


Node-Red Telegram Bot with Temperature Logger Part-2

Will guide you to node-red integration with a telegram bot to send temperature data.

 Difficulté Facile

 Durée 2 heure(s)

 Catégories Électronique

 Coût 35 USD (\$)

Sommaire

Introduction

Étape 1 - Get PCBs For Your Projects Manufactured

Étape 2 - What You Need

Étape 3 - How It Works

Étape 4 - A little recap of the previous blog

Étape 5 - Telegram to Node-Red

Étape 6 - Conclusion

Commentaires

Introduction

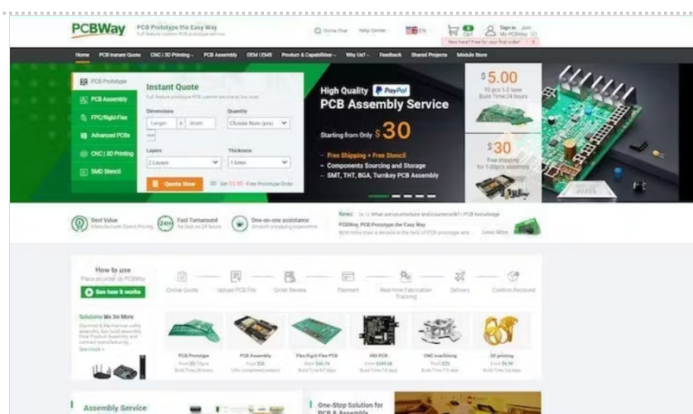
Telegram is a popular messaging app that offers end-to-end encryption, cloud-based storage, and a variety of features for communication. One of these features is the ability to create and interact with bots, which are automated programs that can perform tasks or provide information. In this blog post, I will show you how to use Telegram bots to control your home automation system with Node-RED, a visual programming tool that allows you to connect and orchestrate devices, services, and APIs.

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. Also, check out this useful blog on PCBWay Plugin for KiCad from here. Using this plugin, you can directly order PCBs in just one click after completing your design in KiCad.

Outils



Étape 2 - What You Need

To follow this tutorial, you will need:

- A Telegram account and a smartphone with the Telegram app installed.
- A Node-RED instance running on a device that can access your home automation system. You can use a Raspberry Pi, a computer, or a cloud service. For more information on how to install and run Node-RED, see here.
- The node-red-contrib-telegrambot node provides Telegram bot nodes for Node-RED. You can install it from the Node-RED palette manager or by running the following command in your Node-RED directory:

```
npm install node-red-contrib-telegrambot
```

- A Telegram bot token is a unique identifier for your bot. You can create a bot and get its token by talking to the @BotFather bot on Telegram. For more details.
- A chat ID is a unique identifier for the chat between you and your bot. You can get your chat ID by sending a message to your bot.

Étape 3 - How It Works

The basic idea is to use Node-RED to create a flow that receives messages from your Telegram bot, parses them to extract commands or queries, and then sends commands or responses back to your bot. The node-red-contrib-telegrambot node provides two main nodes for this purpose: the receiver node and the sender node.

The receiver node listens for messages from your bot and outputs a message object with the following properties:

msg.payload.chatId : The chat ID of the sender.

msg.payload.type : The type of the message, such as "message", "photo", "location", etc.

msg.payload.content : The content of the message, such as a text string, a file ID, or an object with additional data.

The sender node takes a message object as input and sends it to your bot. The message object should have the following properties:

msg.payload.chatId : The chat ID of the recipient.

msg.payload.type : The type of the message, such as "message", "photo", "location", etc.

msg.payload.content : The content of the message, such as a text string, a file ID, or an object with additional data.

You can use other nodes in between the receiver and sender nodes to process the messages and perform actions on your home automation system. For example, you can use a switch node to route messages based on their type or content, a function node to write custom logic in JavaScript, or an HTTP request node to call external APIs.

Étape 4 - A little recap of the previous blog

In the previous blog, we have seen how to use the telegram bot with node-red to trigger the temperature data. In this blog, we are going to see how to automate and trigger the Telegram bot to control the GPIO.

Étape 5 - Telegram to Node-Red

In the last article, we have seen how to trigger the sensor readings from Node-Red to Telegram, now let's see how to trigger and get the sensor data from Telegram Bot.

First, we need to use the telegram receiver node.

And configure the node with your bot credentials. Then add a debug block.

Next, just try to send some messages to your bot and look at the response in the debug console.

Now, you can see your texts are coming under payload.content. So, we are going to filter the content then based on the content we are going to trigger the actions.

Here are my complete blocks.

In this, I have tried to filter the contents as Green, Red, Blue, Off, and Env. Based on the contents we are going to trigger the neo pixels with certain colors.

Finally, just deploy the flow, open the telegram bot, and test out the connections.

Let's try out the neo pixels.

```
[
  {
    "id": "eb8f9c0d054be30c",
    "type": "tab",
    "label": "Flow 2",
    "disabled": false,
```

```

    "info": "",
    "env": []
  },
  {
    "id": "ea63aa67.c972f",
    "type": "template",
    "z": "eb8f9c0d054be30c",
    "name": "",
    "field": "payload",
    "fieldType": "msg",
    "format": "handlebars",
    "syntax": "mustache",
    "template": "{\n  \"chatId\": 5379039379,\n  \"type\": \"message\",\n  \"content\": \"Temperature : {{payload}}, Humidity : {{humidity}}\"\n}",
    "output": "json",
    "x": 660,
    "y": 360,
    "wires": [
      [
        "9e00d0a7.d5ccf",
        "600063bd96d765e6"
      ]
    ]
  },
  {
    "id": "9e00d0a7.d5ccf",
    "type": "debug",
    "z": "eb8f9c0d054be30c",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 830,
    "y": 300,
    "wires": []
  },
  {
    "id": "600063bd96d765e6",
    "type": "telegram sender",
    "z": "eb8f9c0d054be30c",
    "name": "roboerto_bot",
    "bot": "ae1a60539b8e5308",
    "haserroroutput": true,
    "outputs": 2,
    "x": 830,
    "y": 380,
    "wires": [
      [],
      []
    ]
  },
  {
    "id": "f2f9819ae972ae60",
    "type": "rpi-dht22",
    "z": "eb8f9c0d054be30c",
    "name": "",
    "topic": "rpi-dht22",
    "dht": "11",
    "pintype": 1,
    "pin": "7",
    "x": 500,
    "y": 380,
    "wires": [
      [
        "ea63aa67.c972f"
      ]
    ]
  },
  {
    "id": "e4b8f2dd860a7973",

```

```

"type": "telegram receiver",
"z": "eb8f9c0d054be30c",
"name": "",
"bot": "ae1a60539b8e5308",
"saveDataDir": "",
"filterCommands": false,
"x": 290,
"y": 200,
"wires": [
  [
    "ea58984a24ea493e"
  ],
  []
]
},
{
  "id": "ea58984a24ea493e",
  "type": "switch",
  "z": "eb8f9c0d054be30c",
  "name": "",
  "property": "payload.content",
  "propertyType": "msg",
  "rules": [
    {
      "t": "eq",
      "v": "Green",
      "vt": "str"
    },
    {
      "t": "eq",
      "v": "Red",
      "vt": "str"
    },
    {
      "t": "eq",
      "v": "Off",
      "vt": "str"
    },
    {
      "t": "eq",
      "v": "Blue",
      "vt": "str"
    },
    {
      "t": "eq",
      "v": "Env",
      "vt": "str"
    },
    {
      "t": "eq",
      "v": "Rainbow",
      "vt": "str"
    }
  ],
  "checkall": "true",
  "repair": false,
  "outputs": 6,
  "x": 450,
  "y": 140,
  "wires": [
    [
      "b410ed94340bb528"
    ],
    [
      "13aa73e421c429c7"
    ],
    [
      "4ed36e19be451327"
    ],
    [
      "db816a1f10392614"
    ],
    [
      "f2f9819ae972ae60"
    ]
  ]
}

```

```

    ],
    []
  ]
},
{
  "id": "b0edb6e1c2807920",
  "type": "rpi-neopixels",
  "z": "eb8f9c0d054be30c",
  "name": "Neo Pixel",
  "gpio": "18",
  "pixels": "8",
  "bgnd": "",
  "fgnd": "",
  "wipe": "60",
  "mode": "pixels",
  "rgb": "rgb",
  "brightness": "100",
  "gamma": true,
  "x": 660,
  "y": 100,
  "wires": []
},
{
  "id": "13aa73e421c429c7",
  "type": "function",
  "z": "eb8f9c0d054be30c",
  "name": "Red Led",
  "func": "\nmsg.payload = \"255,0,0\"\nreturn msg;\n\n\n",
  "outputs": 1,
  "timeout": "",
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 660,
  "y": 200,
  "wires": [
    [
      "b0edb6e1c2807920",
      "0d73aa1efdf96848"
    ]
  ]
},
{
  "id": "b410ed94340bb528",
  "type": "function",
  "z": "eb8f9c0d054be30c",
  "name": "Green Led",
  "func": "\nmsg.payload = \"0,255,0\"\nreturn msg;\n\n\n",
  "outputs": 1,
  "timeout": "",
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 670,
  "y": 160,
  "wires": [
    [
      "b0edb6e1c2807920",
      "c173db3cc3868fe8"
    ]
  ]
},
{
  "id": "4ed36e19be451327",
  "type": "function",
  "z": "eb8f9c0d054be30c",
  "name": "Off",
  "func": "\nmsg.payload = \"0,0,0\"\nreturn msg;\n\n\n",
  "outputs": 1,
  "timeout": "",
  "noerr": 0,
  "initialize": "",

```

```

"finalize": "",
"libs": [],
"x": 650,
"y": 240,
"wires": [
  [
    "b0edb6e1c2807920",
    "02e43d4dd4523371"
  ]
]
},
{
  "id": "db816a1f10392614",
  "type": "function",
  "z": "eb8f9c0d054be30c",
  "name": "Blue Led",
  "func": "\nmsg.payload = \'0,0,255\'\nreturn msg;\n\n",
  "outputs": 1,
  "timeout": "",
  "noerr": 0,
  "initialize": "",
  "finalize": "",
  "libs": [],
  "x": 660,
  "y": 280,
  "wires": [
    [
      "b0edb6e1c2807920",
      "15dcd2ff9b951c47"
    ]
  ]
},
{
  "id": "c173db3cc3868fe8",
  "type": "template",
  "z": "eb8f9c0d054be30c",
  "name": "",
  "field": "payload",
  "fieldType": "msg",
  "format": "handlebars",
  "syntax": "mustache",
  "template": "{\`chatId\`: 5379039379,\n\`type\`: \'message\',\n\`content\`: \'Green LED Triggered\`}",
  "output": "json",
  "x": 860,
  "y": 80,
  "wires": [
    [
      "62f3360544d9ca11",
      "4abe16519bcdcac"
    ]
  ]
},
{
  "id": "62f3360544d9ca11",
  "type": "debug",
  "z": "eb8f9c0d054be30c",
  "name": "",
  "active": true,
  "tosidebar": true,
  "console": false,
  "tostatus": false,
  "complete": "payload",
  "targetType": "msg",
  "statusVal": "",
  "statusType": "auto",
  "x": 1030,
  "y": 80,
  "wires": []
},
{
  "id": "4abe16519bcdcac",
  "type": "telegram sender",
  "z": "eb8f9c0d054be30c",
  "name": "roboerto bot".
}

```

```

    "bot": "ae1a60539b8e5308",
    "haserroroutput": true,
    "outputs": 2,
    "x": 1030,
    "y": 120,
    "wires": [
      [],
      []
    ]
  },
  {
    "id": "0d73aa1efdf96848",
    "type": "template",
    "z": "eb8f9c0d054be30c",
    "name": "",
    "field": "payload",
    "fieldType": "msg",
    "format": "handlebars",
    "syntax": "mustache",
    "template": "{\n  \"chatId\": 5379039379,\n  \"type\": \"message\",\n  \"content\": \"Red LED Triggered\"}",
    "output": "json",
    "x": 860,
    "y": 120,
    "wires": [
      [
        "4795fb81ce5c47a6",
        "12c1a78b02583d94"
      ]
    ]
  },
  {
    "id": "4795fb81ce5c47a6",
    "type": "debug",
    "z": "eb8f9c0d054be30c",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 1030,
    "y": 180,
    "wires": []
  },
  {
    "id": "12c1a78b02583d94",
    "type": "telegram sender",
    "z": "eb8f9c0d054be30c",
    "name": "roboerto_bot",
    "bot": "ae1a60539b8e5308",
    "haserroroutput": true,
    "outputs": 2,
    "x": 1030,
    "y": 220,
    "wires": [
      [],
      []
    ]
  },
  {
    "id": "15dcd2ff9b951c47",
    "type": "template",
    "z": "eb8f9c0d054be30c",
    "name": "",
    "field": "payload",
    "fieldType": "msg",
    "format": "handlebars",
    "syntax": "mustache",
    "template": "{\n  \"chatId\": 5379039379,\n  \"type\": \"message\",\n  \"content\": \"Blue LED Triggered\"}",
    "output": "json",
    "x": 860
  }

```

```

    "y": 160,
    "wires": [
      [
        "751fda5416547778",
        "46ea1f39f11d1b65"
      ]
    ]
  },
  {
    "id": "751fda5416547778",
    "type": "debug",
    "z": "eb8f9c0d054be30c",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 1030,
    "y": 280,
    "wires": []
  },
  {
    "id": "46ea1f39f11d1b65",
    "type": "telegram sender",
    "z": "eb8f9c0d054be30c",
    "name": "roberto_bot",
    "bot": "ae1a60539b8e5308",
    "haserroroutput": true,
    "outputs": 2,
    "x": 1030,
    "y": 320,
    "wires": [
      [],
      []
    ]
  },
  {
    "id": "02e43d4dd4523371",
    "type": "template",
    "z": "eb8f9c0d054be30c",
    "name": "",
    "field": "payload",
    "fieldType": "msg",
    "format": "handlebars",
    "syntax": "mustache",
    "template": "{\n  \"chatId\": 5379039379,\n  \"type\": \"message\",\n  \"content\": \"LED Off!\"",
    "output": "json",
    "x": 860,
    "y": 220,
    "wires": [
      [
        "e9aaa495e9c42799",
        "0730f1a37c585783"
      ]
    ]
  },
  {
    "id": "e9aaa495e9c42799",
    "type": "debug",
    "z": "eb8f9c0d054be30c",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 1030
  }

```



```

    ^ . 1000,
    "y": 380,
    "wires": []
  },
  {
    "id": "0730f1a37c585783",
    "type": "telegram sender",
    "z": "eb8f9c0d054be30c",
    "name": "roboerto_bot",
    "bot": "ae1a60539b8e5308",
    "haserroroutput": true,
    "outputs": 2,
    "x": 1030,
    "y": 420,
    "wires": [
      [],
      []
    ]
  },
  {
    "id": "e30abb1d7e9afcb0",
    "type": "inject",
    "z": "eb8f9c0d054be30c",
    "name": "Green",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "",
    "crontab": "",
    "once": false,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "x": 290,
    "y": 280,
    "wires": [
      [
        "b410ed94340bb528"
      ]
    ]
  },
  {
    "id": "4a2a01d9556ad50f",
    "type": "inject",
    "z": "eb8f9c0d054be30c",
    "name": "Blue",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "",
    "crontab": "",
    "once": false,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "x": 290,
    "y": 360,
    "wires": [
      [
        "db816a1f10392614"
      ]
    ],

```

```

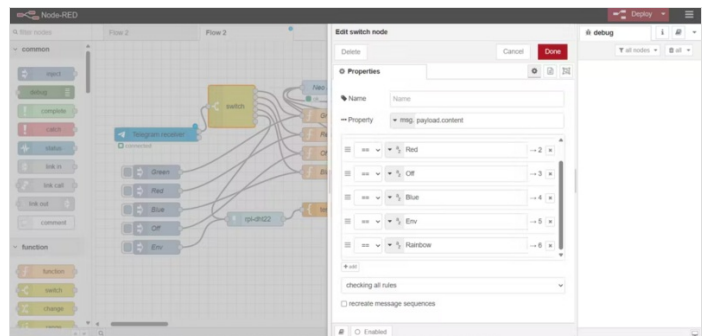
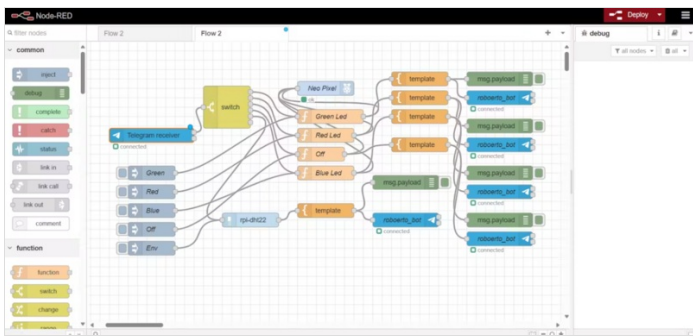
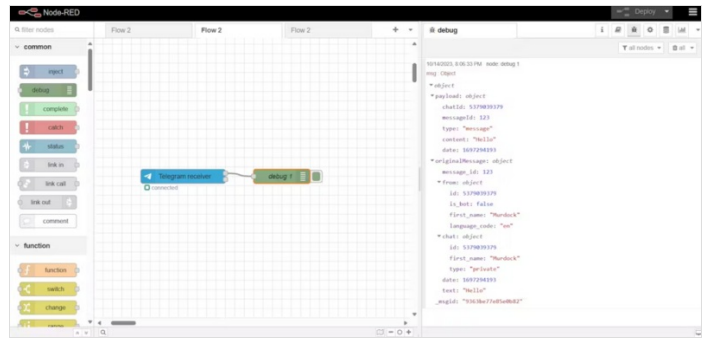
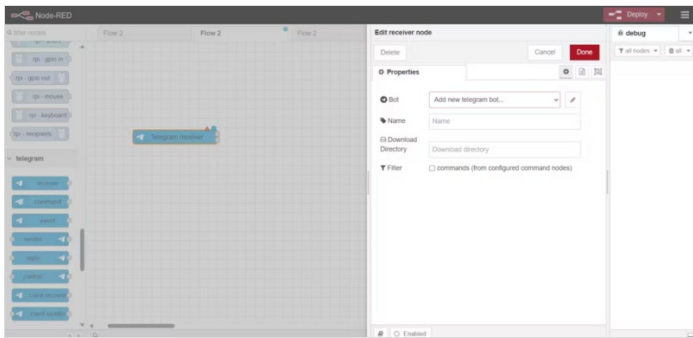
    ]
  ],
  {
    "id": "3f3ce5ea96251201",
    "type": "inject",
    "z": "eb8f9c0d054be30c",
    "name": "Red",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "",
    "crontab": "",
    "once": false,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "x": 290,
    "y": 320,
    "wires": [
      [
        "13aa73e421c429c7"
      ]
    ]
  },
  {
    "id": "3f2c1cebdde09f03",
    "type": "inject",
    "z": "eb8f9c0d054be30c",
    "name": "Off",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ],
    "repeat": "",
    "crontab": "",
    "once": false,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "x": 290,
    "y": 400,
    "wires": [
      [
        "4ed36e19be451327"
      ]
    ]
  },
  {
    "id": "26b87a7af5f875cc",
    "type": "inject",
    "z": "eb8f9c0d054be30c",
    "name": "Env",
    "props": [
      {
        "p": "payload"
      },
      {
        "p": "topic",
        "vt": "str"
      }
    ]
  }

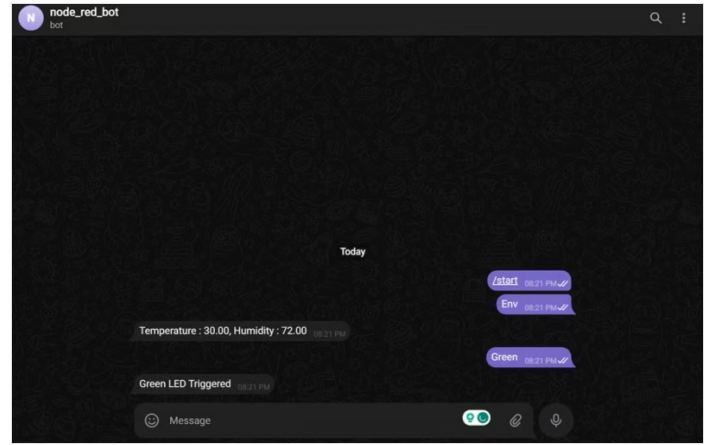
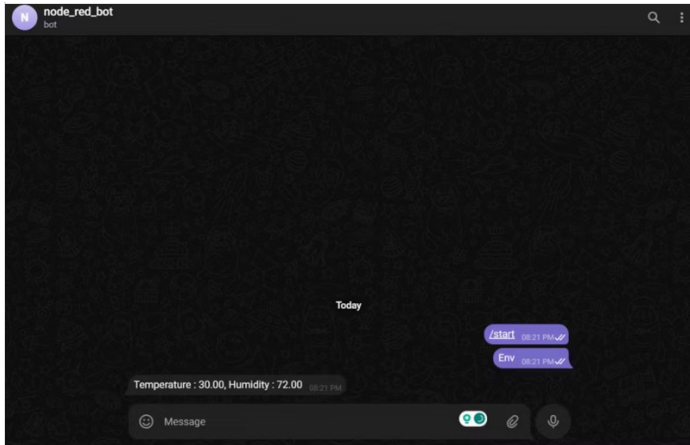
```

```

    ],
    "repeat": "",
    "crontab": "",
    "once": false,
    "onceDelay": 0.1,
    "topic": "",
    "payload": "",
    "payloadType": "date",
    "x": 290,
    "y": 440,
    "wires": [
      [
        "f2f9819ae972ae60"
      ]
    ]
  },
  {
    "id": "ae1a60539b8e5308",
    "type": "telegram bot",
    "botname": "roberto_bot",
    "username": "",
    "chatids": "",
    "baseapiurl": "",
    "updatemode": "polling",
    "pollinterval": "300",
    "usesocks": false,
    "sockshost": "",
    "socksprotocol": "socks5",
    "socksport": "6667",
    "socksusername": "anonymous",
    "sockspassword": "",
    "bothost": "",
    "botpath": "",
    "localbotport": "8443",
    "publicbotport": "8443",
    "privatekey": "",
    "certificate": "",
    "useselfsignedcertificate": false,
    "sslterminated": false,
    "verboselogging": false
  }
]

```





Étape 6 - Conclusion

In this blog post, I showed you how to use Telegram bots to control your home automation system with Node-RED. You can use this method to create your own custom interface for your smart home devices and services. You can also extend the functionality of your bot by adding more nodes and logic to your flow. For example, you can use the command node to create custom commands with parameters, the callback query node to handle inline buttons, or the answer inline query node to provide suggestions. You can also use other nodes from the node-red-contrib-telegrambot package or other Node-RED packages to integrate with more Telegram features or other platforms.

I hope you enjoyed this tutorial and learned something new. If you have any questions or feedback, please leave a comment below. Happy coding!

