

Node red – ouverture d'un coffre

1 Le coffre

Le couvercle est bloqué par un électroaimant : Electroaimant Grove 101020073

<https://wiki.seeedstudio.com/Grove-Electromagnet/>



Un dessin des pièces d'une boîte peut être obtenu rapidement sur certains sites. Exemple

<https://www.festi.info/boxes.py/IntegratedHingeBox?language=fr>

Procédure à suivre :

- Choisir les dimensions
- matériaux en médium de 3mm
- Sauvegarder en .DXF
- Utiliser LibreCad pour ajouter un perçage

ou un logo

- Couche blanche : découpe
- Couche rouge : écriture
- Découpe par laser



2 Commande d'un électroaimant par la liaison série

Téléverser ce code dans la carte Arduino

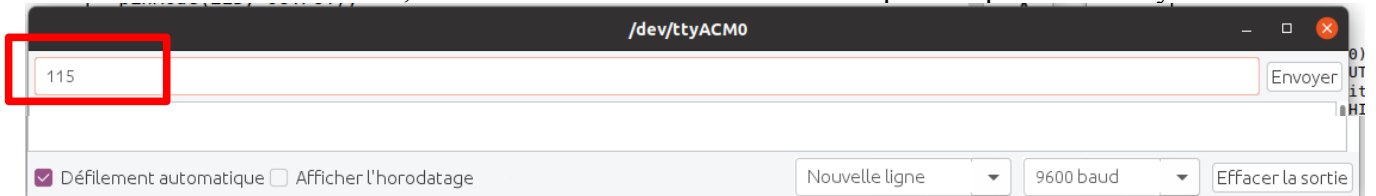
```
//EA => electroAimant
#define EA 8

String recu;

void setup() {
  Serial.begin(9600);
  pinMode(EA, OUTPUT);
  //Pour que la boîte soit fermée à la mise sous tension
  digitalWrite(EA,HIGH);
}

void loop() {
  //lecture port série
  if(Serial.available()>0)
  {
    recu = Serial.readStringUntil('\n');
    // Code bon, ouvrir la boîte
    if(recu=="115") digitalWrite(EA,LOW);
    else
    // Mauvais code, fermer la boîte
    if(recu!="115") digitalWrite(EA,HIGH);
  }
}
```

Dans le moniteur série, écrire le bon code ou une code faux puis cliquer sur Envoyer.

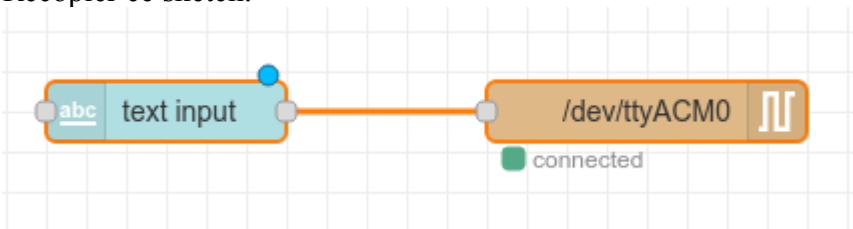


3 Utilisation de node red

Dans manage palette, vérifier que la librairie dashboard est installée.



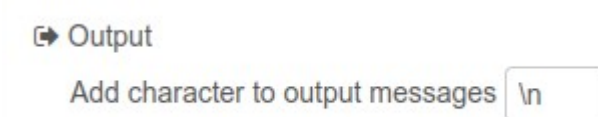
Recopier ce sketch.



Après Deploy, si le point n'est pas vert et la liaison série non connectée, vérifier que la configuration (nom du port et vitesse) est la même que dans l'IDE Arduino



Dans la config du port série, rajouter ceci

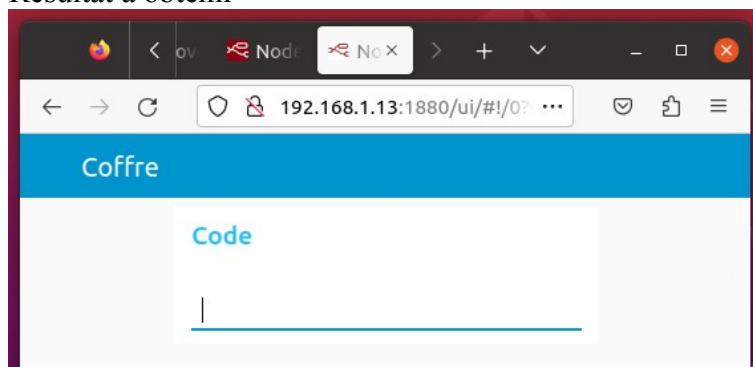


A droite, dans le menu _____ puis dashboard, ajouter une page « coffre » et dedans un groupe « Code ».



Deployer

Résultat à obtenir



4 Configurations

Dans le texte input, delay à 0 attend la touche entrée au lieu d'envoyer toutes les x ms.

The screenshot shows the 'Edit text input node' configuration window. At the top, there are 'Delete', 'Cancel', and 'Done' buttons. Below is a 'Properties' section with various settings:

- Group:** [Coffre] Code
- Size:** auto
- Label:** (empty text box)
- Tooltip:** optional tooltip
- Mode:** text input
- Delay (ms):** 0

Below the properties, there are several checkboxes and a dropdown menu:

- If msg arrives on input, pass through to output:
- Send value on focus leave:
- When changed, send:
 - Payload:** Current value
 - Topic:** msg: topic
- Class:** Optional CSS class name(s) for widget
- Name:** (empty text box)

A yellow tip box at the bottom states: "Setting Delay to 0 waits for Enter or Tab key, to send input."

Dans le serial, ajouter un \n à la fin des messages permet d'utiliser cette ligne du code Arduino « `recu = Serial.readStringUntil('\n');` »

The screenshot shows the 'Edit serial out node' configuration window. At the top, there are 'Delete', 'Cancel', and 'Update' buttons. Below is a 'Properties' section with various settings:

- Serial Port:** /dev/ttyACM0
- Settings:**
 - Baud Rate:** 9600
 - Data Bits:** 8
 - Parity:** None
 - Stop Bits:** 1
 - DTR:** auto
 - RTS:** auto
 - CTS:** auto
 - DSR:** auto
- Input:**
 - Optionally wait for a start character of (empty text box), then
 - Split input:** on the character \n
 - and deliver:** ASCII strings
- Output:**
 - Add character to output messages \n
- Request:**
 - Default response timeout 10000 ms

A yellow tip box at the bottom states: "Tip: the 'Split on' character is used to split the input into separate messages. Can accept chars (\$), escape codes (\n), or hex codes (0x03)."