

Didacticiel de la maquette

Didacticiel papier

Table des matières

Introduction	4
Chapitre 1 - La maquette	5
Chapitre 2 - Présentation des modules	7
1. Le module Lampe	7
2. Le module relais	8
3. Le module Télerrupteur	11
4. Le module transformateur	13
5. Le module bouton poussoir	15
6. Les modules Arduino et Raspberry	18
7. Le module capteur PIR	19
Chapitre 3 - Utiliser NODE RED	23
1. Gérer une relais avec NODE RED	23
2. Gérer un relais connecté à Arduino avec Raspberry	24
3. Comment gérer un télerrupteur avec NODE RED	26
4. Gérer le télerrupteur et avoir un retour sur l'état de la lampe	27
Chapitre 4 - Annexes	30
1. Toutes mes procédures	31
1.1. Ajouter une Arduino sur NODE RED	32
1.2. MISE EN ROUTE ARDUINO	34
1.3. Comment importer et exporter des Clipboard	37
1.4. Ajouter des nœuds sur NODE RED	40
1.5. Utiliser le dashboard NODE RED	42
2. Fichier utile	46

Introduction

Ce didacticiel à pour but de vous initier aux grands principes d'électronique et de programmation avec les éléments de la maquette. Vous pourrez ensuite les mettre en œuvre pour un projet futur.

Chapitre 1

La maquette

Objectif

L'objectif de la maquette domotique est simple, faire en sorte que vous ayez envie d'utiliser la maquette pour découvrir la programmation, les éléments de base tous cela en ayant une certaine liberté puisque grâce à ce didacticiel vous pouvez vous diriger vers différents points qui vous intéressent.

Une fois que vous maîtriserez la maquette, j'espère que vous aurez envie de prendre une Arduino, une Raspberry, des capteurs et de les mettre en oeuvre !

Présentation rapide

La maquette est composé d'une Raspberry, d'une Arduino, de plusieurs modules tels que :

- relais
- lampe
- capteur présence
- télérupteur

A LIRE AVEC ATTENTION

Règle n°1 : Il ne faut jamais être en contact direct avec une partie du circuit de puissance ou de commande

L'intérêt de la domotique est de réaliser des montages qui permettent de piloter des matériels qui fonctionnent sous 220V. Cela demande beaucoup de mesures de sécurité. Si vous êtes prêts à prendre vos précautions, nous allons faire un point sur le courant domestique.

Contrairement à votre Arduino qui fournit du courant continu, le courant de vos prises murales est alternatif, c'est-à-dire qu'il change de polarité (+ et -) très rapidement. En France par exemple, le courant change à la fréquence de 50Hz, donc 50 fois par seconde !

Il ne change pas d'un coup, il passe de +220 V à -220 V en suivant une courbe sinusoïdale.

On ne peut pas connecter n'importe quel appareil sur une tension alternative, car certains ne fonctionnent que sur des tensions continues.

En plus de cette oscillation, il faut savoir que le courant domestique (celui de votre prise) est à 220V, soit 44 fois supérieur à la tension délivrée par l'Arduino. Cette tension est dangereuse pour l'être humain.

Si un vous touchez un fil dénudé porteur de courant domestique, ce courant va être capable de passer dans son corps jusqu'à rejoindre le sol (la terre). Durant ce passage, il va brûler les tissus (muscles, cellules, sang) et donc provoquer des brûlures irréversibles.

Alors il suffit d'enlever son doigt ? !

Même en le voulant, vous ne pourriez pas !

En effet, nos muscles (cœur y compris) sont commandés par notre cerveau grâce à notre réseau de

nerfs. À travers nos nerfs circule un courant électrique très très faible. Nous sommes donc pilotés grâce à de l'électricité !

Mais lorsque vous êtes électrisé (donc parcouru par un courant de forte tension), les signaux électriques envoyés par son cerveau sont brouillés ! Du coup il est paralysé. Il reste bloqué, sans pouvoir faire quoi que ce soit. Et si son propre poids ne le fait pas tomber (ce qui peut arrêter le contact avec le fil) ou si quelqu'un ne coupe pas le courant, alors vous serez électrocuté (c'est-à-dire mort).

On ne cherche pas à tirer ou pousser quelqu'un qui est électrisé, car sinon, on s'électrise soi-même, ce qui fait deux personnes électrisées...

Ce petit paragraphe peut paraître choquant pour les âmes sensibles, mais il n'a qu'un seul but : vous mettre en garde à ce sujet. Donc la première règle à appliquer lorsqu'on utilise un courant domestique :

Règle n° 1 : Il ne faut jamais être en contact direct avec une partie du circuit de puissance ou de commande

Réfléchissez avant d'agir ! Prenez le plus de précaution possible, comme vous avez pu le voir en cours de sécurité. L'important n'est pas d'aller vite, mais d'être prudent à tout moment !

Chapitre 2

Présentation des modules

La présentation des modules peut vous paraître facile, tant mieux, vous avez un bon nombres de bases acquises, mais lisez cette présentation afin de vous approprier au mieux tous les éléments de la maquette pour ensuite exploiter tout le potentiel à votre disposition.

Bonne lecture

1. Le module Lampe

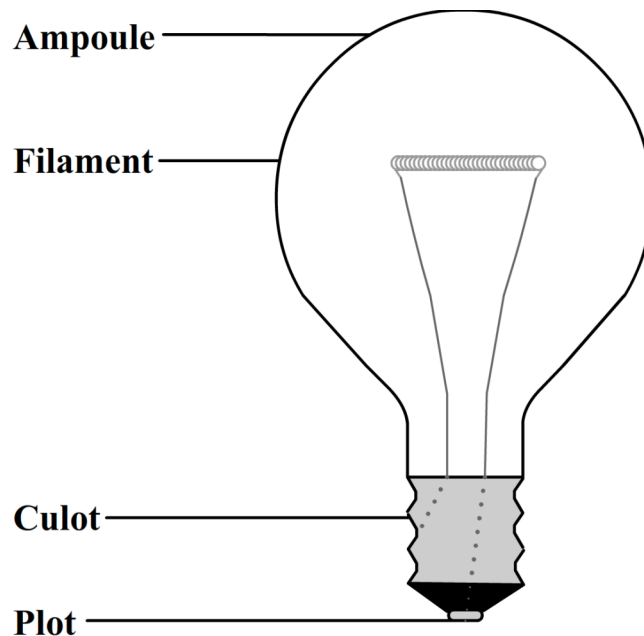
Ce module est très important, c'est en effet la finalité de cette maquette. L'objectif étant d'allumer et éteindre (gérer) une ou plusieurs lampes.

C'est le module le plus simple et le plus facile à visualiser puisque il s'agit d'un effecteur

Présentation générale de la lampe

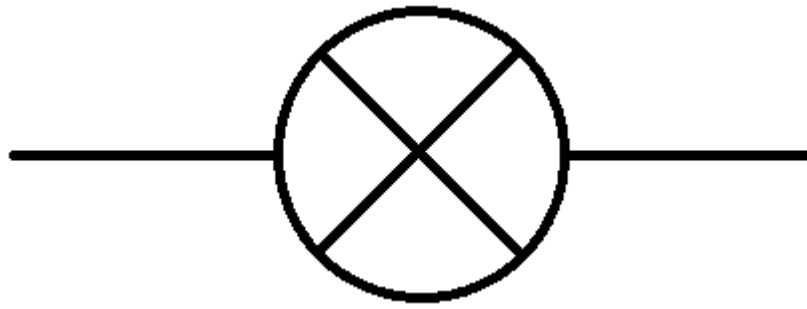
Tous le monde connaît la lampe.

Elle est composé comme ci-dessous :



Plot (-) / Culot (+)

Dans un schéma électrique elle est représentée comme cela :

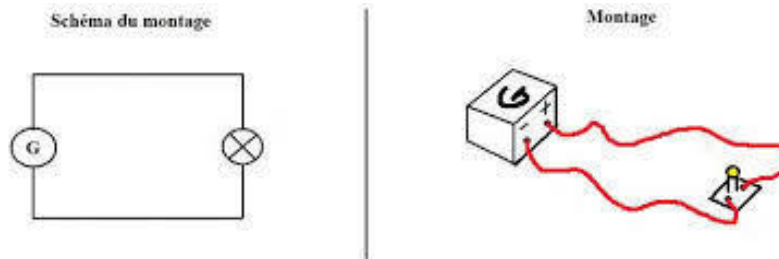


La lampe peut fonctionner avec différentes tensions, certaines en 12V, d'autre en 230V... Il faut donc faire attention lorsque cette dernière sera branchée, puisqu'il faudra choisir l'alimentation en conséquence.

Sur notre maquette la lampe fonctionne en 24V, utiliser donc la bonne tension (il y a un transformateur à cet effet, transformateur que nous détaillerons dans un autre chapitre)

💡 Schéma de branchement

La lampe est très simple à brancher, en effet elle s'intègre facilement dans un circuit, il suffit de la mettre en série dans un circuit avec la bonne tension bien sur.



Nous voyons donc qu'il suffit d'un générateur, de deux fils et de la lampe pour faire fonctionner ce module.

Brancher le + sur une borne de la lampe et le - sur l'autre. le courant et la tension vont traverser la lampe et l'alimenter, cette dernière va donc s'allumer. Si le circuit est rompu dans ce cas la lampe s'éteint. Le circuit peut être rompu par un interrupteur mais aussi par un fil coupé.

👁️ Code du module

Ce module n'a besoin d'aucun code pour fonctionner puisque c'est un effecteur.

2. Le module relais

Info module :

📁 Présentation générale

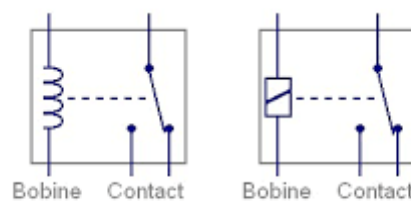
Un relais électromécanique est un organe électrique permettant de dissocier la partie puissance de la partie commande : il permet l'ouverture et la fermeture d'un circuit électrique d'une certaine tension grâce à une tension de commande parfois très faible (5V...)



Les relais peuvent être de deux types :

- mécanique : une bobine faisant commuter les contact
- statique : silencieux et pouvant commuter à des fréquences importantes

Voici un schéma de relais mécanique :



Symboles du relais

Le relais fonctionne comme un interrupteur : on lui applique une tension de commande, il laisse passer la tension de puissance, lorsqu'on relâche la tension de commande alors le contact s'ouvre et la tension de puissance ne passe plus. MAIS, le relais possède parfois trois broches pour la tension de puissance : une pour le commun (le fil qui arrive) une NO et une NC.

Le relais est très utile, il peut commander n'importe quel système puisque il laisse passer simplement le courant ou non.

Petit plus relais statique

avec les relais statique il est également possible de générer des trains d'impulsions afin de commander des moteurs à courant continu et de les faire tourner plus ou moins vite.

💡 Schéma de branchement

Ce module se branche tout aussi facilement qu'une lampe. Il comporte 6 broches mais seulement 5 nous serviront.



Il y a deux côtés, un côté puissance (vis avec protection bleu), et un côté commande

avec des broches 5V type Arduino.

Côté commande - Microcontrôleur (Arduino ou Raspberry) :

- : brancher sur le GND du microcontrôleur

+: brancher sur le 5V du microcontrôleur

S : brancher sur la pin de l'UC qui va contrôler le relais

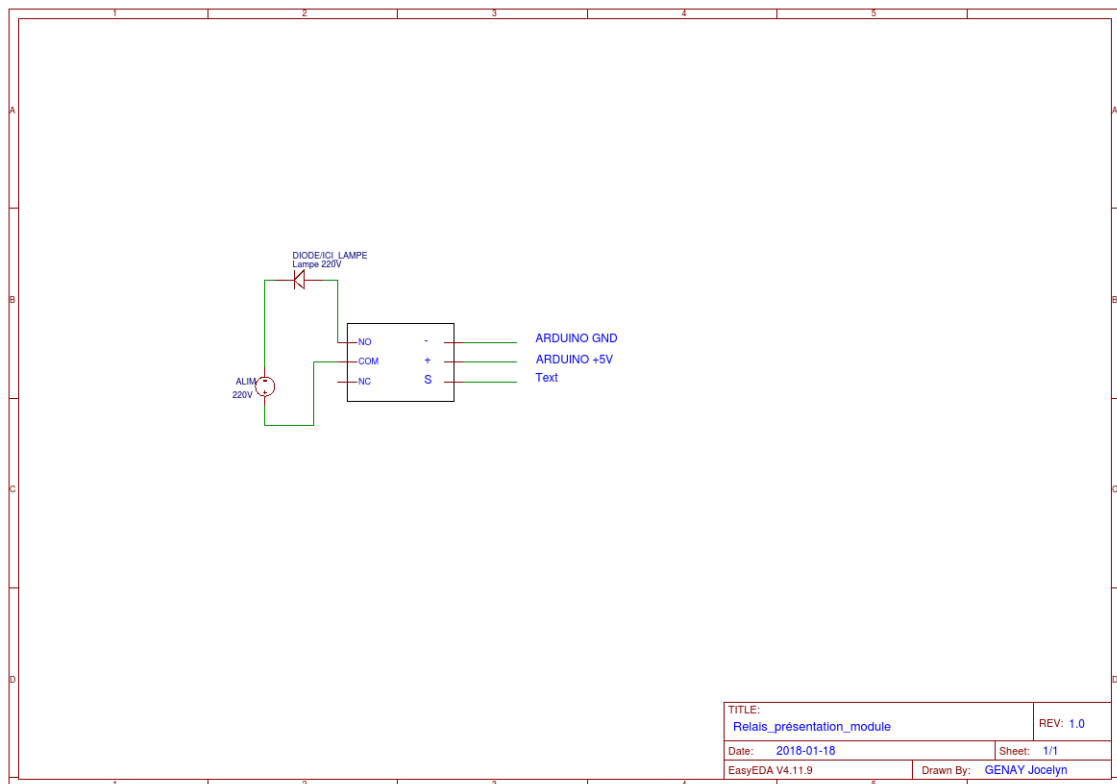
Côté puissance - 220V (ou autre tension plus ou moins grande) :

Commun : brancher le + du générateur

NO : brancher un fil jusque l'effecteur que l'on veut commander (la lampe par exemple).

NC : ici cette sortie ne sert pas puisque l'on ne veut pas commander la lampe en logique inverse.

Après avoir mis un fil sur un côté de la lampe, il faut remettre un fil de celle-ci jusque au - du générateur (voir [Module Lampe \(cf. Le module Lampe\)](#)^[p.7])



Code du module

Il est possible d'actionner un relais avec la Raspberry ou avec une Arduino.

Commençons par l'Arduino :

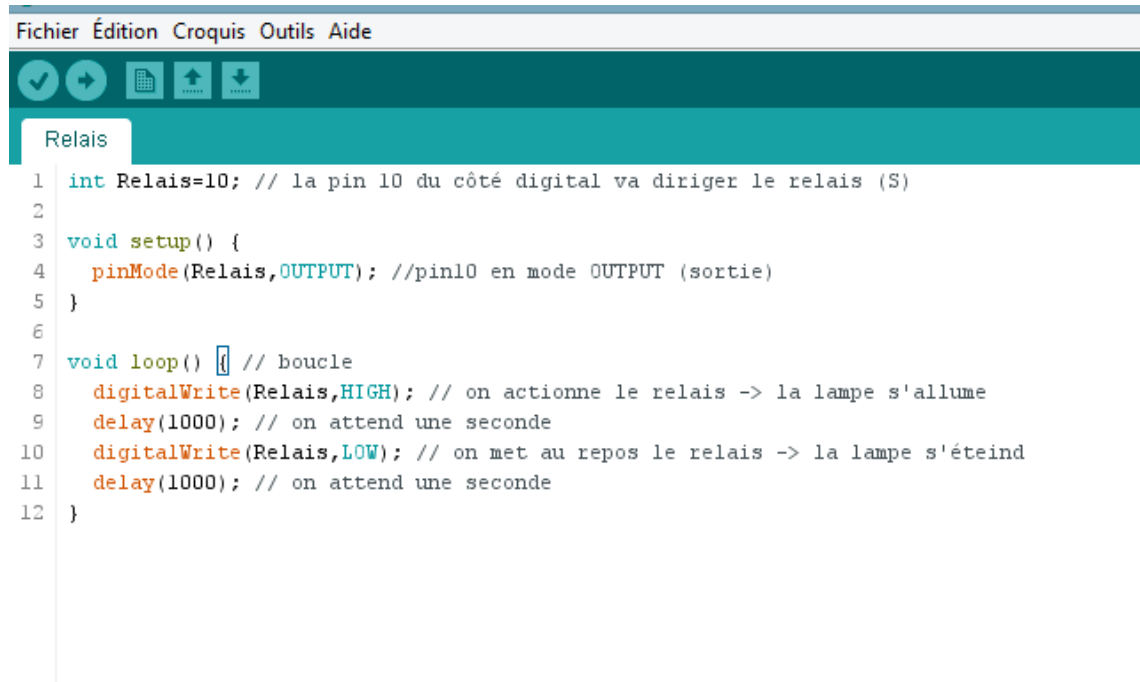
Si vous ne connaissez pas Arduino, vous trouverez ici une [procédure explicative \(cf. MISE EN ROUTE ARDUINO\)](#)^[p.34] afin d'avoir quelque base pour comprendre la programmation

Voir pour rediriger vers quelque site web utile

Explication du code :

Le but va être tout simple d'envoyer 5V (HIGH) sur une broche lorsque l'on veut allumer la lampe. Par exemple ci-dessous je met le code pour actionner le relais pendant une seconde, puis le mettre au repos une seconde. Puisque un lampe est branchée à ce relais, elle va s'allumer une seconde et s'éteindre une seconde en suivant le relais.

Code :



```

Fichier  Édition  Croquis  Outils  Aide
Relais
1  int Relais=10; // la pin 10 du côté digital va diriger le relais (S)
2
3  void setup() {
4      pinMode(Relais,OUTPUT); //pin10 en mode OUTPUT (sortie)
5  }
6
7  void loop() { // boucle
8      digitalWrite(Relais,HIGH); // on actionne le relais -> la lampe s'allume
9      delay(1000); // on attend une seconde
10     digitalWrite(Relais,LOW); // on met au repos le relais -> la lampe s'éteint
11     delay(1000); // on attend une seconde
12 }

```

3. Le module Télérupteur

Info module :

Ce module est utile lorsque l'on veut commander un effecteur avec plusieurs bouton poussoir, ici nous allons voir comment l'utiliser pour allumer une lampe avec un bouton physique dans un pièce, mais aussi comment faire avec NODE RED et utiliser sa structure pour nous permettre de connaître l'état de l'interrupteur.

Présentation générale

Si vous n'avez pas encore vu le module **RELAIS** (cf. *Le module relais*)^[p.8], il serait judicieux de le faire puisqu'un télérupteur est un relais un peu spécial.

Pour le relais, il fallait envoyer une tension de commande pour l'activer, lorsque l'on enlève cette tension, la tension de puissance se coupe elle aussi. Et bien le télérupteur fonctionne presque pareil, mais comme ceci :

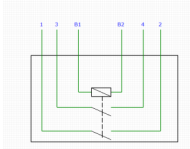
on envoi une tension de commande, la bobine est sous tension, l'interrupteur se ferme et la tension de puissance passe.

on arrête quelque millisecondes après d'envoyer la tension de commande, et la bobine reste active, donc l'interrupteur est toujours fermé et la tension de puissance passe toujours.

Pour arrêter la tension de puissance il faut renvoyer une impulsion de commande sur la bobine, et l'interrupteur s'ouvre.

On observe donc que le télérupteur fonctionne avec des impulsions.

Voici le schéma d'un télérupteur :



On observe sur ce télérupteur qu'il y a deux interrupteur, ceci peut être très pratique pour commander deux tension de puissance différentes. Nous nous en servirons plus tard pour connaître l'état du télérupteur.

B1/B2 : bobine de commande, celle-ci peut être commandé en 220V ou autre, ici elle sera commandé en 12V

1/2 : interrupteur 1

3/4 : interrupteur 2

Mais vous me direz, pourquoi mettre un télérupteur alors que le relais fait le même travail, mis à part l'auto-maintient des interrupteurs !

Eh bien voilà, dans une maison domotique, on veut commander se lumière avec la Raspberry, très bien mais si je suis dans la pièce et que je n'ai pas mon téléphone, ou si c'est quelqu'un qui ne possède pas de téléphone, il faut bien allumer la lumière avec un interrupteur réel. Il suffit donc d'en mettre un dans la pièce mis en parallèle sur le télérupteur et il sera possible de commander le télérupteur avec un UC mais aussi avec un interrupteur réel qui sera le bouton poussoir que l'on verra plus tard, en effet ce dernier sert à envoyer des impulsions que l'on va prendre pour une impulsions de commande.

INTÉRÊT à retenir : Permettre de commander une lampe avec la Raspberry et avec un bouton poussoir !

Schéma de branchement

Pour ce branchement vous aller avoir besoin de :

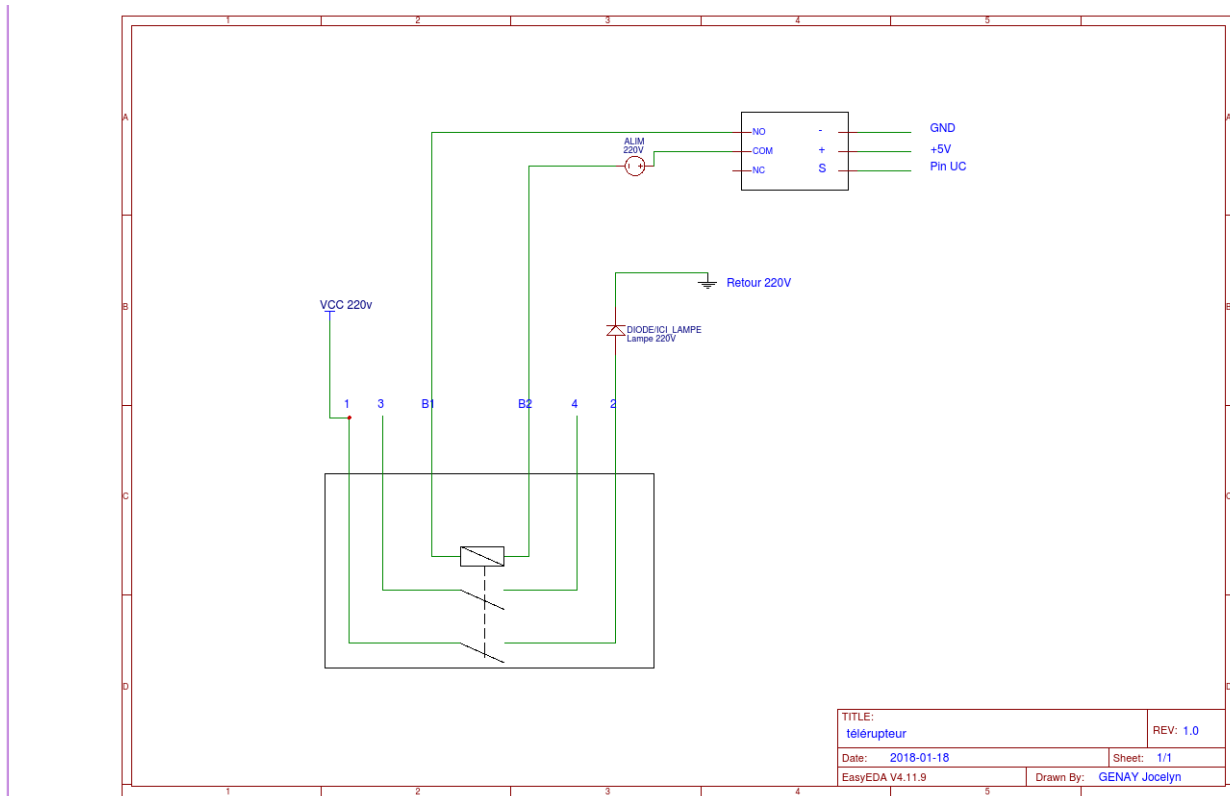
télérupteur

Lampe

Relais

UC

Il faut suivre le branchement ci-dessous :



Code du module

Nous allons faire le code avec la Raspberry :

Suivre la procédure pour allumer NODE-RED

Puis faire ceci :

Lorsque l'on le souhaite, envoyer un 1 sur le relais, qui va alimenter la Raspberry quelque secondes, puis arrêter, le télérupteur fait donc passer la tension de puissance.

Puis lorsque l'on le souhaite, envoyer de nous un 1 pour créer une impulsion de commande avec le relais et éteindre le télérupteur. La tension de puissance ne passe plus.

4. Le module transformateur

Info module :

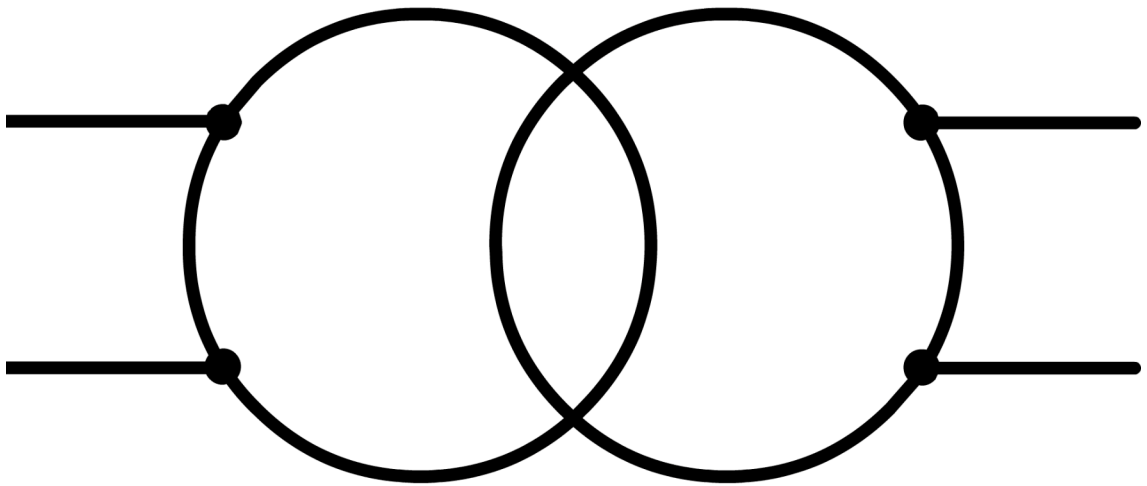
Présentation générale

Sur la maquette il y a deux type de circuit : le circuit commande et le circuit puissance.

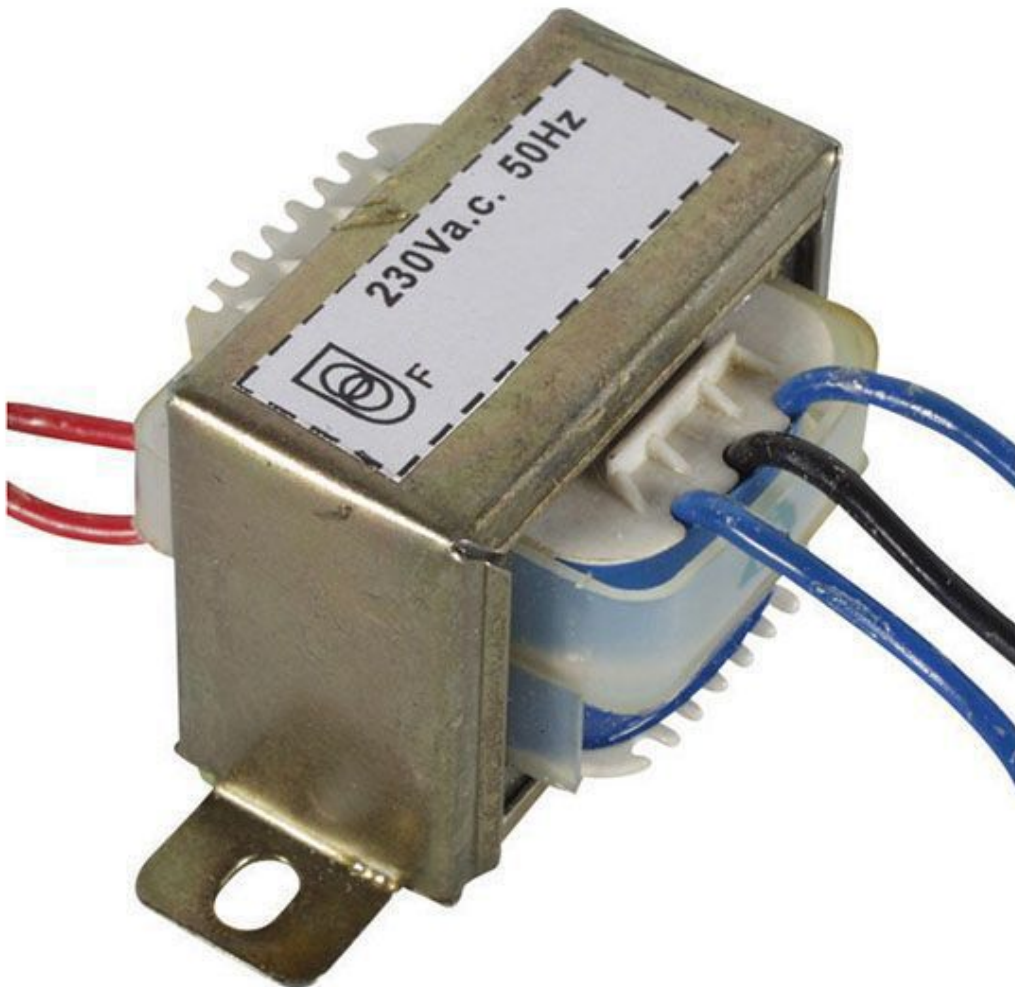
Circuit commande : généralement en basse tension pour économiser de l'énergie, il permet d'envoyer des signaux de commande comme son nom l'indique. Ici sur la maquette il y a du 5V et du 3.3V pour ce circuit.

Circuit puissance : tout système à besoin de puissance, et parfois de très grande puissance. Ici pour des questions de sécurité la puissance sera du 24V alternatif (cf cours de sécurité, tension de sécurité = 50 V dans un local sec). Le 230V serait trop élevé et c'est pour cette raison que le transformateur entre en jeu. Il faut un élément qui transforme le 230V et 24V : le transformateur

Voici le symbole d'un transformateur :



Voici à quoi ressemble un vrai transformateur. On peut le distinguer facilement sur la maquette.



Le transformateur sur la maquette transforme donc du 230V alternatif (réseau EDF) en 24V alternatifs permettant d'allumer la lampe par exemple.

Notions :

Il existe :

- des transfo abaisseur, qui abaissent la tension (c'est notre cas)
- des transfo élévateur, qui élèvent la tension
- Le primaire, le côté de la source, c'est le signal que l'on veut transformer
- Le secondaire, l'autre côté, c'est le signal que l'on obtient après transformation.
- Il est possible de convertir un signal sinusoïdale en continu et vice versa.



Schéma de branchement

Le branchement est assez simple, du côté du primaire (le réseau EDF pour notre cas) on branche la phase et le neutre. La terre peut être raccordé si c'est un transformateur de sécurité par exemple, c'est le cas ici.

De plus il y a pour notre transformateur un fusible qui protège l'ensemble. La partie en aval et en amont sont donc indépendante, pratique si il y a des erreurs de montage...

Du côté secondaire nous obtenons une source avec du 0V et du 24V

Pour des raisons de sécurité ces branchements seront déjà effectué et vous aurez simplement à venir vous servir de l'alimentation en 24V alternatif.



Code du module

Pas besoin de code pour ce module, c'est simplement une alimentation (l'alimentation du circuit de puissance).

5. Le module bouton poussoir

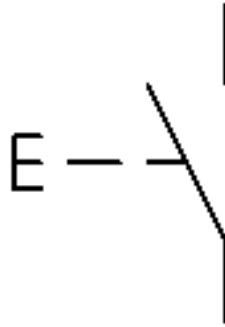
Info module :



Présentation générale

Un bouton poussoir c'est assez simple, c'est comme un interrupteur, sauf que ça permet d'envoyer une impulsion. un interrupteur classique possède deux positions : ouvert ou fermé, et bien pour le bouton poussoir c'est la même chose. Prenons le cas d'un bouton poussoir NO (normalement ouvert), si on l'insère dans un circuit, le bouton ne laisse pas passer la tension, et dès que nous restons appuyé sur le bouton il laisse passer la tension, et dès qu'on le relâche la tension ne passe plus car il reprend sa position initiale.

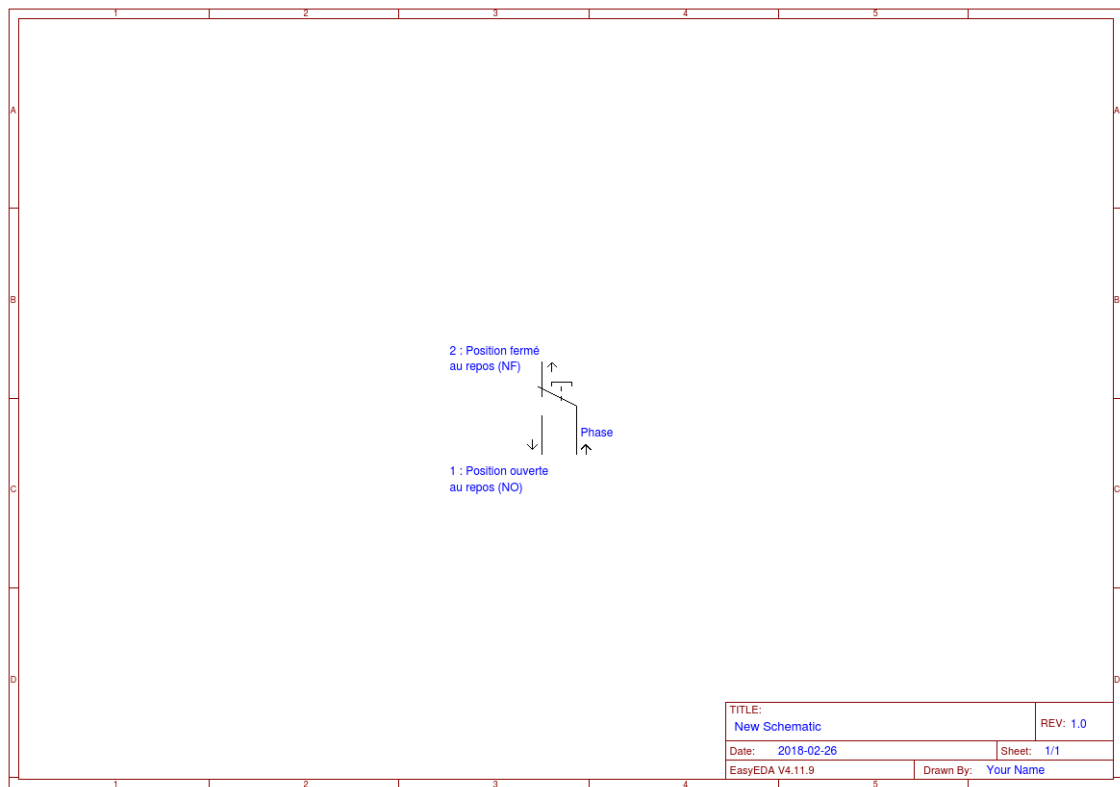
Voici son symbole :



Dans notre maquette le bouton poussoir est un bouton poussoir que l'on peut retrouver dans une maison :



Ce bouton poussoir peut être utilisé pour faire un bouton poussoir NF ou NO, en effet, sur le schéma que le constructeur nous propose on peut voir ceci :



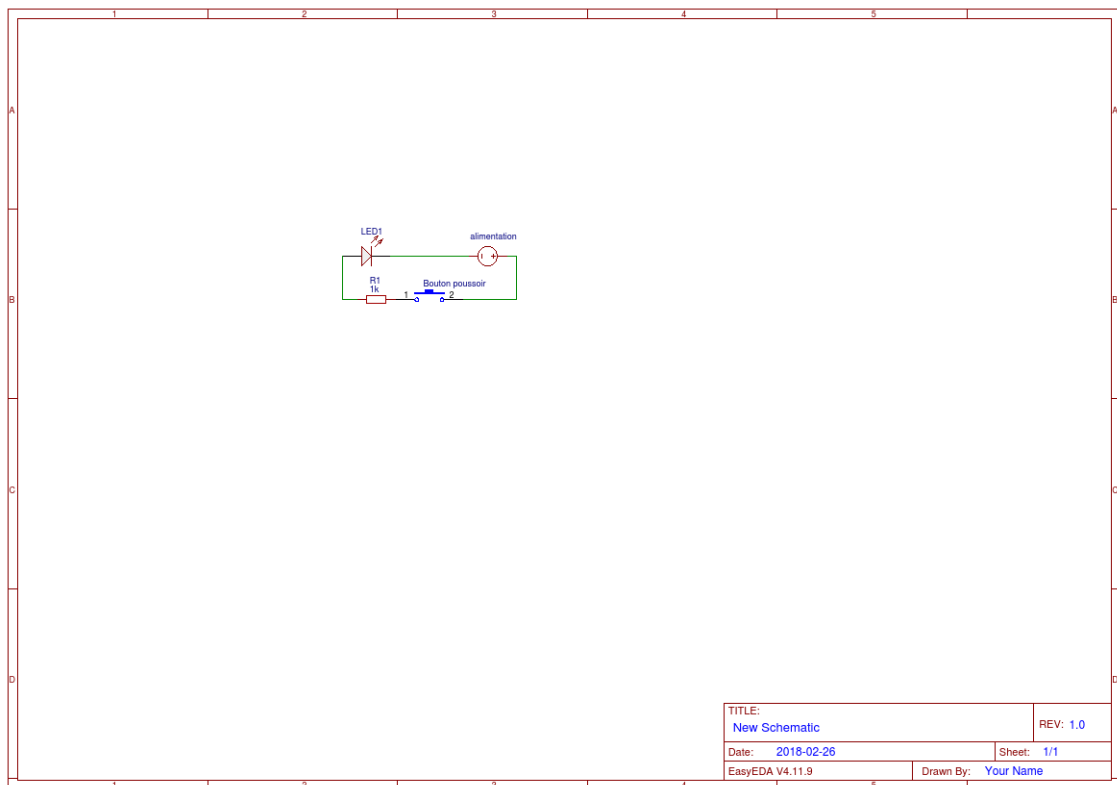
Pourquoi insérer un bouton poussoir dans la maquette.

Eh bien c'est simple, nous avons vu précédemment le télérupteur et ce dernier servait à commander une lampe. Pour commander ce télérupteur il fallait une impulsion, eh bien grâce à ce bouton poussoir il sera très simple de commander le télérupteur, il suffira de le mettre sur la bobine de ce dernier, et d'appuyer une fois pour laisser une impulsion passer et commander le télérupteur.

💡 Schéma de branchement

Nous allons voir comment intégrer le bouton poussoir dans un circuit simple.

Voilà le branchement :



Lorsque l'on appuie sur le bouton poussoir et qu'on le maintient, la tension passe et la lampe est alimentée, dès que l'on lâche le bouton poussoir celui-ci ne laisse plus passer la tension et la lampe s'éteint.

Maintenant il faut imaginer que l'on remplace la lampe par la bobine du télérupteur... Eh oui, nous pouvons tout simplement commander le télérupteur avec une impulsion, ce dernier commandera la lampe.

Si vous vous dites : "pourquoi ne pas mettre un simple interrupteur ?" Je ne peux que vous conseiller de retourner dans le chapitre télérupteur et de comprendre son intérêt.

👁 Code du module

Le bouton poussoir est un interrupteur physique qui n'a besoin d'aucun code pour fonctionner. Il suffit de bien le câbler et de l'utiliser ensuite.

6. Les modules Arduino et Raspberry

Info module :

📁 Présentation générale

L'homme a un cerveau, les machines ont le leur. Ici nous avons deux cerveaux, la Raspberry et l'Arduino, ce sont tous les deux des microcontrôleurs très pratiques. Ils permettent de mettre en œuvre divers montages et vont nous être très utiles.

Pourquoi Arduino et Raspberry ? Eh bien c'est simple, l'Arduino est très pratique pour la mise en place de capteurs, tandis que Raspberry est très pratique en termes de communication, interface,

gestion de programme... Allier les deux est un atout, il ne faut donc certainement pas les opposer.

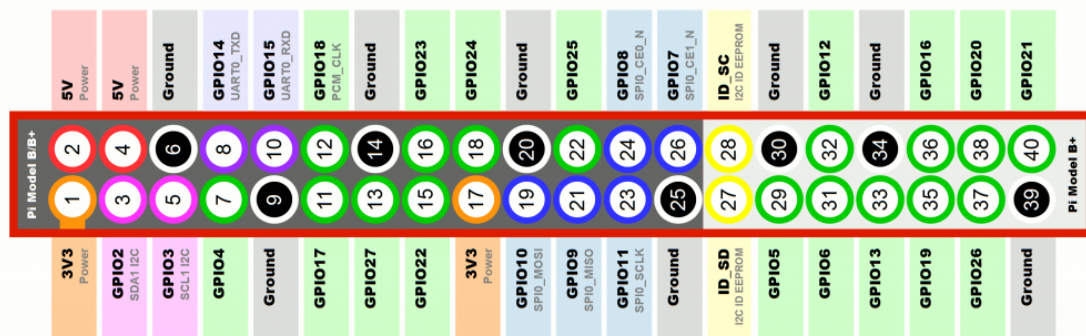
Dans notre cas nous n'avons pas beaucoup de capteur, seulement celui de présence d'ailleurs, mais il en existe beaucoup et je ne peux que vous conseiller d'y consacrer du temps car c'est très intéressant. Ici l'Arduino nous sert plutôt de relais.

En effet une Raspberry coûte tout de même dans les 40€, et mettre une Raspberry par pièce n'est pas très économe ni productif, je me suis dit, pourquoi ne pas créer un cerveau principale, avec plusieurs cerveaux dans chaque pièces. La Raspberry sera donc notre cerveau principale, et plusieurs Arduino composeront nos pièces et feront office de relais entre pièce et Raspberry.

Nous utiliserons NODE-RED avec Raspberry, ils nous servira à collecter les informations de l'Arduino et de les gérer.

Pour l'Arduino il est possible d'utiliser l'IDE Arduino mais normalement nous n'en aurons pas besoin. Cependant il se peut que je communique quelques code afin que vous puissiez l'utiliser avec le relais par exemple...

📁 Dénomination des broches Raspberry



💡 Schéma de branchement

Cela dépendra des montages.

👁️ Code du module

Aucun code, ce sont eux les exécutants des codes.

7. Le module capteur PIR

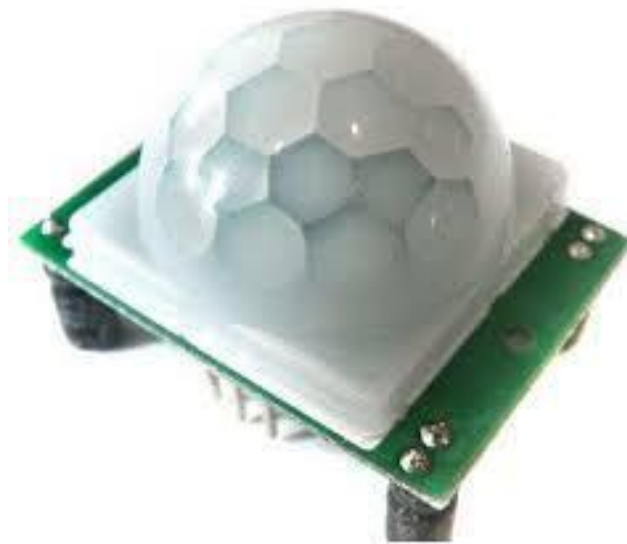
Info module :

📁 Présentation générale

Le capteur de présence s'appelle également PIR motion (Pyroelectric InfraRed sensor). Il fonctionne facilement, il détecte une variation d'onde infrarouge et génère alors un courant.

Ce capteur très simple d'utilisation va nous être très pratique.

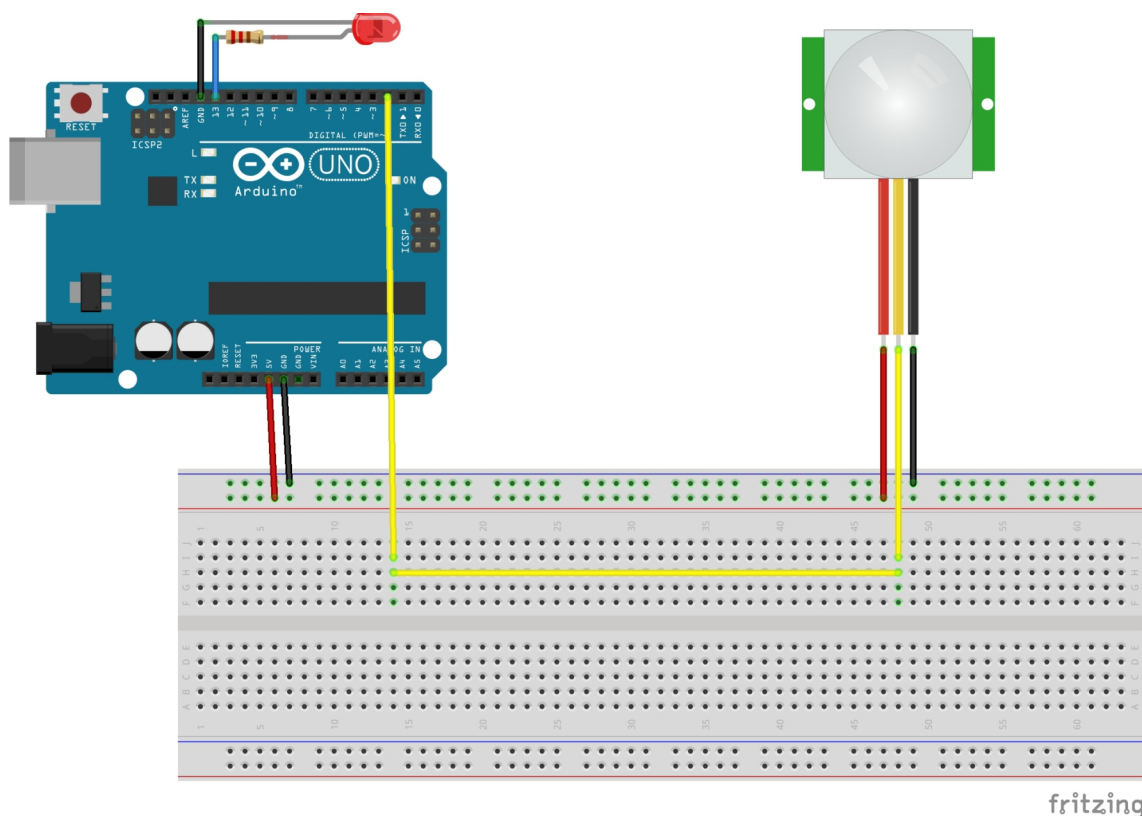
Voilà à quoi il ressemble :



Il possède trois broche : le 5V, le GND, le signal de sortie (1 ou 0) en fonction de son état.

💡 Schéma de branchement

Voici le schéma de branchement de ce module :



Bien sur dans le cas de notre maquette, il faudra brancher la sortie 13 au relais, qui commandera la lampe. Mais rien ne vous empêche d'allumer un LED.

👁️ Code du module

Pour ce qui est du code :

A chaque fois que l'on passe devant le PIR, une LED s'allume et un message est envoyé sur le terminale Arduino en disant la valeur du capteur (0 ou 1) et lorsqu'il passe à 1 il envoie un message

comme quoi un mouvement est détecté, et lorsqu'il passe à 0 il envoie un message comme quoi il y a un mouvement. De plus au départ il y a la calibration donc il est transmis sur la terminale qu'il y a la calibration qui est effectuée avec une succession de petit point le temps de la calibration.

Il y a des commentaires (//commentaires) à côtés des lignes pour préciser à quoi elles correspondent.

Le code pour le capteur PIR est le suivant : (à utiliser avec Arduino. Un autre chapitre sera dédié à l'utilisation avec NODE RED)

```
int calibrationTime = 10; // temps que l'on accorde au capteur de présence pour s'initialiser (entre 10 et 60 secondes)
```

```
int ledPin = 13; // broche de la LED
```

```
int inputPin = 2; // Entrée de la valeur du capteur
```

```
int pirState = LOW; // on commence avec aucun mouvement détecté
```

```
int val = 0; // valeur de la lecture du capteur
```

```
void setup() {
```

```
  pinMode(ledPin, OUTPUT); // la broche de la LED est une sortie
```

```
  pinMode(inputPin, INPUT); // la broche du capteur est une entrée
```

```
  Serial.begin(9600);
```

```
  Serial.print("calibration du capteur ");
```

```
  for(int i = 0; i < calibrationTime; i++){
```

```
    Serial.print("."); // mettre des petits points, un point par seconde jusqu'à la fin de la calibration
```

```
    delay(1000);
```

```
  }
```

```
}
```

```
void loop(){
```

```
  val = digitalRead(inputPin); // lire la valeur du capteur et la stocker dans val
```

```
  Serial.println(val); // écrire la valeur val
```

```
  if (val == HIGH) { // Si la valeur = 1 ( high ) alors il y a un mouvement
```

```
    digitalWrite(ledPin, HIGH); // allumer la LED
```

```
    delay(150);
```

```
    if (pirState == LOW) {
```

```
      Serial.println("mouvement detecte!");
```

```
      pirState = HIGH;
```

```
    }
```

```
  }
```

```
  else {
```

```
    digitalWrite(ledPin, LOW); // éteindre la LED
```

```
    delay(300);
```

```
    if (pirState == HIGH){
```

```
      Serial.println("aucun mouvement");
```

```
      pirState = LOW;
```

```
    }
```

| }
| }

Chapitre 3

Utiliser NODE RED

Information

Vous avez normalement acquis assez de connaissances sur tous les modules et sur le fonctionnement de leurs environnements. Nous allons pouvoir maintenant utiliser NODE RED pour gérer notre maquette...

Nous allons pouvoir gérer les lampes en fonction des gens dans une pièce, ou en fonction de son envie avec des boutons switches sur un tableau de bord appelé dashboard. Vous pourrez découvrir grâce à ce tutoriel des fonctionnalités intéressantes et j'espère que vous aurez envie d'en apprendre plus en allant chercher des infos sur internet.

1. Gérer une relais avec NODE RED

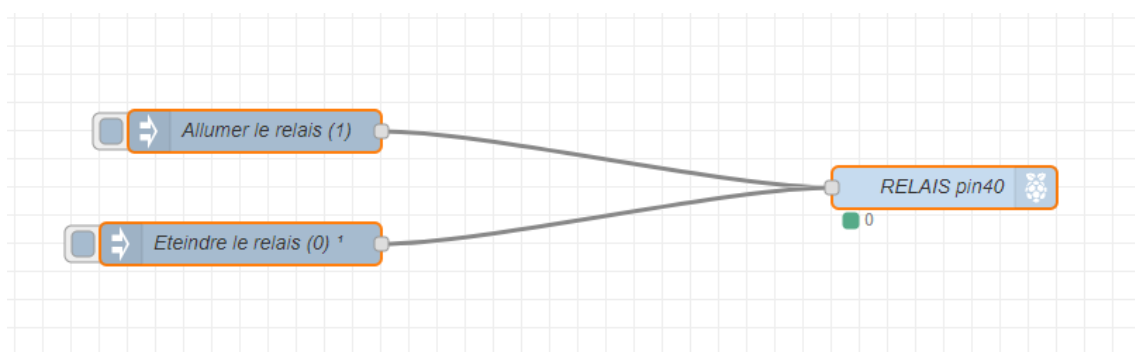
OBJECTIF

L'objectif ici est de pouvoir commander une lampe en 24V depuis notre Raspberry en utilisant NODE RED

Lorsque l'on enverra 1 sur une pin de la raspberry, ça alimentera le relais, qui alimentera quant à lui la lampe en 24V.

NODE RED

On doit obtenir cela :



Lorsque l'on appuie sur 0, la pin40 se met à 0 (0V), lorsque l'on appuie sur 1, la pin40 se met à 1 (+5V).

On peut voir que c'est très simple de gérer un relais, ce petit programme peut également commander une led 5V, un moteur 5V ...

Le code si besoin

Si vous ne savez pas comment exporter ou importer un clipboard NODE RED : [Procédure](#)

Comment importer et exporter des Clipboard (cf. *Comment importer et exporter des Clipboard*)^[p.37]

```
[{"id":"3028667f.3f040a","type":"inject","z":"b07a0d8d.116a8","name":"Eteindre le relais (0)","topic":"","payload":"0","payloadType":"num","repeat":"","crontab":"","once":true,"x":280,"y":260,"wires":[["f2ae24b3.15e278"]]}, {"id":"d9a2e314.1ad37","type":"inject","z":"b07a0d8d.116a8","name":"Allumer le relais (1)","topic":"","payload":"1","payloadType":"num","repeat":"","crontab":"","once":false,"x":290,"y":180,"wires":[["f2ae24b3.15e278"]]}, {"id":"f2ae24b3.15e278","type":"rpi-gpio out","z":"b07a0d8d.116a8","name":"RELAIS pin40","pin":"40","set":"","level":"0","freq":"","out":"out","x":780,"y":220,"wires":[]}]
```

MONTAGE

brancher la pin40 à l'effecteur (led, relais...) vous savez comment brancher le relais grâce à la présentation du module relais.

SUPPLÉMENT REMARQUE

Il est très important que vous sachiez comment faire pour ce code assez. Forcez vous à le faire et à comprendre, cela vous sera utile quand vous voudrez faire vos propres codes.

2. Gérer un relais connecté à Arduino avec Raspberry

OBJECTIF

L'objectif comme je l'avais dit précédemment est de gérer un élément avec NODE RED. Cet élément se trouvant dans une autre pièce il faut passer par l'Arduino. Nous allons voir comment lier Arduino et Raspberry simplement.

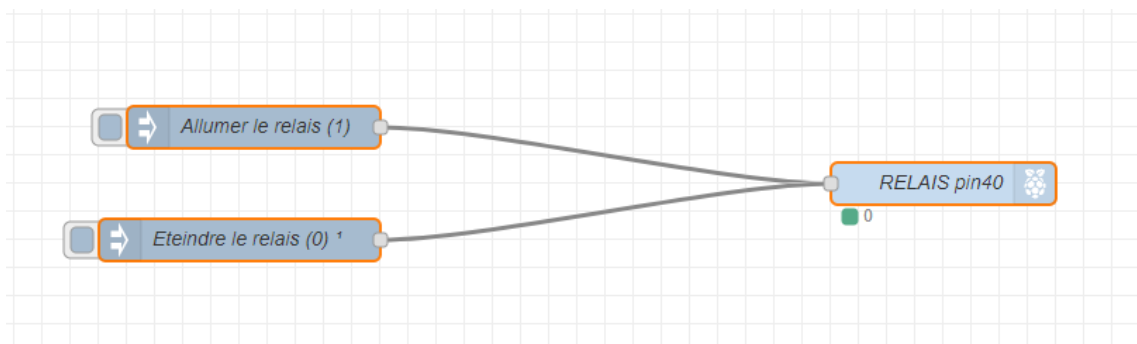
NODE RED

Tout d'abord il faut installer une librairie permettant d'avoir d'autre noeuds : procédure pour ajouter des noeuds

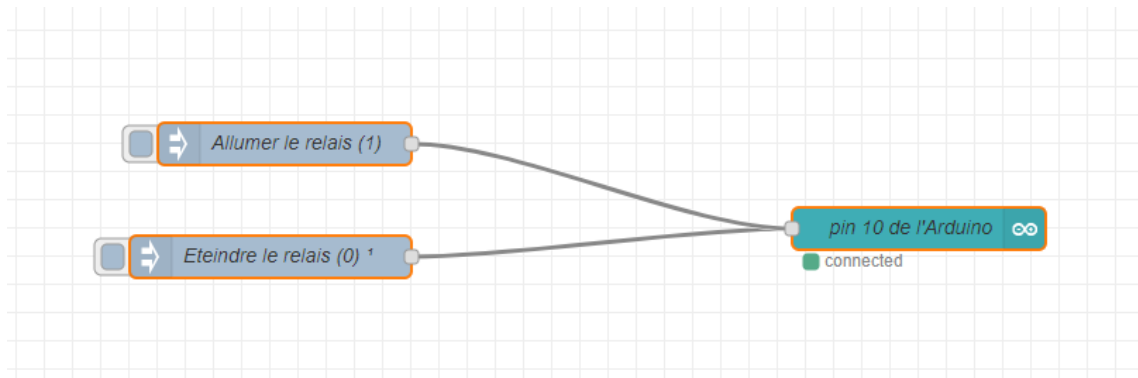
Ajouter la librairie : node-red-node-arduino V 0.0.14

Vous obtiendrez deux neouds très intéressant qui vont vous permettre de communiquer avec l'Arduino en USB. (voir montage)

Voilà ce que vous devez avoir pour commander le relais sur la Raspberry pin40



Si vous voulez commander un relais sur la pin10 de l'Arduino voilà ce que vous aurez :



Pour ajouter une Arduino à votre raspberry voilà la procédure : [ajouter une arduino \(cf. Ajouter une Arduino sur NODE RED\)](#) [p.32]

Le code si besoin

Voici le code si vous en avez besoin : [Procédure Comment importer et exporter des Clipboard \(cf. Comment importer et exporter des Clipboard\)](#) [p.37]

Pour commander le relais sur la Raspberry :

```
[{"id":"c12dc250.713fb","type":"inject","z":"b07a0d8d.116a8","name":"Eteindre le relais (0)","topic":"","payload":"0","payloadType":"num","repeat":"","crontab":"","once":true,"x":300,"y":440,"wires":
[["30fd15f5.fe317a"]]}, {"id":"ba34ffdd.9f3b","type":"inject","z":"b07a0d8d.116a8","name":"Allumer le
relais (1)","topic":"","payload":"1","payloadType":"num","repeat":"","crontab":"","once":false,"x":310,"y":
360,"wires":[["30fd15f5.fe317a"]]}, {"id":"30fd15f5.fe317a","type":"rpi-gpio out","z":"b07a0d8d.116a8","
name":"","pin":"40","set":"","level":"0","freq":"","out":"out","x":750,"y":420,"wires":[]}]
```

Pour commander le relais sur l'Arduino

```
[{"id":"3028667f.3f040a","type":"inject","z":"b07a0d8d.116a8","name":"Eteindre le relais (0)","topic":"","
payload":"0","payloadType":"num","repeat":"","crontab":"","once":true,"x":280,"y":260,"wires":
[["764a9f3e.2120f"]]}, {"id":"d9a2e314.1ad37","type":"inject","z":"b07a0d8d.116a8","name":"Allumer le
relais (1)","topic":"","payload":"1","payloadType":"num","repeat":"","crontab":"","once":false,"x":290,"y":
180,"wires":[["764a9f3e.2120f"]]}, {"id":"764a9f3e.2120f","type":"arduino out","z":"b07a0d8d.116a8","
name":"pin 10 de l'Arduino","pin":"10","state":"OUTPUT","arduino":"f199b929.941438","x":740,"y":
240,"wires":[]}, {"id":"f199b929.941438","type":"arduino-board","z":"","device":"/dev/ttyACM0"}]
```

MONTAGE

Pour le montage, c'est très simple il suffit de mettre l'Arduino en USB avec la Raspberry.

Si vous commander beaucoup d'entrée sortie il sera important d'alimenter l'Arduino et non pas de la laisser alimentée par la Raspberry car l'intensité pourra être trop élevé . Privilégier donc une alimentation et laisser l'USB pour la communication.

Nous commandons la pin10 de l'arduino, mettez donc un relais ou une lampe sur cette sortie, le tour et joué, vous gérer votre effecteur en passant par l'Arduino, tout ça depuis NODE RED.

SUPPLÉMENT REMARQUE

C'est la même procédure pour récupérer une valeur, nous le verrons pour le capteur de présence dans un autre chapitre.

3. Comment gérer un télérupteur avec NODE RED

OBJECTIF

Nous avons pu voir précédemment l'intérêt d'un télérupteur. C'est assez simple de le gérer avec un bouton poussoir, mais c'est du physique. Maintenant il faut pouvoir le gérer avec notre Raspberry. Rassurer vous, c'est très simple !

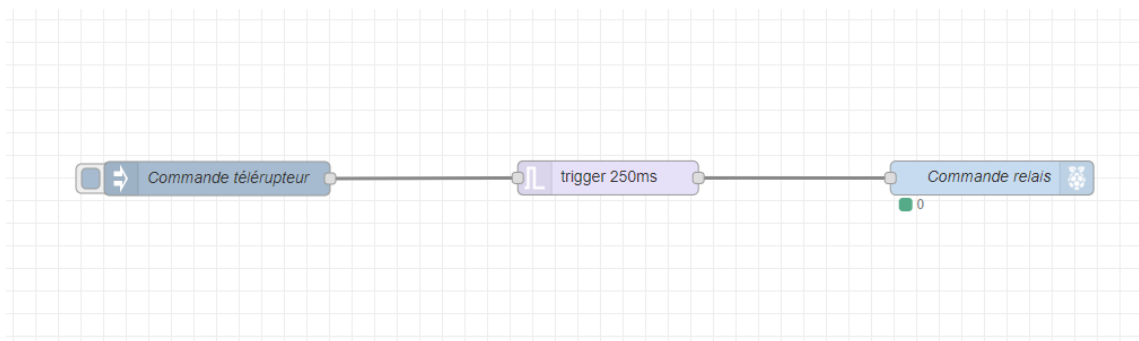
NODE RED

Nous allons utiliser le trigger que nous avons vu dans le chapitre précédent (lire la valeur du PIR Arduino et Raspberry)

En effet le télérupteur ne supporte que des impulsions de commande, c'est exactement ce que fait le trigger puisque lorsque nous allons demander l'allumage de la lumière, le trigger va générer une impulsion, de même lorsque l'on demandera d'éteindre.

Nous verrons dans un autre chapitre comment connaître l'état de la lampe. Vous allez comprendre..

Voilà ce que l'on doit avoir :



En fait c'est simple, on injecte une valeur 1 (peut importe, que l'on veuille allumer ou éteindre, ceci alimente la bobine du télérupteur pendant 250ms grâce au relais (car le télérupteur fonctionne en 24V) et le télérupteur se met donc soit au repos soit au travail. Nous avons tout simplement simulé le bouton poussoir physique.

Nous comprenons là que nous n'avons aucun retour sur l'état de la lampe, ceci fait l'objet d'un autre chapitre.

Le code si besoin

Voici le code si vous en avez besoin :

```
[{"id":"4ec5e1d.a6c062","type":"inject","z":"b07a0d8d.116a8","name":"Commande télérupteur","topic":"","payload":"1","payloadType":"num","repeat":"","crontab":"","once":false,"x":220,"y":360,"wires":[["6358d281.64647c"]]}, {"id":"276ddf6c.777ed","type":"rpi-gpio out","z":"b07a0d8d.116a8","name":"Commande relais","pin":"32","set":true,"level":"0","out":"out","x":906.1888427734375,"y":359.5322256088257,"wires":[[]]}, {"id":"6358d281.64647c","type":"trigger","z":"b07a0d8d.116a8","op1":"1","op2":"0","op1type":"str","op2type":"str","duration":"250","extend":false,"units":"ms","reset":"","name":"","x":566.1819248199463,"y":359.4350576400757,"wires":[["276ddf6c.777ed"]]}]
```

MONTAGE

Le montage est assez simple. on peut voir que le relais se trouve sur la pin32 de la Raspberry, ensuite le relais partie puissance alimente la bobine du télérupteur, ce dernier alimente la lampe.

SUPPLÉMENT REMARQUE

Voir chapitre suivant pour avoir un retour sur l'état de la lampe (ceci peut être important si nous ne sommes pas chez nous et que nous voulons éteindre la lumière par exemple, pour pouvoir l'éteindre, il faut déjà savoir qu'elle est allumé) !

4. Gérer le télérupteur et avoir un retour sur l'état de la lampe

OBJECTIF

remarque : il est nécessaire d'avoir suivi la leçon précédente (gérer un télérupteur avec NODE RED)

L'objectif ici est de connaître l'état de la lampe pour gérer au mieux son état.

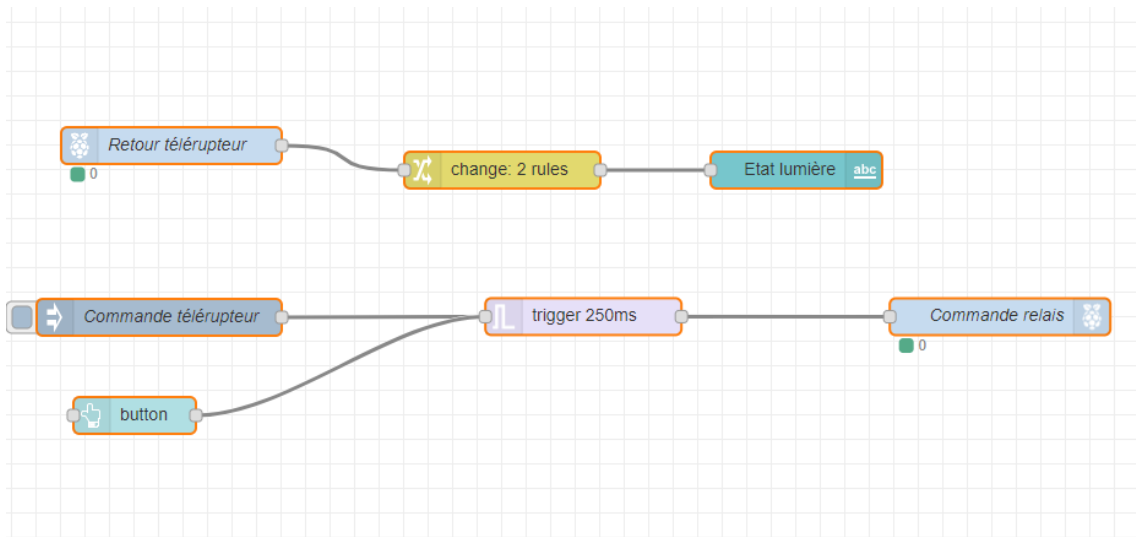
Ce télérupteur est assez pratique puisqu'il contient deux contacts NO, le but va donc d'en utiliser un pour alimenter la lampe (partie puissance) et le deuxième pour alimenter un port Raspberry en 5V (partie commande). Si nous avons 5V c'est que la lampe est allumée car les contacts sont fermés, si nous avons 0V c'est que les contacts du télérupteur sont ouverts et donc que la lampe est éteinte.

Remarque : si nous ne savons pas comment faire pour avoir un dashboard sur NODE RED, vous devez impérativement suivre cette procédure : utiliser le dashboard (cf. *Utiliser le dashboard NODE RED*)^[p. 42]

NODE RED

Reprenez le code précédent, il suffira d'y greffer un bouton sur le dashboard. Ensuite vous allez aller avoir le retour du télérupteur en 5V sur la pin38 du raspberry, vous allez lire cette valeur et la changer en allumée ou éteinte avec un node "change", vous renverrez ce message sur le dashboard. (n'oubliez pas de suivre la procédure donnée dans les objectifs de ce chapitre)

Voilà ce que vous devez avoir :



Le code si besoin

Voici le code si vous en avez besoin :

```
[{"id":"4ec5e1d.a6c062","type":"inject","z":"b07a0d8d.116a8","name":"Commande télérupteur","topic":"","payload":"1","payloadType":"num","repeat":"","crontab":"","once":false,"x":220,"y":360,"wires":[["6358d281.64647c"]]}, {"id":"276ddf6c.777ed","type":"rpi-gpio out","z":"b07a0d8d.116a8","name":"Commande relais","pin":"32","set":true,"level":"0","freq":"","out":"out","x":906.1888427734375,"y":359.5322256088257,"wires":[]}, {"id":"6358d281.64647c","type":"trigger","z":"b07a0d8d.116a8","op1":"1","op2":"0","op1type":"str","op2type":"str","duration":"250","extend":false,"units":"ms","reset":"","name":"","x":566.1819248199463,"y":359.4350576400757,"wires":[["276ddf6c.777ed"]]}, {"id":"720f5de7.aa5764","type":"rpi-gpio in","z":"b07a0d8d.116a8","name":"Retour télérupteur","pin":"38","intype":"tri","debounce":"25","read":false,"x":230,"y":220,"wires":[["25d065ec.7cd00a"]]}, {"id":"9b9397ae.40b658","type":"ui_text","z":"b07a0d8d.116a8","group":"deeb80d0.760fa","order":0,"width":0,"height":0,"name":"Etat lumière","label":"Etat lumière","format":"{msg.payload}","layout":"row-left","x":740,"y":240,"wires":[]}, {"id":"25d065ec.7cd00a","type":"change","z":"b07a0d8d.116a8","name":"","rules":[{"t":"change","p":"payload","pt":"msg","from":"0","fromt":"num","to":"éteinte","tot":"str"}, {"t":"change","p":"payload","pt":"msg","from":"1","fromt":"num","to":"allumée","tot":"str"}],"action":"","property":"","from":"","to":"","reg":false,"x":500,"y":240,"wires":[["9b9397ae.40b658"]]}, {"id":"216baac1.1d95a6","type":"ui_button","z":"b07a0d8d.116a8","name":"","group":"deeb80d0.760fa","order":0,"width":0,"height":0,"passthru":false,"label":"button","color":"","bgcolor":"","icon":"","payload":"1","payloadType":"num","topic":"","x":200,"y":440,"wires":[["6358d281.64647c"]]}, {"id":"deeb80d0.760fa","type":"ui_group","z":"","name":"Etat lumière","tab":"a811bdcd.f4953","order":1,"disp":true,"width":"6"}, {"id":"a811bdcd.f4953","type":"ui_tab","z":"","name":"Maquette domotique","icon":"dashboard","order":1}]
```

MONTAGE

Pour ce montage vous devez réaliser le même que précédemment, il faut rajouter une liaison en 5V sur le télérupteur qui alimentera la pin38.

SUPPLÉMENT REMARQUE

Vous pouvez maintenant gérer votre lampe soit avec votre bouton poussoir physique, soit le bouton poussoir de NODE RED, tout en connaissant son état à tout moment !

Chapitre 4

Annexes

1. Toutes mes procédures

1.1. Ajouter une Arduino sur NODE RED

- 1 Brancher votre Arduino à la Raspberry par USB
- 2 Lancer l'IDE Arduino comme si vous alliez faire un programme Arduino

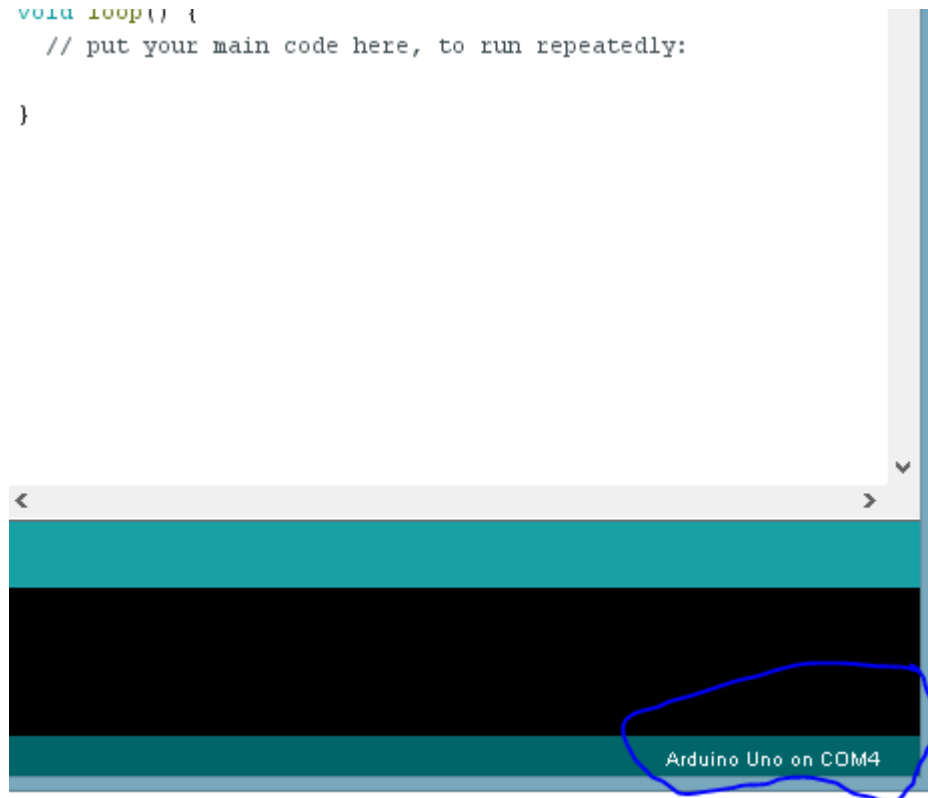
En bas vous allez en fait pouvoir lire sur quel port est branché l'Arduino

Sur un windows vous avez "Arduino en COM4"

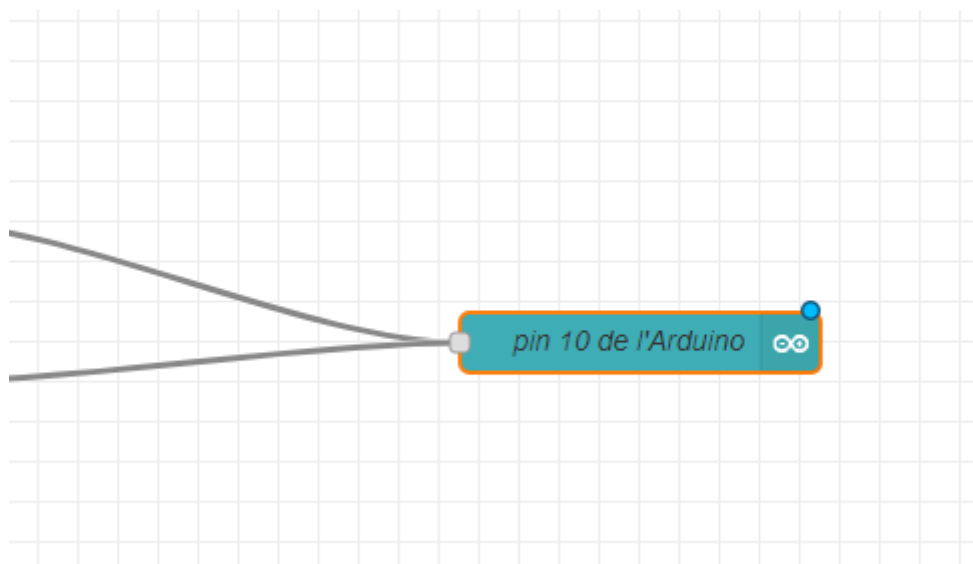
Sur un linux vous aurez Arduino on "dev/ttyACM0"

Pour lire cette information il faut regarder en bas de la fenêtre de l'IDE Arduino. Voilà maintenant vous savez sur quel port de la Raspberry votre Arduino est branchée.

```
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```



- 3 Cliquer sur le node Arduino que vous avez pu insérer précédemment.



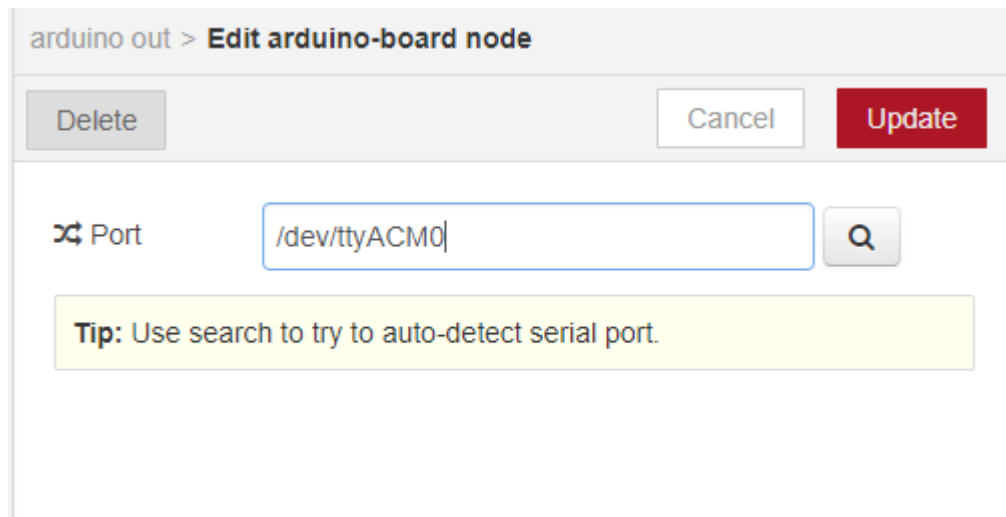
- 4 Vous avez une fenêtre qui apparaît cliquer sur l'onglet entouré pour accéder au paramétrage d'une nouvelle carte Arduino

The screenshot shows a dialog box titled 'Edit arduino out node'. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below these is a section titled 'node properties' with a dropdown arrow. The properties are listed as follows:

- Port:** A text field containing '/dev/ttyACM0'. To its right is a small icon of a USB cable, which is circled in blue.
- Type:** A dropdown menu showing 'Digital (0/1)'.
- Pin:** A text field containing '10'.
- Name:** A text field containing 'pin 10 de l'Arduino'.

At the bottom of the dialog, there is a yellow note box with the text: **Note:** You cannot use the same pin for both output and input.

- 5 Entrer le port que vous avez obtenu à l'étape où il fallait lire le port sur l'IDE



Puis cliquer sur update

Voilà vous avez ajouter le port usb où est connecté l'Arduino

6 Revenir sur l'IDE arduino

Vous devez simplement charger un programme dans l'Arduino pour que la communication se fasse :

Aller dans file ou fichier

Aller dans exemples ou exemple

Aller dans Firmata puis cliquer sur StandartFirmata

Compiler le programme, en même temps vérifier que le port est toujours le même que précédemment sinon changer le port dans NODE RED

Voilà, une fois ce programme chargé vous pourrez interagir avec votre Arduino.

1.2. MISE EN ROUTE ARDUINO

Cette procédure vous permet de mettre en oeuvre un petit programme et d'avoir un (très) petit aperçu de ce que l'on peut faire avec Arduino

Lien WEB très intéressant ![\[https://openclassrooms.com/courses/programmez-vos-premiers-montages-avec-arduino\]](https://openclassrooms.com/courses/programmez-vos-premiers-montages-avec-arduino)

✓ Prérequis

Il faut pour suivre cette procédure posséder une carte Arduino (Uno), avoir installé le logiciel Arduino et si possible quelques câbles une résistance de 300Ohm et une LED afin d'effectuer un premier branchement.

1 Lancer l'IDE Arduino

IDE

2 Copier le programme ci-dessous

```
int LED=10; // la pin 10 du côté digital va allumer ou non la LED
void setup() {
  pinMode(LED,OUTPUT); //pin10 en mode OUTPUT (sortie)
}
void loop() { // boucle
  digitalWrite(LED,HIGH); // la LED s'allume
  delay(1000); // on attend une seconde
  digitalWrite(LED,LOW); // la LED s'éteint
  delay(1000); // on attend une seconde
}
```

3 Faire le branchement

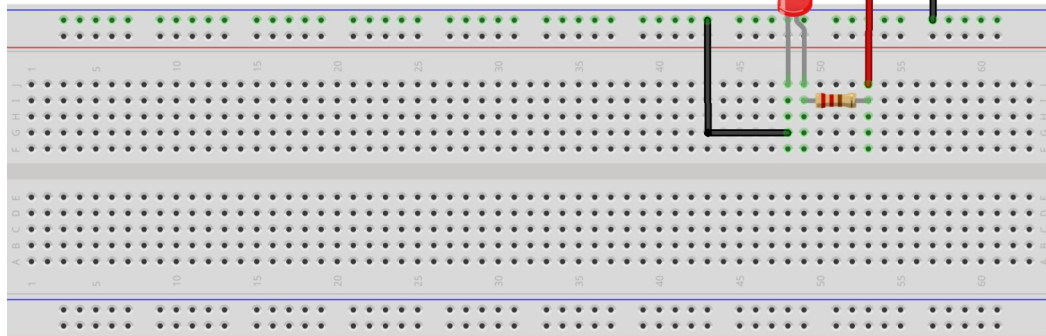
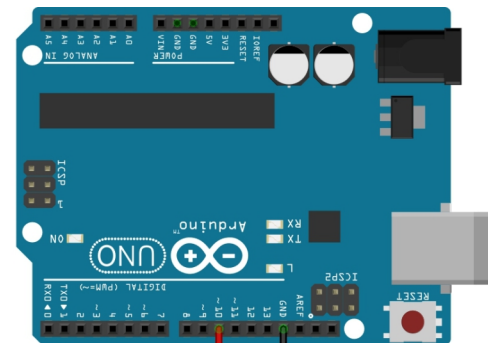
Branchement :

(montage fait sur le logiciel Fritzing)

Le montage est très simple.

à noter que $R=300\Omega$

Lorsque l'on enverra un signal 1 sur la pin 10 la LED s'allumera. Lorsque l'on enverra un signal 0 sur la pin 10 la LED s'éteindra.



fritzing

4 Explication du code !

1. On déclare les variables, il y a plusieurs type de variable, ici c'est un entier donc on déclare cette variable avec "int", puis on lui donne un nom : LED. On dit que l'on branche notre led sur la broche10 de l'arduino, côté digital (1/0) et non analogique (valeur allant de 0 à 1024)
2. le setup : c'est ici que l'on pose les bases pour la suite du programme, on va ici déclarer que notre broche 10 est une sortie, on va envoyer une information dessus, et ce n'est pas l'Arduino qui va recevoir une information
3. loop : c'est la boucle, c'est le cœur du programme. le programme s'exécute puis revient au départ et recommence et ainsi de suite.

on met la sortie 10 à l'état 1 (HIGH=haut=1) puis on dit d'attendre 1000 millisecondes, soit 1 seconde avec la fonction delay() ; puis on met la sortie 10 à l'état 0 et on attend encore une fois 1 seconde. Le programme tourne en boucle jusqu'à ce que l'on débranche l'Arduino ou que l'on téléverse un autre programme.

5 Téléversement

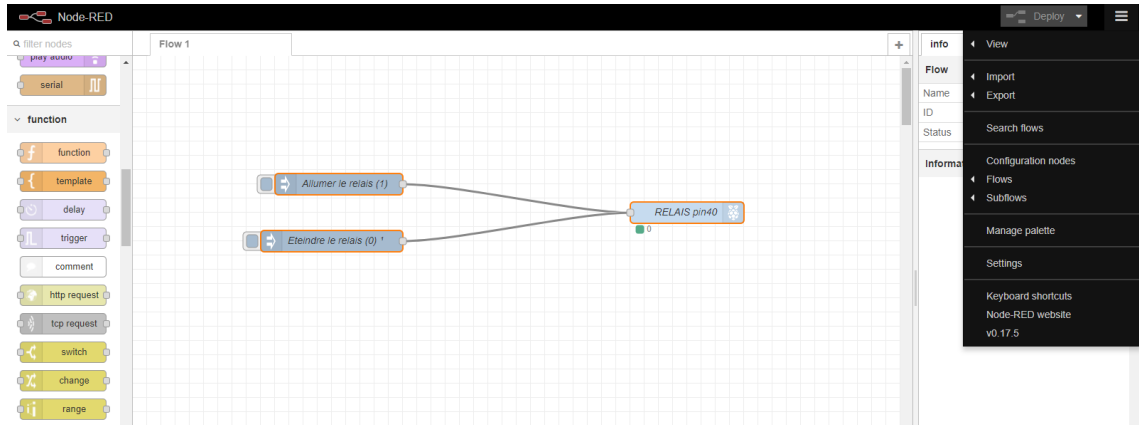
Après avoir fait le branchement et fait le programme,

Il suffit de brancher l'Arduino sur l'ordinateur grâce à son câble, en haut à gauche il y a un premier bouton vérifier le programme afin de s'assurer qu'il n'y a pas d'erreur, on peut lire en bas les messages d'erreur. Le deuxième bouton -> sert à téléverser le programme dans la carte, on va en fait mettre le programme dans la carte. Le programme reste dans la carte même si on l'arrête puis la redémarre ! Attention de pas brancher n'importe quoi car il pourrait y avoir des courts circuits dû à un programme déjà existant.

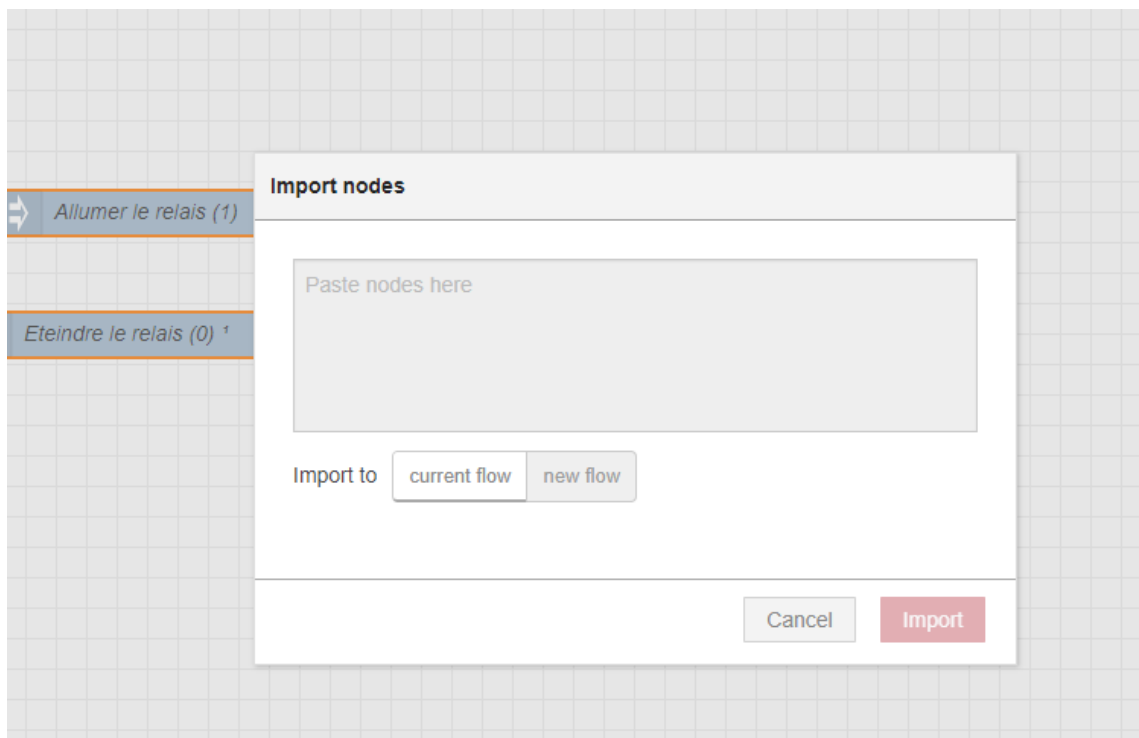
1.3. Comment importer et exporter des Clipboard

- 1 Pour importer un clipboard.

Aller dans le menu en haut à gauche



cliquer sur import puis sur clipboard

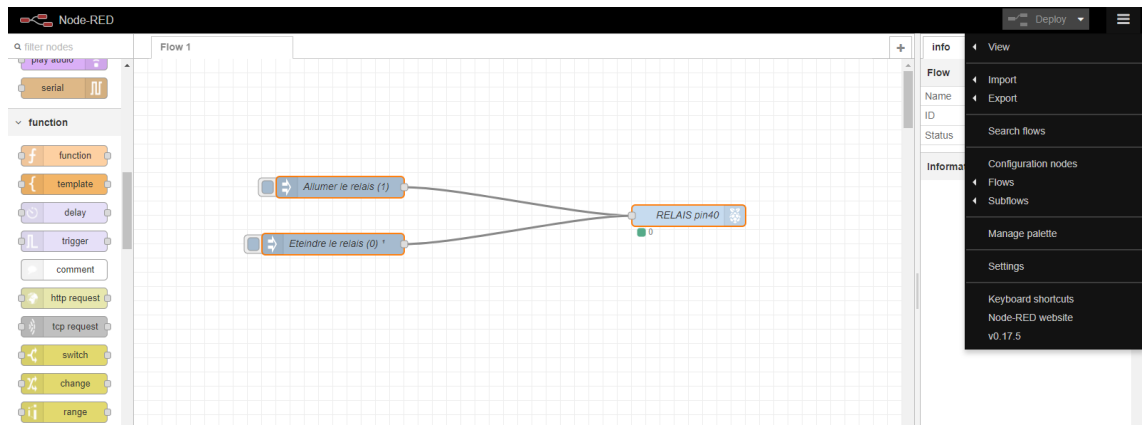


Voici pour vous entraîner :

```
[{"id":"3028667f.3f040a","type":"inject","z":"b07a0d8d.116a8","name":"Eteindre le relais (0)","topic":"","payload":"0","payloadType":"num","repeat":"","crontab":"","once":true,"x":280,"y":260,"wires":[["f2ae24b3.15e278"]]}, {"id":"d9a2e314.1ad37","type":"inject","z":"b07a0d8d.116a8","name":"Allumer le relais (1)","topic":"","payload":"1","payloadType":"num","repeat":"","crontab":"","once":false,"x":290,"y":180,"wires":[["f2ae24b3.15e278"]]}, {"id":"f2ae24b3.15e278","type":"rpi-gpio out","z":"b07a0d8d.116a8","name":"RELAIS pin40","pin":"40","set":"","level":"0","freq":"","out":"out","x":780,"y":220,"wires":[]}]
```

Copier votre code dans cet espace réservé et appuyer sur import

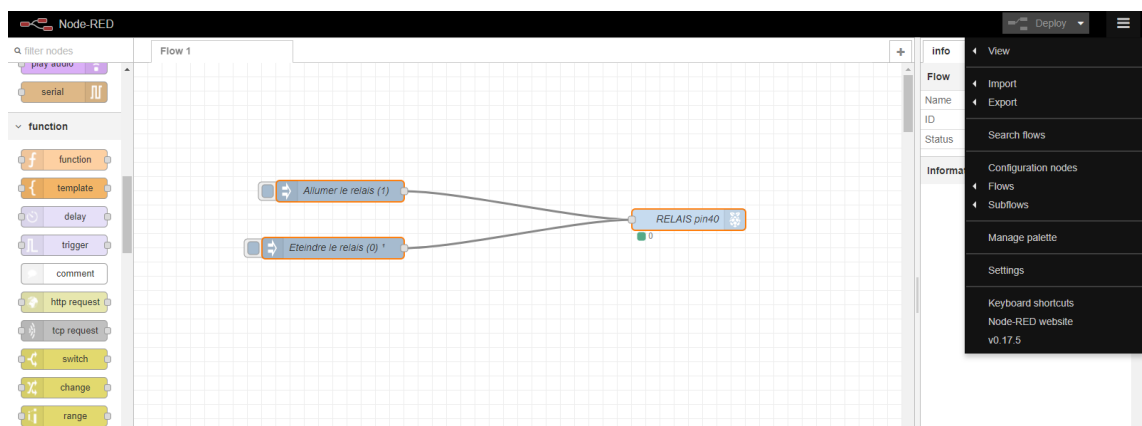
Voilà la transcription du code apparaît.



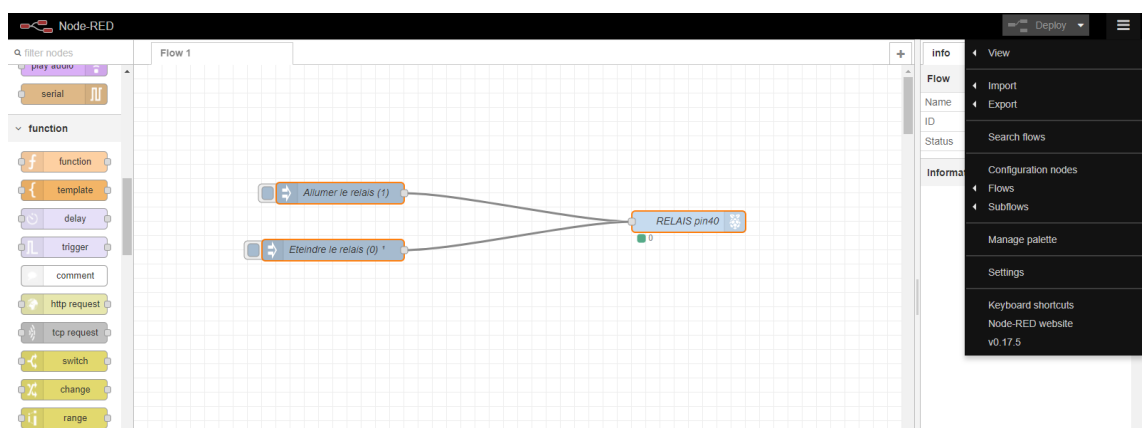
Remarque : pour stocker vos codes vous disposer également d'une librairie dans NODE RED, c'est un peu votre dossier de stockage, il suffit de cliquer sur librairie pour les stocker dans ce dossier, vous le nommerez comme vous le voulez, pour le réimporter il suffit de suivre la procédure importer un clipboard, mais d'aller dans librairie et de sélectionner votre fichier.

2 Exporter du code

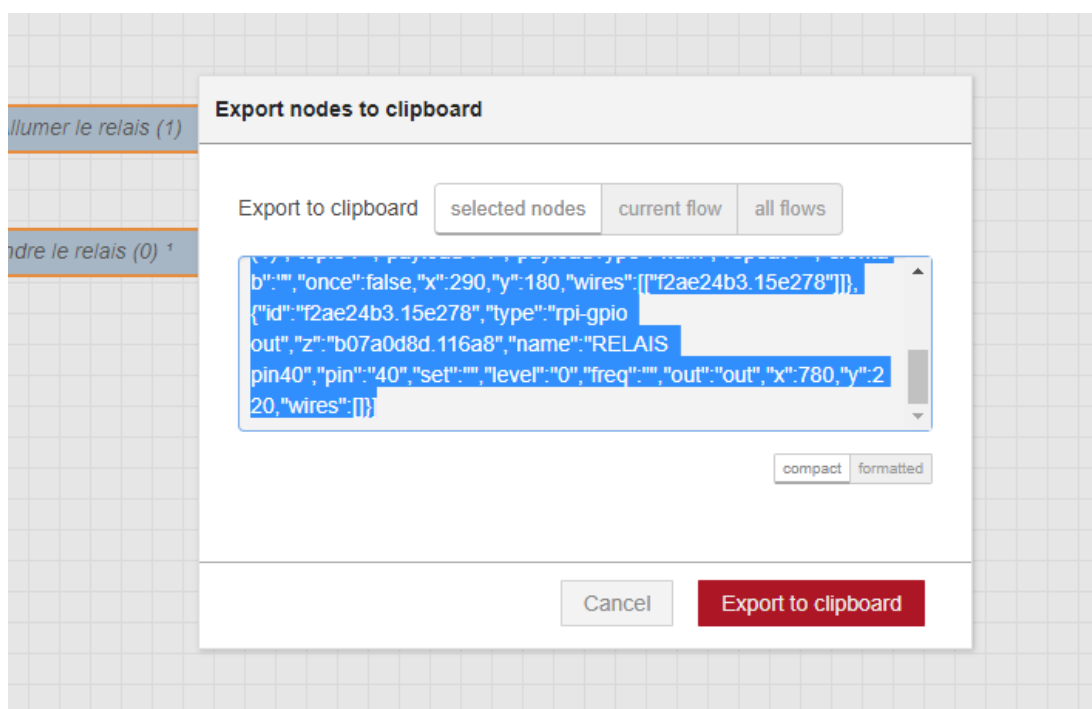
Sélectionner le code à exporter avec la souris



Aller dans le menu en haut à gauche



Cliquer sur export puis clipboard

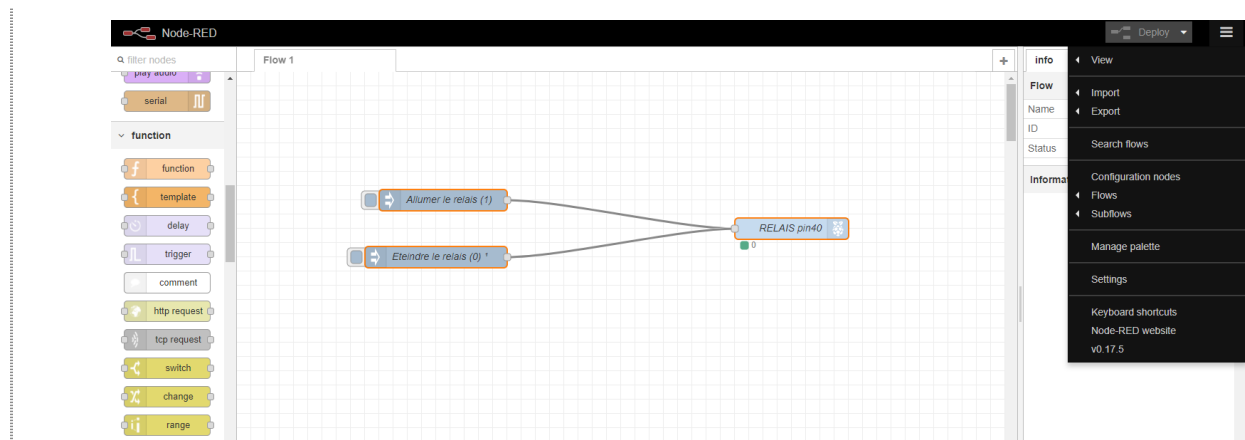


Voilà vous obtenez un code à copier que vous pouvez transmettre à un collègue ou que pouvez stocker.

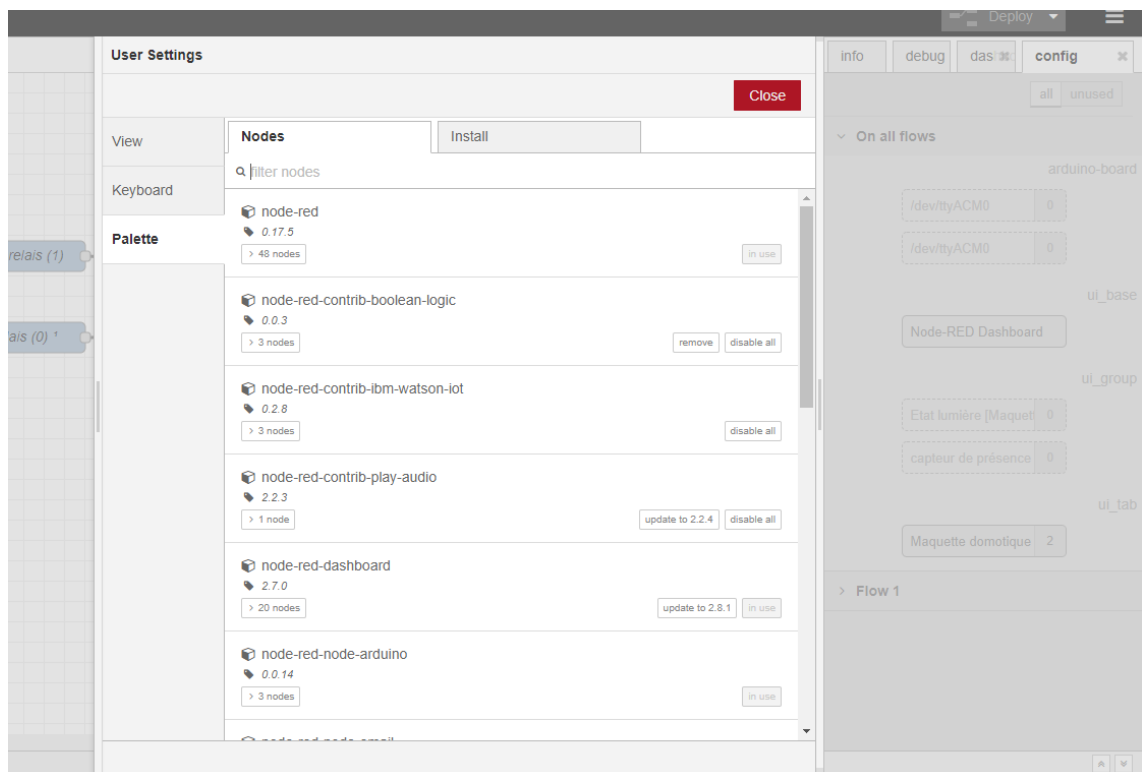
Remarque : pour stocker vos codes vous disposer également d'une librairies dans NODE RED, c'est un peu votre dossier de stockage, il suffit de cliquer sur librairie pour les stockers dans ce dossier, vous le nommerez comme vous le voulez, pour le réimporter il suffit de suivre la procédure importer un clipboard, mais d'aller dans librairies et de sélectionner votre fichier.

1.4. Ajouter des nœuds sur NODE RED

- 1 Allumer NODE RED, puis aller dans le menu de gauche



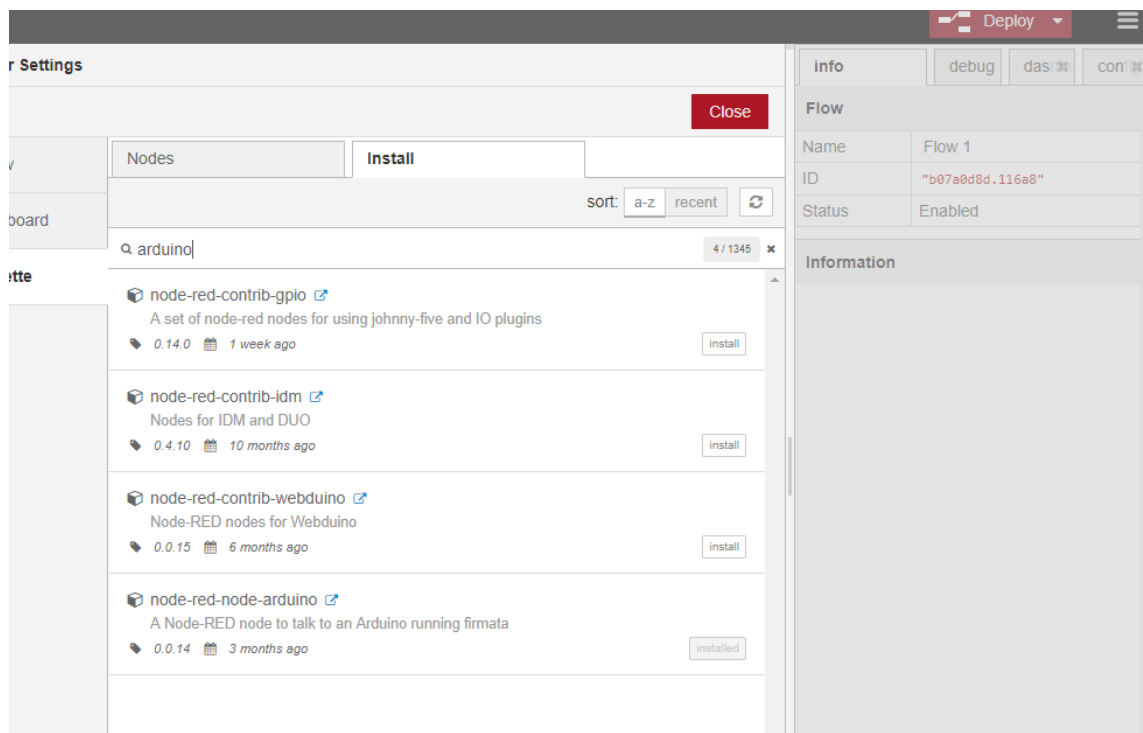
- 2 Cliquer sur manage palette



Sur l'onglet nodes (nœuds en français) on peut voir ceux que l'on possède déjà

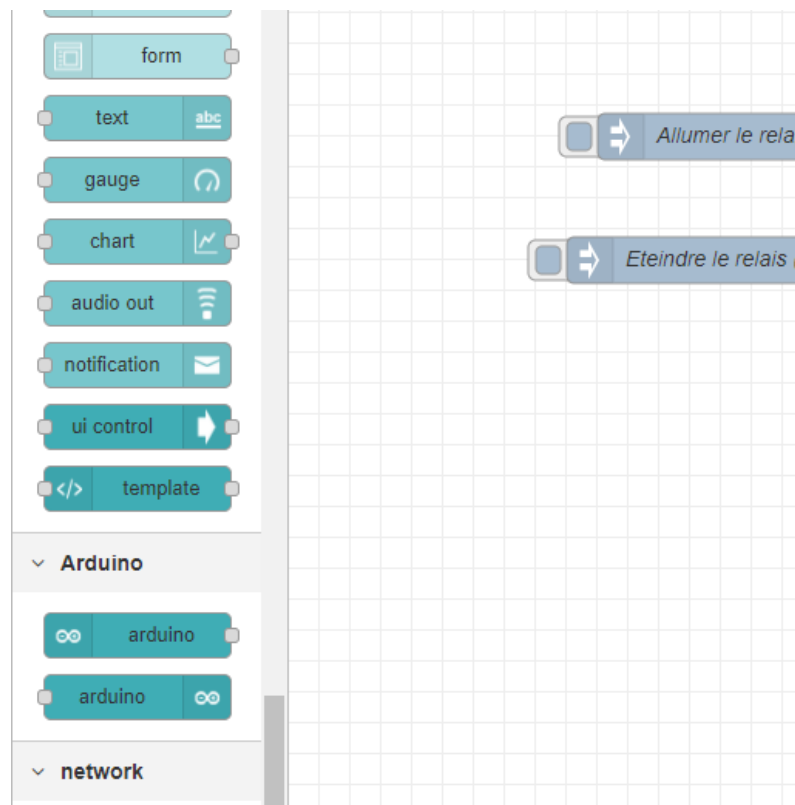
Sur l'onglet install c'est pour rechercher des nœuds existants que l'on voudrait avoir

- 3 Cliquer sur install et rechercher Arduino



On peut voir sur le côté que l'on peut installer les différents modules, vous pouvez par exemple ajouter celui que j'ai déjà qui est utile pour communiquer avec une Arduino, c'est la V 0.0.14

- 4 Attendre que les modules s'installent et ils apparaîtront dans votre liste de nodes automatiquement.



- 5 Voilà, vous pouvez désormais importer de nouveaux nœuds...

Vous pourrez les désinstaller en allant dans Manage palette et dans nodes (ceux installé) il suffit de cliquer sur disable all, si vous ne pouvez pas c'est simplement que vous les utiliser à un endroit

..... dans NODE RED.

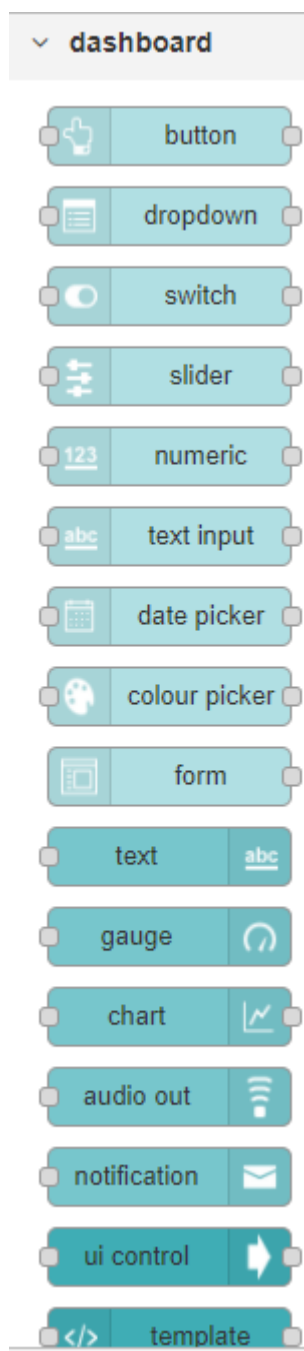
1.5. Utiliser le dashboard NODE RED

Nous allons voir comment gérer un dashboard, il y a un site internet très bien fait qui explique également cette démarche, voici le lien pour ceux que ça intéresse. Il y a notamment une explication pour personnaliser un petit peu plus son interface : lien[<https://projetsdiy.fr/node-red-dashboard-interface-partie1/>]

Prérequis

Télécharger les nodes dashboard. Moi j'ai la version 2.7.0

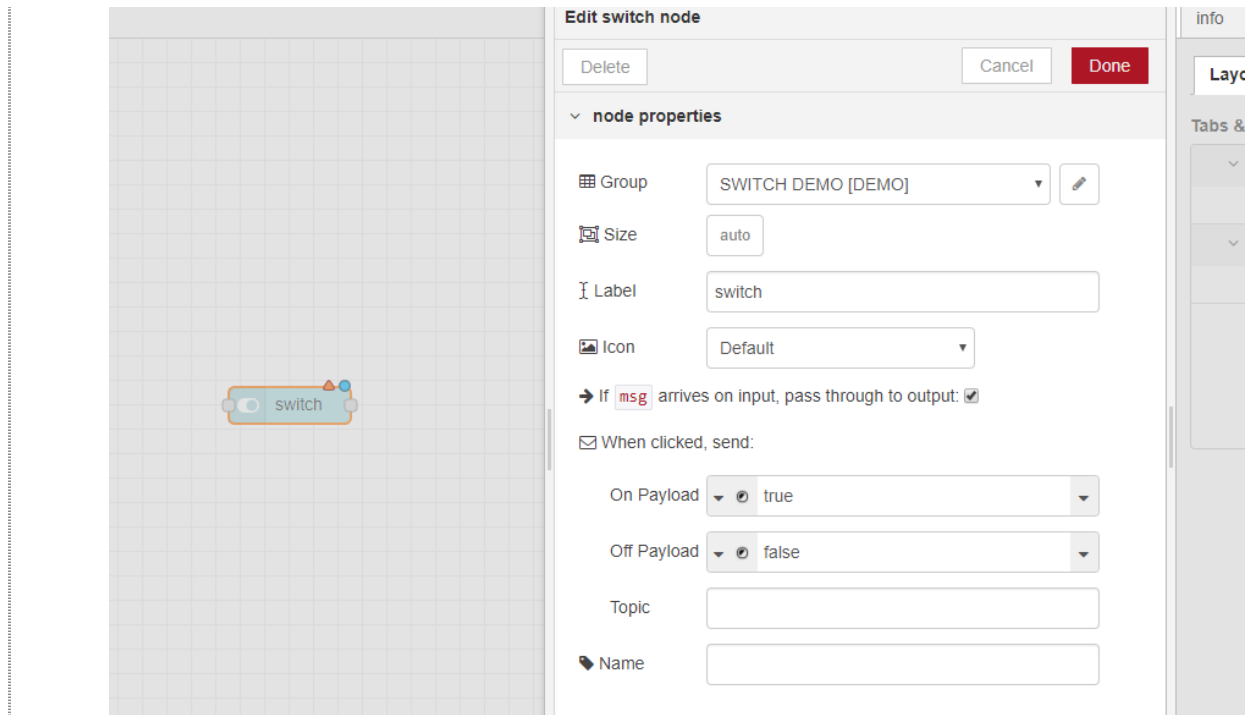
Voilà les nodes que vous devez avoir :



1 Créer un switch

Mettre un switch

Nous allons d'abord créer un switch, puis nous l'utiliserons pour afficher soit un message soit l'autre. Mettre un switch et cliquer dessus pour le paramétrer :



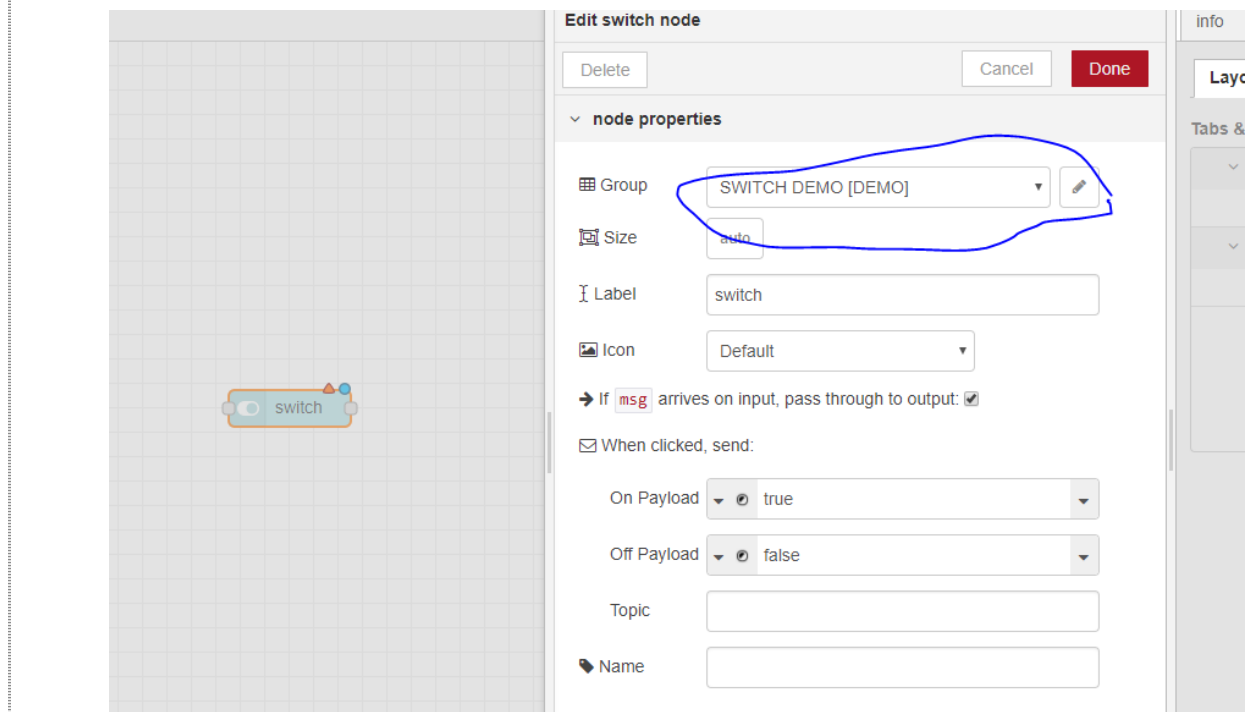
2 Paramétrer le switch

Vous devez commencer par indiquer dans quel "fichier" se range le bouton, vous pouvez créer n'importe quoi, par exemple : MAISON, Pièce 5

Un autre switch sera dans MAISON, pièce 2

et l'autre dans : CABANE, établi

Vous avez compris ? C'est à vous ! Vous devez obtenir ceci :



3 Paramétrer ce que le switch envoie

Lorsque l'on est d'un côté du switch il envoie 1, lorsque l'on est de l'autre on envoie 0

☑ When clicked, send:

On Payload

Off Payload

Topic

📌 Name

4 Accéder au switch

Tout d'abord il faut déployer (deploy) Si il y a un souci, il faut revoir le paramétrage. Le paramétrage est très simpliste pour ne pas faire d'erreur

Ensuite aller à l'adresse de votre NODE RED mais ajoutez /ui/ au bout, comme ceci :

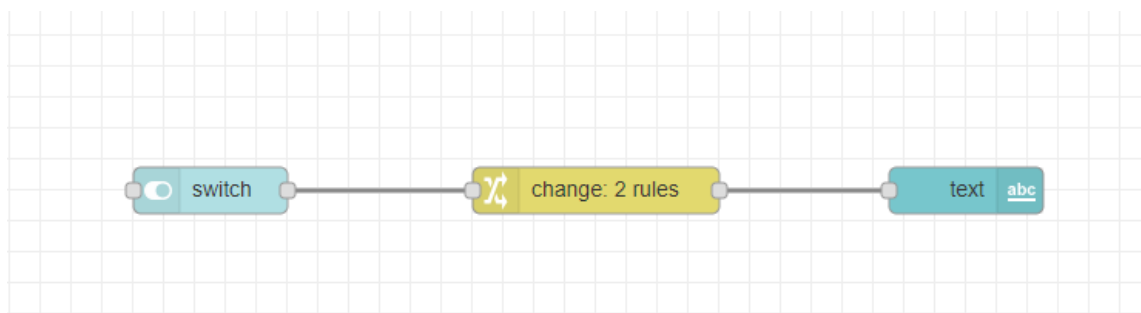
📄 192.168.45.152:1880/ui/

Vous allez être dirigé vers une page internet qui va vous ouvrir une interface ! Il y a désormais votre switch, vous pouvez le bouger. Sympa ! Si votre téléphone est connecté au même réseau (Wifi par exemple) que votre ordinateur ou votre raspberry vous pouvez également accéder au switch avec cette même adresse !

5 Même principe mais dans l'autre sens, c'est le programme qui interagit avec nous via l'ui

Avec ce switch vous commandez un node qui vous transforme le 0 ou 1 en ON ou OFF, et le node text va vous faire apparaître ça dans l'UI. (utilisation du node CHANGE)

Voilà ce que l'on obtient :



Sur le dashboard :



6 C'est à vous !

Vous avez compris comment ça marchais ? C'est à vous, il y a plein de node très intéressant comme les graphique, les boutons....Amusez vous à les tester ils pourraient vous servir !

2. Fichier utile

Fichier qui peut vous être utile !

Mon compte rendu de projet GIM 1 2016/2017

Vous pouvez consulter ce document que j'avais fait, on peut y trouver des éléments très intéressant, ne lisez pas tout mais sachez au moins ce qu'il contient, cela vous permettra de retrouver des infos si besoin.  [compte rendu gim1 projet.pdf](#) (cf. - *compte rendu gim1 projet.pdf*)