simple device that emits light when current flows through it. You will connect the LED to one of the GPIO pins of the ESP32 and control its state (on or off) with MQTT messages. In my case I'm going to use the onboard LED in the ESP32 Dev board.



### Étape 5 - Write Code for ESP32

The code for the ESP32 will do the following tasks:

- Connect to your Wi-Fi network
- Connect to the Qubitro MQTT broker on Raspberry Pi
- Receive messages from "output" and turn on or off the LED accordingly

<your_ssid></your_ssid>	<your_password></your_password>	You can copy and paste the c	code below into your Arduino IDE. Make sure to replace , , with your own
		<your_qubtro_credientials></your_qubtro_credientials>	values.
			<pre>void setup() {     pinMode(Relay1, OUTPUT);     Serial.begin(115200);     setup_wifi();     client.setServer(mqtt_server, 1883);     client.setCallback(callback); } void loop() {     if (!client.connected())     {         reconnect();         }         client. Loop();     } </pre>

After writing the code, upload it to your ESP32 board by selecting the right board and port from the Tools menu and clicking the upload button.



#### Étape 6 - Create Node-RED Flow

The Node-RED flow will do the following tasks:

- Connect to the MQTT broker on Raspberry Pi
- Subscribe to a topic named "output"

[

- Publish messages "true" or "false" to a topic named "output"
- Create a dashboard with a button and a text node

You can create the Node-RED flow by dragging and dropping nodes from the palette and connecting them with wires. You can also import the flow from this link or use the JSON code below:

```
{
   "id": "eb8f9c0d054be30c",
   "type": "tab",
   "label": "Flow 2",
   "disabled": false,
   "info": "",
   "env": []
},
{
   "id": "4ce6cd876fd5441f",
   "type": "mqtt out",
   "z": "eb8f9c0d054be30c",
   "name": "",
   "topic": "output",
   "qos": "",
   "retain": "",
   "respTopic": "",
   "contentType": "",
   "userProps": "",
   "correl": "",
   "expiry": ""
   "broker": "6d40b7b21c734b53",
   "x": 870,
   "y": 240,
   "wires": []
},
{
   "id": "974a7a8bb6db9bf9",
   "type": "mqtt in",
   "z": "eb8f9c0d054be30c",
   "name": "",
   "topic": "output",
   "qos": "2",
   "datatype": "auto-detect",
   "broker": "6d40b7b21c734b53",
   "nl": false,
   "rap": true,
   "rh": 0,
   "inputs": 0,
   "x": 670,
   "y": 320,
   "wires": [
     [
        "d0dc7378c7bfb03b",
        "f1219a2eeabe825f"
     ]
  ]
},
{
   "id": "d0dc7378c7bfb03b",
   "type": "debug",
   "z": "eb8f9c0d054be30c",
   "name": "debug 4",
   "active": true,
   "tosidebar": true,
   "console": false,
   "tostatus": false,
   "complete": "payload",
   "targetType": "msg",
   "statusVal": "",
   "statusType": "auto",
```

```
"x": 880,
   "y": 320,
   "wires": []
},
{
   "id": "6bd227b280e372b7",
   "type": "ui_switch",
   "z": "eb8f9c0d054be30c",
   "name": "",
   "label": "Light One",
   "tooltip": "",
   "group": "cd687a95.00e108",
   "order": 0,
   "width": 0,
   "height": 0,
   "passthru": true,
   "decouple": "false",
   "topic": "topic",
   "topicType": "msg",
   "style": "",
   "onvalue": "true",
   "onvalueType": "bool",
   "onicon": "",
   "oncolor": "",
   "offvalue": "false",
   "offvalueType": "bool",
   "officon": "",
   "offcolor": ""
   "animate": false,
   "x": 680,
   "y": 240,
   "wires": [
     [
        "4ce6cd876fd5441f"
     ]
  ]
},
{
   "id": "f1219a2eeabe825f",
   "type": "ui_text",
   "z": "eb8f9c0d054be30c",
   "group": "cd687a95.00e108",
   "order": 1,
   "width": "6",
   "height": "2",
   "name": "",
   "label": "Status : ",
   "format": "{{msg.payload}}",
   "layout": "row-center",
   "x": 1060,
   "y": 320,
   "wires": []
},
{
   "id": "6d40b7b21c734b53",
   "type": "mqtt-broker",
   "name": "Qubitro Downlink",
   "broker": "broker.qubitro.com",
   "port": "1883",
   "clientid": "",
   "autoConnect": true,
   "usetls": false,
   "protocolVersion": "4",
   "keepalive": "60",
   "cleansession": true,
   "autoUnsubscribe": true,
   "birthTopic": "r43MsJYzcVwZtUXVfZo6XD0Ym7CRegewPQXMt$ho",
   "birthQos": "0",
   "birthPayload": "",
   "birthMsg": {},
   "closeTopic": "",
   "closeQos": "0",
   "closePayload": "",
   "closeMsg": {},
```

```
"willTopic": "",
     "willQos": "0",
     "willPayload": "",
     "willMsg": {},
     "userProps": ""
     "sessionExpiry": ""
  },
  {
     "id": "cd687a95.00e108",
     "type": "ui_group",
     "name": "ESP32 Home Controller",
     "tab": "aa146f4d.b53ca",
     "order": 1,
     "disp": true,
     "width": "6",
     "collapse": false
  },
  {
     "id": "aa146f4d.b53ca",
     "type": "ui tab",
     "name": "Demo Lab",
     "icon": "dashboard",
     "order": 1,
     "disabled": false,
     "hidden": false
  }
]
```

The input switch will send "true" when it is on, and it will send "false" when it triggers off. Then click on the Qubitro uplink pallet and edit the property. Here you need to replace your connection details and credentials. Next, just deploy the flow. And navigate to the /ui of the node-red server. Here you can toggle the switch to turn the lead on and off. Also, open the serial monitor and check the node-red response.

# Étape 7 - Conclusion

In this tutorial, we have seen how to control the LED with Node-Red and MQTT Server.



#### Notes et références

Link was incorrect, so I have corrected it.