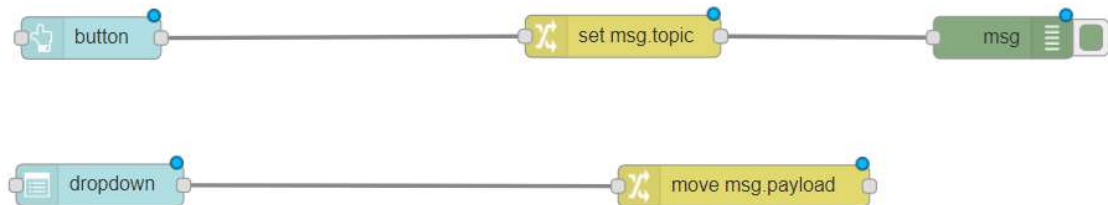


Sélection d'imprimante via menu déroulant

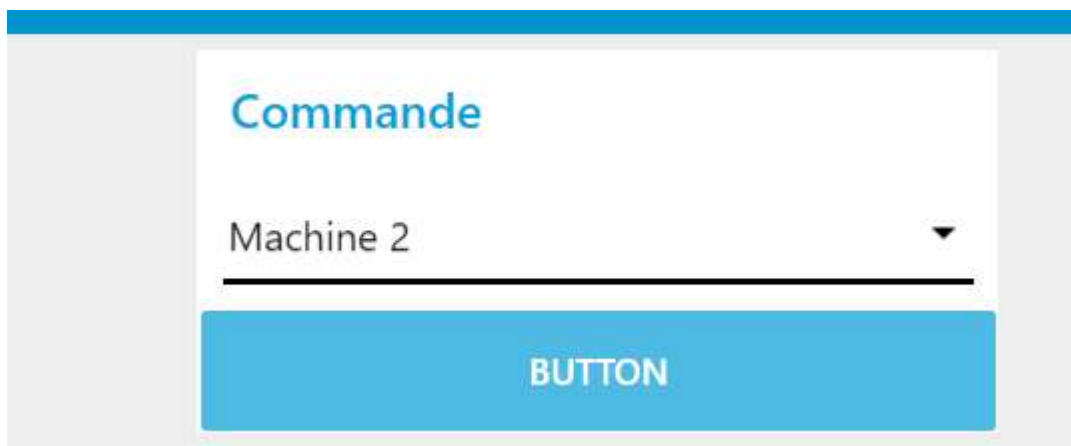
Pour piloter différentes imprimantes sur la même page (et non pas en dupliquant comme nous le faisons dans la procédure ci-joint), sur la demande de notre enseignant, nous avons réalisé un menu déroulant pour sélectionner la machine que nous souhaitons piloter. Ce menu déroulant change le topic sur lequel nous postons le message en MQTT. Chaque imprimante a donc son propre topic associé.

Importez le flow associé que vous trouverez ci-joint dans Node-RED IBM (se référencer à la procédure **Duplicate_IBM** où tout est expliqué sur la manière d'importer un flow).

Voici le flow en question :



Ainsi que l'interface graphique :



Explication du flow :

Dropdown : Nous associons du texte qui sera visible sur l'interface avec un topic

Edit dropdown node

Delete

Cancel

Done

▼ node properties

Group

Commande [Surveillance imprimante C ▼]

Size

auto

Label

optional label

Placeholder

Select option

Options

▼ a_z printcam/s

Machine 1

×

▼ a_z printcam/s

Machine 2

×

▼ a_z printcam/s

Machine 3

×

+ option

→ If msg arrives on input, pass through to output: ☒

> node settings

Page 2 sur 6

Change : Deux Nodes communiquent entre eux en envoyant un « msg.payload ». Avec le Node « change », nous allons déplacer le msg.payload (ici le topic) entré par l'utilisateur via le menu déroulant en **flow.topic**. Ceci permet d'accéder depuis n'importe quel autre Node à ce topic

Edit change node

Delete

Cancel

Done

node properties

Name

Name

Rules

Move

▼

▼

msg. payload

to

▼

flow. topic

×

Page 3 sur 6

Button : On paramètre ici l'esthétique du bouton (couleur, icône, taille et encore le texte associé). Pour cet exemple, nous ne nous attarderons pas sur ce point, c'est pour cela que les champs sont vides. Lors d'un appui sur le bouton, nous envoyons un message « oui ». Le topic est laissé vide car il est paramétré via le menu déroulant.

▼ node properties

Group

Commande [Surveillance imprimante GI

▼

Size

auto

Icon

optional icon

Label

button

Colour

optional text/icon color

Background

optional background color


When clicked, send:

Payload

▼ a_z

oui

Topic

➔ If `msg` arrives on input, pass through to output: 

Page 4 sur 6

Change : On récupère la variable stockée dans « flow.topic » et on met à jour le topic du bouton de cette manière. En appuyant sur le bouton, on recevra un msg.payload (paramétré dans le bouton) et un msg.topic qui correspondra au topic paramétré dans le menu déroulant.

Edit change node

Delete Cancel Done

▼ node properties

Name

Rules

Set ▼ msg. topic

to ▼ flow. topic

A l'aide du Node **Debug**, nous pouvons voir le changement de topic

06/04/2018 à 09:55:48 node: dd65b6ee.52f528

printcam/servo2 : msg : Object

```
▶ { payload: "oui", topic:
  "printcam/servo2", socketid:
  "5bQYhUY6Umo11ycZAAAA", _msgid:
  "d9cd88c0.49f0f8" }
```

06/04/2018 à 10:12:30 node: dd65b6ee.52f528

printcam/servo : msg : Object

```
▶ { payload: "oui", topic:
  "printcam/servo", socketid:
  "5bQYhUY6Umo11ycZAAAA", _msgid:
  "2fb320ba.08a35" }
```

Si nous souhaitons l'envoyer par MQTT, rajouter un bloc MQTT OUTPUT à la place du Node Debug.

Conclusion :

Dans notre projet, nous utilisons plusieurs topics par imprimante (ex : un topic pour les commandes, un pour l'envoi de l'image, un pour l'envoi de la température la CPU), cette méthode n'est donc pas la plus appropriée pour ce cas, car il faudrait plusieurs menus déroulants pour une imprimante. Préférez la méthode de duplication décrite ci-joint.