

# Programmation Chauffage avec le noeud "Thermostat-Dynamique"

node-red-contrib-dynamic-thermostat 0.0.10



## Thermostat dynamique

```
npm install node-red-contrib-dynamic-thermostat
```

### Intégration continue Node.js

Un module Node-RED, un thermostat dynamique, vous permet de définir une température souhaitée (msg.payloadhystérésis) msg.payload et la température actuelle msg.payload pour activer ou désactiver le chauffage. Il est donc très dynamique et vous n'avez pas besoin de paramétrer le nœud lui-même.

### Installation

Ce nœud nécessite Node 10.x+.

- \$ cd ~/.node-red
- \$ npm install node-red-contrib-dynamic-thermostat

node-red-contrib-dynamic-thermostat 0.0.10

Thermostat dynamique

```
npm install node-red-contrib-dynamic-thermostat Thermostat dynamique
```

Intégration continue Node.js

Un module Node-RED, un thermostat dynamique, vous permet de définir une température souhaitée (msg.payloadhystérésis) msg.payload et la température actuelle msg.payload pour activer ou désactiver le chauffage. Il est donc très dynamique et vous n'avez pas besoin de paramétrer le nœud lui-même. Installation

Ce nœud nécessite Node 10.x+.

```
$ cd ~/.node-red $ npm install node-red-contrib-dynamic-thermostat
```

## Configurations

Le nœud possède une entrée (4 topic-payloads) et une sortie, (object with 4 payloads) comme décrit ci-dessous. La sortie contient onoff une valeur booléenne indiquant si le chauffage doit être allumé ou éteint. Si l'hystérésis est active, la valeur onoff est null :

## Saisir

Vous devez définir 4 charges utiles pour la fonction de succès :

- topic : switch, charge utile : vrai/faux/auto, vous pouvez forcer la mise en marche ou l'arrêt du thermostat, "auto" pour revenir au mode automatique
- topic : target, charge utile : température cible, par exemple 23
- topic : current, charge utile : température actuelle, par exemple 19 (provenant de votre thermomètre)
- topic : hysteresis, charge utile : température cible, par exemple 0.3

## Sortir

Le résultat est un objet dans la charge utile, comme suit :

[exem1.json](#)

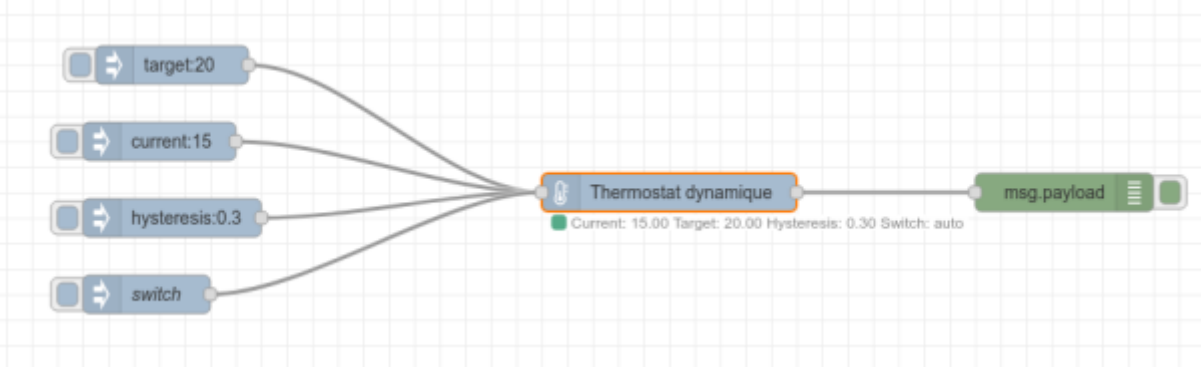
```
{
  switch: true | false,
  onoff: true | false | null,
  current: 19,
  target: 23,
  hysteresis: 0.3
}
```

## Exemple de flux

[exem2.json](#)

```
[{"id":"7c89a60c.c53b78","type":"tab","label":"Flow 1","disabled":false,"info":""},{ "id":"3bca5a6f.b51676","type":"inject", "z":"7c89a60c.c53b78","name":"","props":[{"p":"payload"}, {"p":"topic", "vt":"str"}], "repeat":"","crontab":"","once":false, "onceDelay":0.1, "topic":"target", "payload":"20", "payloadType":"str", "x":230, "y":180, "wires":
```

```
[["7382ac40.df0f54"]]}, {"id": "6717a2da.fcd2ec", "type": "debug", "z": "7c89a60c.c53b78", "name": "", "active": true, "tosidebar": true, "console": false, "tostatus": false, "complete": "false", "statusVal": "", "statusType": "auto", "x": 670, "y": 220, "wires": []}, {"id": "485cee9b.051ff", "type": "inject", "z": "7c89a60c.c53b78", "name": "", "props": [{"p": "payload"}, {"p": "topic", "vt": "str"}], "repeat": "", "crontab": "", "once": false, "onceDelay": 0.1, "topic": "current", "payload": "21", "payloadType": "str", "x": 220, "y": 240, "wires": [{"7382ac40.df0f54"}]}, {"id": "9ffdf3df.750a7", "type": "inject", "z": "7c89a60c.c53b78", "name": "", "props": [{"p": "payload"}, {"p": "topic", "vt": "str"}], "repeat": "", "crontab": "", "once": false, "onceDelay": 0.1, "topic": "hysteresis", "payload": "0.3", "payloadType": "str", "x": 230, "y": 300, "wires": [{"7382ac40.df0f54"}]}, {"id": "7382ac40.df0f54", "type": "dynamic thermostat", "z": "7c89a60c.c53b78", "name": "", "x": 460, "y": 220, "wires": [{"6717a2da.fcd2ec"}]}
```



From: <https://www.fablab37110.chanterie37.fr/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link: <https://www.fablab37110.chanterie37.fr/doku.php?id=start:nodered:thermostatdyn&rev=1766096128>

Last update: 2025/12/18 23:15

