**Domotique : pilotez une lampe grâce aux relais**

Vous voici au dernier chapitre de ce cours. Comme promis, nous allons aborder les notions de domotique, c'est-à-dire la possibilité de piloter des matériels fonctionnant sur le courant fourni par EDF.

Vous aurez besoin pour ce chapitre de :

* Un relais qui est piloté par 5V, et qui, en contact, supporte 5A et 220V AC (courant alternatif). Voici le relais que j'utilise ([JQX-14FC-2C](http://www.sah.rs/SklopnaTehnika/RELAY/SHENLER/JQX-14FC-2C_eng.html)) qui a l'avantage de pouvoir se fixer sur une breadboard.
* Une LED avec sa résistance de 220Ω.
* Une diode type 4N4007.
* Un transitor NPN (et une résistance de 1kΩ) ou un MOSFET canal N (et une résistance de 10kΩ).

Je vais aborder avec vous la théorie et je vous expliquerai les montages. Pour des raisons de sécurité, ces montages seront d'abord réalisés avec des composants fonctionnant sous 5V.

Je vous indiquerai après (en insistant sur les précautions à prendre) comment modifier le circuit pour réaliser une petite carte à piloter qui pourra utiliser du 220V. C'est ce que j'appelle le test en réel. Étant donnés les risques liés à ce type de montage, n’oubliez pas que la responsabilité tient entièrement entre vos mains ! (et je vous le dirai à nouveau plus loin)

Si vous souhaitez passer au test réel (avec du courant domestique), il vous faudra :

* Une rallonge simple, câble 2 x 0,75 mm²(mâle d'un côté, femelle de l'autre) pour vous brancher sur une prise de votre maison (cette rallonge devra être coupée pour être utilisée) ;
* Des [dominos](https://www.adafruit.com/products/725) pour connecter les fils, qui seront ensuite soudés sur plaque d'essai à pastilles ;
* Une [plaque d'essai à pastille](http://www.selectronic.fr/plaque-bakelite-pastillee-200x100mm.html) et du matériel de soudure.

Une fois ces montages réalisés, nous pourrons voir comment créer une interface simple pour piloter une lampe de chevet depuis votre réseau.

Bon, je pense qu'on peut y aller !

**Le circuit de commande et le circuit de puissance**

Nous avons vu dans le cours d'initiation que malgré toutes ses possibilités, l'Arduino n'est pas indestructible, et qu'il n'est pas non plus le roi pour fournir de l'énergie électrique. Ainsi, lorsqu'il s'agit de faire tourner un moteur, la carte Arduino devait être secondée en puissance électrique et protégée de tout risque de "roue libre".

Vous avez donc appris à [séparer le circuit de commande du circuit de puissance](https://openclassrooms.com/courses/programmez-vos-premiers-montages-avec-arduino/le-moteur-a-courant-continu-partie-1-transistors-et-sorties-pwm#/id/r-3285327). Petit rappel :

* Le circuit de commande est en fait la carte Arduino. Elle va, grâce à du courant envoyé par ses pins, pouvoir déclencher des passages de courants, en pilotant soit des transistors, soit des MOSFET.
* Le circuit de puissance, lui, est le circuit dans lequel un courant plus important circule. Il alimente les charges (moteurs, lampes...) et est passif, c'est-à-dire qu'il ne pilote rien, mais se fait piloter. Donc si le circuit de commande interdit le passage du courant dans le circuit de puissance, ce dernier ne peut qu'obéir.
* Cette séparation nécessite des composants de pilotage : le transistor ou le MOSFET. J'explique le fonctionnement de ces deux composants dans le cours d'initiation.

Si, lorsqu'il s'agit de piloter un moteur, les risques sont de griller la carte ou que le moteur ne démarre pas, il n'y a rien d'alarmant.

Lorsqu'il s'agit de courant domestique (le 220V de votre maison), les risques sont bien plus importants, puisqu'ils concernent des dommages éventuels sur le corps humain, souvent irréversibles (électrisation, brûlures voire électrocution !), et des risques de courts-circuits pouvant déclencher un incendie.

(Oui Lukas, vous faites bien de vous inquiéter, avec beaucoup de précautions !)

Je dirai donc :

Si vous décidez de manipuler le courant domestique pour réaliser les montages qui suivront, vous devrez le faire en prenant toutes les précautions que je vous indiquerai et en toute connaissance des risques éventuels. Vous êtes seul responsable de vos manipulations. OpenClassrooms et moi-même ne pourrons être tenus pour responsables de tout accident (corporel ou domestique) survenu dans le cadre de vos expériences.

Ben du coup on ne peut pas suivre ce chapitre ?!

Bien sûr que si ! Tout d'abord, les montages que je vous propose ne concernent pas uniquement les circuits liés au 220v de votre maison. Ils sont tout à fait utiles pour piloter des charges sur du 12V et 24V, tout en protégeant l'Arduino.

Ensuite, si vous êtes adulte (donc responsable) il est tout à fait possible de réaliser ces montages en prenant les risques en considération, les précautions nécessaires et en utilisant le matériel qui convient.

Pour en revenir à notre question de circuit de commande et circuit de puissance, le pilotage de moteurs que l’on avait vu dans le cours d’initiation nous permettait de bien comprendre le principe. En revanche, nous faisions une chose qu'il est complètement interdit de faire lorsqu'on manipule du courant domestique.  Voici la règle que nous ne respections pas :

Lorsqu'il s'agit de courant domestique, il est obligatoire que les deux circuits soient complètement séparés. Les alimentations doivent être séparées, ainsi que les connexions au ground (à la terre) qui doivent se faire sur deux connections différentes.

Or nous avions un ground commun. Il nous faut donc organiser notre montage différemment. De plus, un transistor ou MOSFET n'est pas considéré comme un composant qui sépare correctement les deux circuits.

Il existe donc un matériel très efficace pour réaliser ce travail correctement : le **relais**.

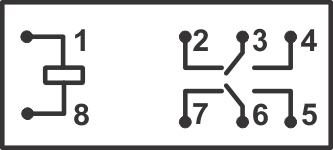
On peut aussi utiliser des optocoupleurs qui utilisent la lumières pour activer ou désactiver une connexion (donc aucun contact entre les deux circuits) mais ils ne feront pas l'objet de ce cours.

**Le relais**

Voici une image du relais que j'utilise pour réaliser ce chapitre. Il a deux avantages : il peut se connecter sur une breadboard (ou être facilement soudé sur une planche à pastille) et est transparent. Vous pouvez, du coup, mieux voir le mécanisme :

Relais JQX-14FC-2C

Lorsqu'on l'observe, on voit un petit schéma sur le côté du relais. Le voici en plus lisible :

Schéma de connexion du JQX-14FC-2C

Il s'agit en fait du schéma des connexions du relais.

Lorsque vous achetez un relais, il est important de savoir comment le connecter. La première remarque que l'on peut faire est que l'on voit bien la séparation de deux (voire trois) circuits :

* Le circuit qui concerne les entrées 1 et 8 : c'est ici que le relais se commande, il fera donc partie du circuit de commande.
* Le circuit qui concerne les pins 2,3 et 4 : qui fera partie du circuit de puissance. Les pins 5,6 et 7 fonctionnent en parallèle. Du coup un relais peut piloter deux circuits séparés.

Mais comment fonctionne ce relais, il y a forcément un courant qui y passe ?

(Vous voici de retour Cunégonde, et pile au bon moment !)

Oui, du courant passe, mais tout l'intérêt du relais se situe dans l'espèce de tube blanc que l'on voit sur l'image.

En fait, dans ce tube est situé une bobine qui peut actionner un électro-aimant (là encore, vous trouverez les explications dans le cours d'initiation). Lorsque cet électro-aimant va fonctionner, il va entraîner un petit mécanisme de bascule qui va permettre de créer une connexion entre les entrées 3 et 4 ou 6 et 7. Donc :

* Au repos, le courant peut passer entre les entrées 2 et 3 ou 5 et 6 ;
* En action, le courant peut passer entre les entrées 3 et 4 ou 6 et 7.

Il suffira donc de connecter judicieusement votre charge (moteur, ampoule, diode...) pour qu'elle s'allume (ou s'éteigne) lorsqu'on active le relais.

Oui mais j'insiste... Le courant qui passe dans la bobine de l'électro-aimant, comment ne touche-t-il pas le reste du circuit ?

C'est tout le principe de l'électro-aimant ! La bobine entoure un noyau de fer doux, mais sans aucun contact entre l'un et l'autre ! Le passage du courant dans la bobine magnétise le fer (sans avoir à le toucher) qui devient un aimant qui déclenche la bascule.

Il est d'ailleurs assez intéressant de regarder le mécanisme en action grâce à un relais à boîtier transparent ! C'est hypnotique, tic, tic, tic....

Le mécanisme du relais en action crée un bruit (un petit tic) très particulier qui permet d'entendre le contact se faire.

C'est d'ailleurs l'une des limites du relais. En effet, du fait de sa partie mécanique, un relais est considéré comme ayant une durée de vie limitée dans le temps. Rassurez-vous, c'est de l'ordre de milliers de permutations, mais c'est une limite tout de même.

Voyons donc comment brancher tout ceci !

(Lukas, rassurez-vous et descendez de ce tabouret, on ne va pas encore sur du courant domestique...)

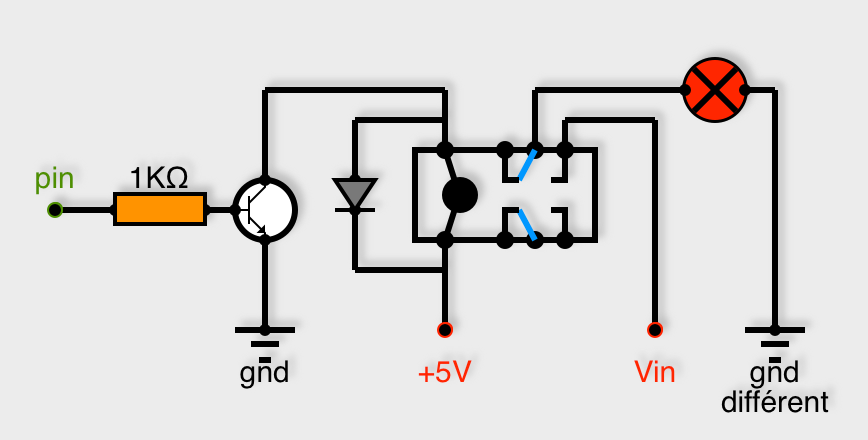
**Connectez votre relais**

Nous n'allons pas connecter directement le relais à notre carte Arduino. En effet, comme tous les composants à bobine, il y a risque de courant non contrôlé qui peut abîmer le pin de commande de l'Arduino. Nous allons donc utiliser la même technique que pour piloter un moteur :

L'Arduino va piloter un transistor (ou MOSFET) qui va piloter le relais qui va piloter le circuit de puissance.

Une chaîne de commande très obéissante (presque comme dans l'armée) contrôlée par un programme que vous aurez conçu.

Voici le schéma électrique qui utilise un transistor NPN :

Circuit pour piloter un relais avec un transistor NPN

Si vous avez des difficultés pour la connexion du transistor NPN, vous pouvez vous référer au [cours d'initiation](https://openclassrooms.com/courses/programmez-vos-premiers-montages-avec-arduino/le-moteur-a-courant-continu-partie-1-transistors-et-sorties-pwm#/id/r-3285313).

Quelques points sur ce schéma de montage :

* La diode aux bornes du relais permet d'éviter des retours de courant non contrôlés, elle protège donc le transistor.
* "Vin" correspond à l'alimentation du circuit de puissance.
* Le +5V de l'Arduino peut-être utilisé pour alimenter le relais.
* Il est préférable de positionner la charge (ici une lampe) du côté du ground (le neutre pour un circuit alternatif), et donc de couper, grâce au relais, l'arrivée du courant (la phase pour un circuit alternatif).

Voici un programme très simple qui permet de faire clignoter la lampe :

int pinRelais=7; //variable pour le pin qui commande le relais

void setup() {

pinMode(pinRelais,OUTPUT); //pin en mode OUTPUT

}

void loop() {

digitalWrite(pinRelais,HIGH); // on commande l'allumage

delay(1000);

digitalWrite(pinRelais,LOW); // on commande l'extinction

delay(1000);

}

Je vous conseille, avant de passer à des tensions plus importantes, de tester ce circuit en connectant une LED (et sa résistance) à la place de la charge (la lampe). Vous atténuerez  ainsi les risques liés à de mauvaises connexions (pour le transistor, pour le relais et pour la charge) et donc les éventuels courts-circuits.

**Passez sur le courant domestique (facultatif)**

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

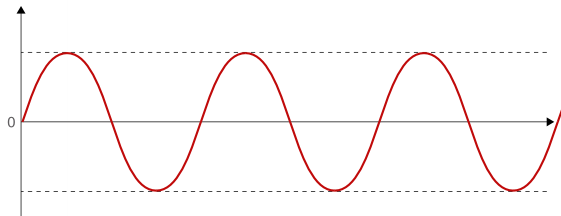
**Le 220V alternatif**

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.

(Lukas, même en allant chercher dans le dictionnaire, c'est un peu long à comprendre...)

Voici une représentation d'une telle courbe :

Courbe sinusoïdale

Le courant oscille donc entre la valeur maximum et la valeur minimum continuellement.

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.

Prenons l'exemple d'un imbécile... (non Lukas, je ne vous regarde pas !)

Si un imbécile touche 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 que j'enlève mon doigt ?!

(Et bien non Lukas ! 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 ! https://openclassrooms.com/bundles/common/images/smiley/waw.png

Mais lorsque l'imbécile est  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 cet imbécile sera électrocuté (c'est-à-dire mort).

(Non Lukas, on ne cherche pas à tirer ou pousser quelqu'un qui est électrisé, car sinon, on s'électrise soi-même, ce qui fait deux imbéciles...).

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.

**Les normes électriques**

Lorsque vous bidouillez votre circuit électrique, sachez qu'il faut respecter certaines normes. Ces normes sont éditées chaque années, et peuvent faire l'objet d'un contrôle par un consuel par exemple.

De plus, lorsqu'il s'agit d'appareils domestiques, ils doivent passer certains test pour obtenir un label de norme électrique Française ou Européenne.

Sachez donc que votre bidouillage avec l'Arduino, même s'il respecte toutes les conditions de sécurité, ne correspond pas à une norme (même si vous écrivez NF ou NE dessus ;)). Du coup, si par accident, vous provoquez des dégâts électriques (ou plus), vous ne serez pas remboursés par les assurances.

La seconde règle que nous appliquerons donc est la suivante :

**Règle n°2** : Toutes les parties du matériel construit doivent être isolées. Toute la partie du circuit de puissance doit être située à une distance d'au moins 1cm du circuit de contrôle. Lors de la mise en œuvre du montage, vous devez rester dans les environs, afin de minimiser les risques.

Le relais que je vous propose d'utiliser répond à la norme de distance. En revanche, il faudra veiller à isoler les parties liées au circuit de puissance.

**L'échauffement**

Lorsqu'on utilise une tension importante, il faut savoir qu'un échauffement est possible dans le circuit. Il est créé par le passage du courant dans le métal. L'échauffement est directement lié à l'ampérage.

Le courant qui peut passer dans une prise est limité par votre tableau électrique. En effet, des fusibles ou des disjoncteurs son placés, qui vont limiter la quantité de courant dans la prise. Ces matériels servent surtout à protéger d'un éventuel court-circuit, ou, justement, d'un échauffement important des câbles électriques.

Prenons une ampoule de 100W qui fonctionne sur 220V. Le Watt (symbole W) est la puissance électrique (et pas lumineuse) d'un matériel.

La formule *P*=*UxI*

﻿signifie que la puissance (en Watt) est égale à la tension (en Volts) multipliée par l'intensité (en ampères).

Donc pour notre ampoule : 100=200*xI*

, ce qui donne : *I*=100/220=0,45*A*

.

Les prises sont limitées à 16 Ampères, ce qui nous donne une bonne marge. Mais attention, le relais lui, est limité à 5 Ampères.

Donc nous ne serons pas hors normes en pilotant une ampoule. En revanche, vous ne pourriez pas piloter un convecteur électrique (chauffage) ou un chauffe-eau avec ce relais. En effet un convecteur a une puissance d'environ 2000W. Donc  2000=220*xI*

, soit *I*=2000/220=9,1*A*

. Ce qui dépasse les capacités de votre relais 5 Ampères.

L'échauffement est aussi lié à la section du fil utilisé (diamètre). Plus le fil est petit, plus il résiste au passage du courant, et donc plus il va chauffer.

Il faut donc veiller à utiliser des fils de section suffisante : 1,5 mm pour piloter un lampe.

Mais on ne peut pas utiliser la breadboard alors ?!

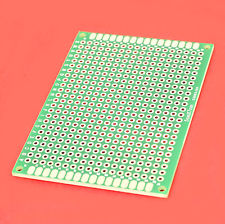
(Et bien non Cunégonde, on oublie la planche à pain !!)

En effet, les fils que vous pouvez connecter sur votre breadboard sont trop fins pour les utiliser avec un courant domestique.

Il va donc falloir utiliser un montage soudé !

**Réalisez un montage sur plaque d'essai à souder**

Les plaques d'essai à souder sont très pratiques, surtout lorsqu'on souhaite fixer un montage pour se libérer de la breadboard.

Exemple de plaque à souder

Sur l'image, on comprend bien le principe : La plaque contient des trous entourés de métal. On y glisse les pattes des composants, on les soude. On peut éventuellement relier des trous entre eux avec de la soudure (étain). On crée ainsi un circuit.

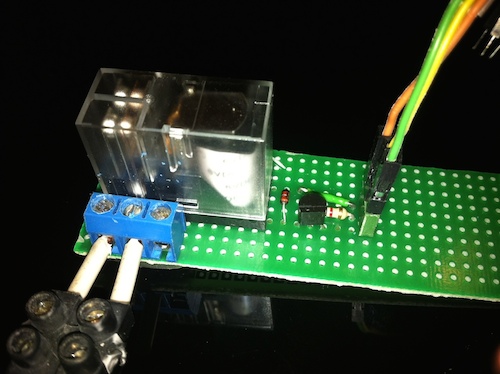
**Soudez les composants**

L'idée est donc de préparer le circuit et de le souder correctement. Les composants à souder sont donc :

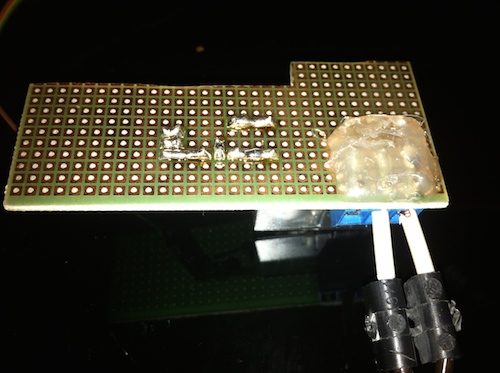
* Le relais ;
* La diode anti retour ;
* Le transistor NPN ;
* La résistance 1KΩ ;
* Le domino ;
* Les sorties vers l'Arduino.

Libre à vous d'organiser votre propre montage (en respectant bien sûr le schéma !).

Voici une photo d'un montage personnel (avant la préparation pour la mise en boîte de sécurité) :

Le montage vu de dessus

Vous pouvez observer les composants, le domino, et le relais.

Le montage vu de dessous

Même si ce montage va ensuite être fixé dans une boîte sécurisée, j'ai ajouté de la colle au pistolet (colle à chaud) pour isoler les soudures liées au circuit de puissance.

Ces plaques d'essais se découpent assez facilement (et avec de la patience) grâce à un cutter. J'utilise pour ma part du matériel de modélisme, dont un disque fin de découpe, avec une petite perceuse.

**Connectez la rallonge**

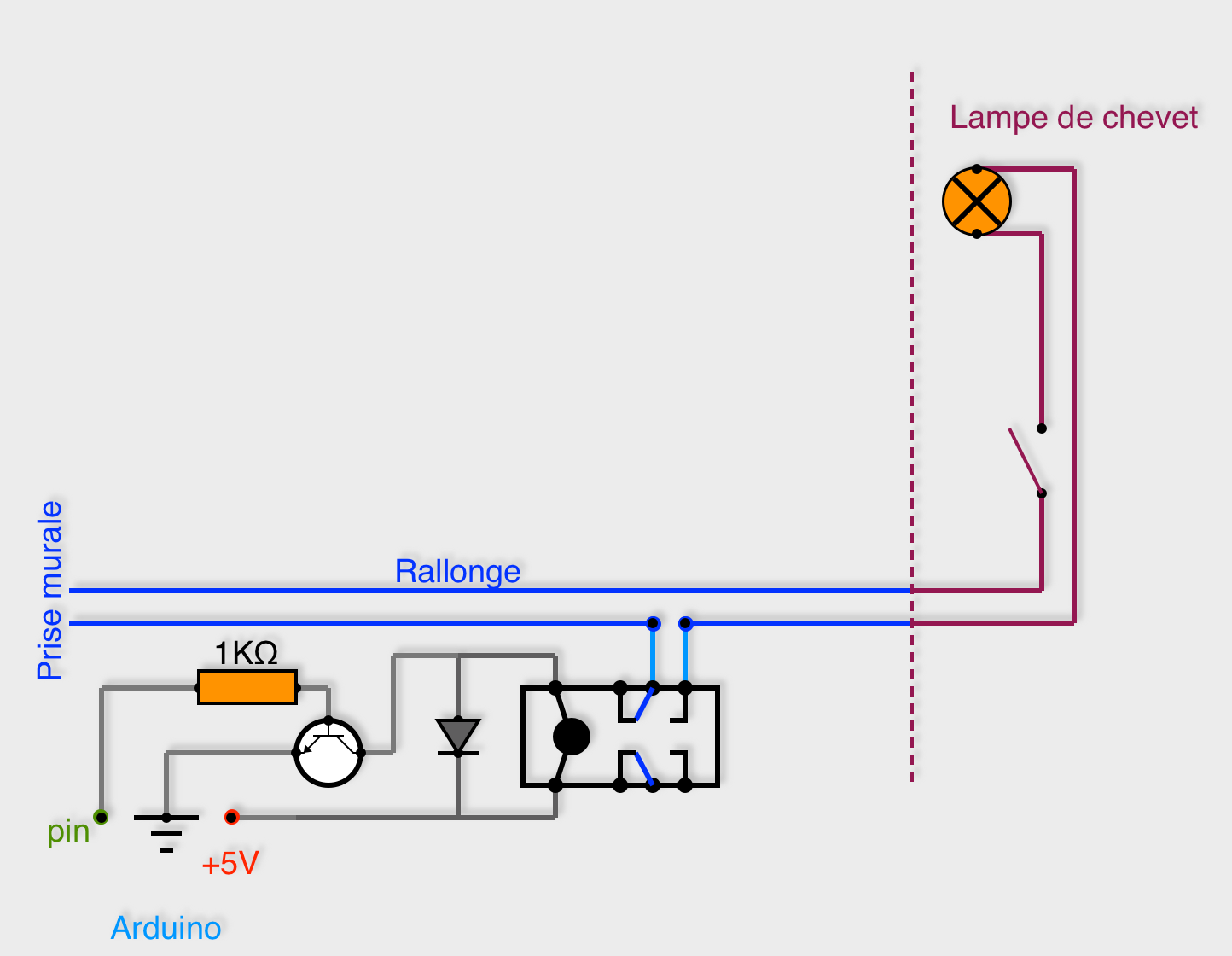
Si vous avez dans l'idée de connecter une lampe (type lampe de chevet), il faut prévoir une manière facile de l'intégrer au circuit. C'est là qu'intervient la rallonge.

Rallonge souple pour se brancher sur le courant mural et passer par la carte relais

En effet, Il suffit de couper l'un de ses deux fils (choisissez la distance qui vous convient le mieux) en laissant l'autre  entier. Une fois ce fil coupé, vous pouvez dénuder chaque côté au niveau de la section, et les connecter au domino de votre carte-relais.

Du coup, d'un côté, vous branchez votre rallonge à la prise murale, et de l'autre, vous branchez votre lampe à la rallonge. Vous avez maintenant une rallonge pilotable.

Voici un schéma récapitulatif :

Résumé du circuit électrique global

J'insiste encore sur la nécessité que votre montage soit complètement isolé pour ne pas toucher les parties liées au 220V par inadvertance.

Si vous connectez le pin 7 de  l'Arduino, à votre carte relais, que vous branchez la prise et que vous allumez l'interrupteur de votre lampe, en lançant le programme précédent, votre lampe de chevet va clignoter ! ;)

Les ampoules de lampe de chevet ne sont pas faites pour subir des clignotements rapides...

Allez, une dernière étape pour le plaisir. Grâce à tout ce que nous savons faire, nous allons piloter notre lampe grâce depuis le réseau !

**Pilotez votre lampe par le réseau**

Les chapitres précédents vous ont montré comment piloter des diodes par le réseau (privé ou publique), vous imaginez donc facilement comment piloter votre lampe maintenant. En effet, allumer une diode ou envoyer un ordre à un transistor, c'est la même chose !

Voici donc les étapes générales pour réussir ce tour de force (https://openclassrooms.com/bundles/common/images/smiley/pinch.png) :

* Créer un programme sur l'Arduino en mode serveur qui gère une petite interface réseau (type on/off) ;
* Connecter correctement votre lampe à votre carte-relais ;
* Connecter votre carte relais à votre Arduino ;
* Lancer le programme, lancer un navigateur web sur un appareil distant (téléphone, ordinateur...) ;
* Faites mumuse.

Voici un exemple de code qui me permet d'allumer une lampe depuis mon téléphone mobile (avec un navigateur ouvert sur la bonne adresse...) :

#include <SPI.h> //bibliothèqe pour SPI

#include <Ethernet.h> //bibliothèque pour Ethernet

byte mac[] = {0x90, 0xA2, 0xDA, 0x0F, 0xDF, 0xAB}; //adresse mac de votre carte

byte ip[] = {192, 168, 1, 123}; //adresse IP

int pinRelais = 7; //variable de pin pour le Relais

boolean etatRelais = 0; //etat du pin

EthernetServer serveur(80); // déclare l'objet serveur au port d'écoute 80

void setup() {

pinMode(pinRelais, OUTPUT); //pin en mode OUTPUT

Serial.begin (9600); //initialisation de communication série

Ethernet.begin (mac, ip); //initialisation de la communication Ethernet

Serial.print("\*\n-> Le serveur est sur l'adresse : ");

Serial.println(Ethernet.localIP()); //on affiche l'adresse IP de la connexion

serveur.begin(); // démarre l'écoute

}

void loop() {

gestionClient(); // fonction qui gère toute la communication avec le client

}

//----------------------Fonctions----------------------

//fonction qui gère la communication avec le client

void gestionClient() {

EthernetClient client = serveur.available(); //on écoute le port

if (client) { //si client existe

Serial.println("Client en ligne"); //on le dit...

if (client.connected()) { // si le client est connecté

GET(client); //appel de la fonction de décodage

//réponse au client

entete(client); // fonction pour l'entête de la page HTML

corps(client); // fonction pour le corps

piedPage(client); // fonction pour le pied de page

Serial.println("Fin de communication avec le client\n");

client.stop(); //on déconnecte le client

}

}

}

//fonction de fabrication de l'entête HTML

void entete(EthernetClient cl) {

//infos pour le navigateur

cl.println("HTTP/1.1 200 OK"); // type du HTML

cl.println("Content-Type: text/html; charset=ascii"); //type de fichier et encodage des caractères

cl.println("Connection: close"); // fermeture de la connexion quand toute la réponse sera envoyée

cl.println();

//balises d'entête

cl.println("<!DOCTYPE HTML>");

cl.println("<html>");

cl.println("<head><title>Relais</title></head>");

cl.println("<body><h1>Relais</h1><hr><br>");

}

//fonction de fabrication du corps de page

void corps(EthernetClient cl) {

cl.println("<br>"); // saut de ligne

cl.print("<br>Relais ");

if (etatRelais) { //si état sur 1

cl.print("ON "); // on l'écrit

}

else { // sinon

cl.print("OFF "); //on l'écrit aussi ;)

}

cl.print(" <a href=?"); //création du lien inutile de répéter l'adresse du site

cl.print('c'); //code à recevoir

cl.println(" target=\_self >Change</a><br>");

}

//fonction de fabrication du pied de page

void piedPage(EthernetClient cl) {

//balises de pied de page

cl.println("<br><hr>");

cl.println("</body>");

cl.println("</html>");

}

//fonction décodage GET

void GET(EthernetClient cl) {

boolean lu = 0; //variable pour indiquer l'état de lecture

while (cl.available()) { // tant qu'il a des infos à transmettre

char c = cl.read(); // on lit le caractère

delay(1); //delai de lecture

if (c == '?' && lu == 0) { //si "?" repéré

c = cl.read(); //on lit le caractère suivant qui contient la donnée

if (c == 'c') { //si code reçu

etatRelais = !etatRelais; //on change l'état du relais

digitalWrite(pinRelais, etatRelais); //on met à jour le pin

Serial.println(etatRelais);

}

delay(10);

lu = 1; // on dit qu'on a lu l'info

}

}

}

Si vous avez tout suivi jusque là, vous êtes à même de piloter une ou plusieurs lampe de votre maison, depuis n'importe quel navigateur connecté sur votre Arduino ;). Ça fait toujours son effet !

**Pour conclure**

Nous arrivons à la fin de ce cours. J'espère qu'il a répondu à vos attentes !

Vous avez pu y trouver des informations sur la construction de plateforme mobiles, l'utilisation du shield moteur, et du son.

Vous avez abordé différentes façon de communiquer avec votre carte Arduino en construisant des interfaces homme/machine qui peuvent être embarquées.

Vous avez découvert l'univers de l'Arduino sur le Web et quelques-unes des possibilités qui s'offrent à vous.

Vous avez même, pour certains, testé la magie de la domotique.

Mais ce n'est pourtant pas tout ce que peut vous offrir cette petite carte Italienne.

En allant plus loin, vous pourrez voir comment l'interfacer avec d'autres langages comme le Python, le C ou Processing (qui se marie très bien avec le langage de l'IDE).

Vous verrez qu'il existe d'autres shields (wifi, xBee...) et une multitude de matériels à connecter à l'Arduino, qui permettent encore plus de possibilités d'extension.

En surfant sur les forums (très nombreux) vous verrez les réalisations de chacun, le foisonnement d'idées, de possibilités.

Enfin, pour les plus audacieux, vous verrez qu'il est possible de créer ses propres bibliothèques !

(Je remercie aussi Lukas et Cunégonde, pour leur participation ! Non Lukas, pas de bisou...)

Je vous souhaite donc de programmer, de vous amuser, de créer sans limite ! Et de ne pas hésiter à partager vos expériences !

Nanomaitre.