

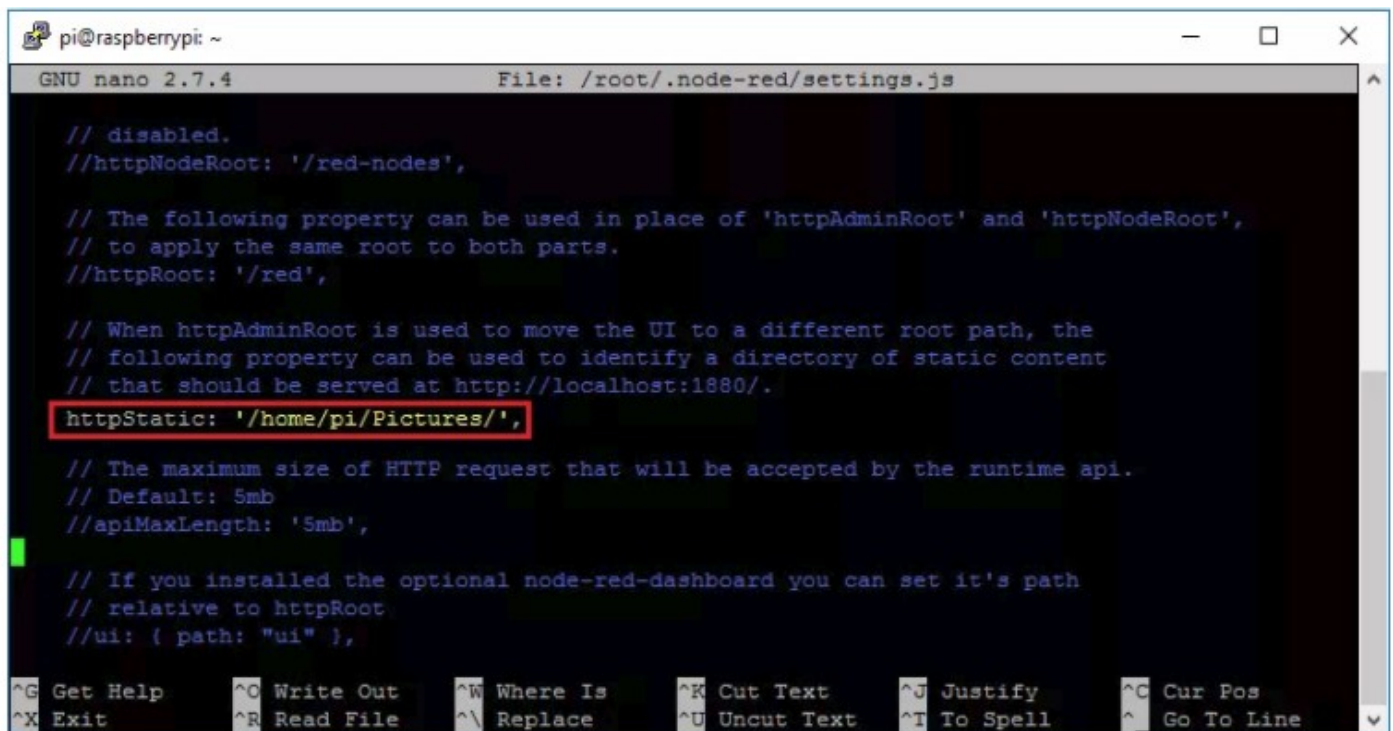
Node-RED : Raspberry Pi Camera (prendre une photo)

<https://randomnerdtutorials.com/node-red-with-raspberry-pi-camera-take-photos/>

1. Installer physiquement la caméra sur le connecteur . (côté bleu vers connecteur RJ45)
2. dans raspi-config Enabled la Caméra .
3. Il faut installer dans Manage Palette le module node-red-contrib-camerapi
4. Il faut choisir le repertoire pour l'enregistrement de la photo :

pi@raspberrypi:~ \$ **sudo nano ~/.node-red/settings.js**

dans **setting.js**, il faut localiser : httpStatic : '/home/pi/Pictures/'



```
pi@raspberrypi: ~
GNU nano 2.7.4 File: /root/.node-red/settings.js

// disabled.
//httpNodeRoot: '/red-nodes',

// The following property can be used in place of 'httpAdminRoot' and 'httpNodeRoot',
// to apply the same root to both parts.
//httpRoot: '/red',

// When httpAdminRoot is used to move the UI to a different root path, the
// following property can be used to identify a directory of static content
// that should be served at http://localhost:1880/.
httpStatic: '/home/pi/Pictures/',

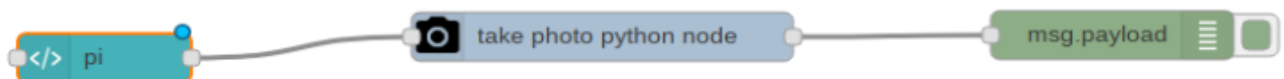
// The maximum size of HTTP request that will be accepted by the runtime api.
// Default: 5mb
//apiMaxLength: '5mb',

// If you installed the optional node-red-dashboard you can set it's path
// relative to httpRoot
//ui: { path: "ui" },

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Maintenant on peut se connecter à Node-RED (exemple : ip : 192.168.0.101 :1880)

Création du Dashboard



On dessine 3 nodes à l'intérieur d'un flow : **template**, **camerapi** **takephoto** et **debug**

On édite le **template node** :

node properties

Template type

Widget in group

Group

Camera raspberry [Camera]

Size

13 x 18

Name

pi

☒ Pass through messages from input.

☒ Add output messages to stored state.

Template

```
i 1 <script>
2 var value = "1";
3 // or overwrite value in your callback function ...
4 this.scope.action = function() { return value; }
5
6 function updateF() {
7   var source = '/photo1.JPEG',
8   timestamp = (new Date()).getTime(),
9   newUrl = source + '?_=' + timestamp;
10  document.getElementById("photo").src = newUrl;
11 }
12 </script>
13
14 <md-button ng-click="send({payload:action()})" onclick="setTimeout(updateF, 2500);" style="padding:20px; margin-bottom: 20px;" >
15   <ui-icon icon="camera"></ui-icon>
16   Take a photo<br>
17 </md-button>
18
19 <div style="margin-bottom:20px;">
20   
21 </div>
```

Puis on édite le **Camera Takephoto node** :

Edit camerapi-takephoto node

Delete

Cancel

Done

node properties

File Mode

Filemode

File Name

photo1.JPEG

File default path

Yes

File Format

JPEG

640x480

Rotation

0

Flip Picture

No

No

Brightness

50

Contrast

0

Sharpness

0

Image Effect

none

Name

Take photo node

Script du node Template pour copier/coller :

```
<script>

var value = "1";

// or overwrite value in your callback function ...

this.scope.action = function() { return value; }


function updateF() {

  var source = '/photo1.JPEG',

  timestamp = (new Date()).getTime(),

  newUrl = source + '?_=' + timestamp;

  document.getElementById("photo").src = newUrl;

}

</script>


<md-button ng-click="send({payload:action()})" onclick="setTimeout(updateF, 2500);"
style="padding:20px; margin-bottom: 20px;" >

  <ui-icon icon="camera"></ui-icon>

  Take a photo<br>

</md-button>


<div style="margin-bottom:20px;">

</div>
```


On modifie l'option **Width** dans le groupe crée :

Edit dashboard group node


Delete

Cancel


Update


 Name

Camera raspberry

 Tab

Camera



 Width

13

☒ Display group name