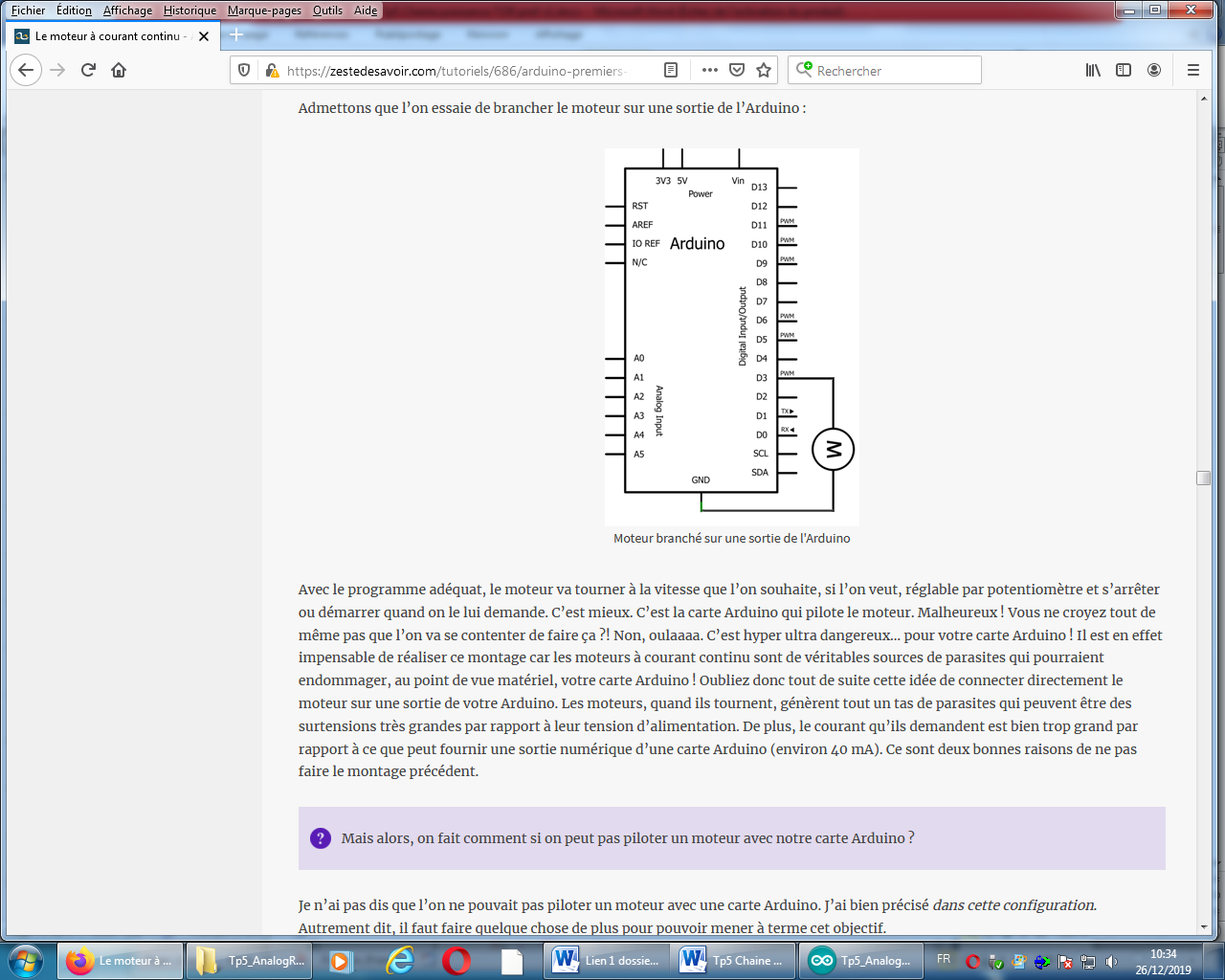
**Objectif** : **Contrôler un moteur à courant continu en utilisant un potentiomètre rotatif et un transistor de puissance.**

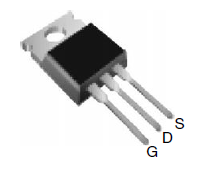


**Si vous êtes en présentiel :**

Se munir d’une carte ARDUINO et d’une plaque de câblage rapide (breadboard) , 1câble « USB », moteur miniature RM2 alimentation de 0 à 9 Vcc consommation 65 mA sous 3 Vcc à vide; d’un accumulateur 9 v ; 1transistor MOSFET 520 N; 1 entrée un potentiomètre rotatif avec un pont milieu; d’une diode ; 6 fils fins. 1 résistance 10 k Ohm.

**Si vous êtes en distanciel :**

Se connecter au site Tinkercad, ouvrir un logiciel de capture d’écran et d’un traitement de texte.



**IFR520N**



**A0**

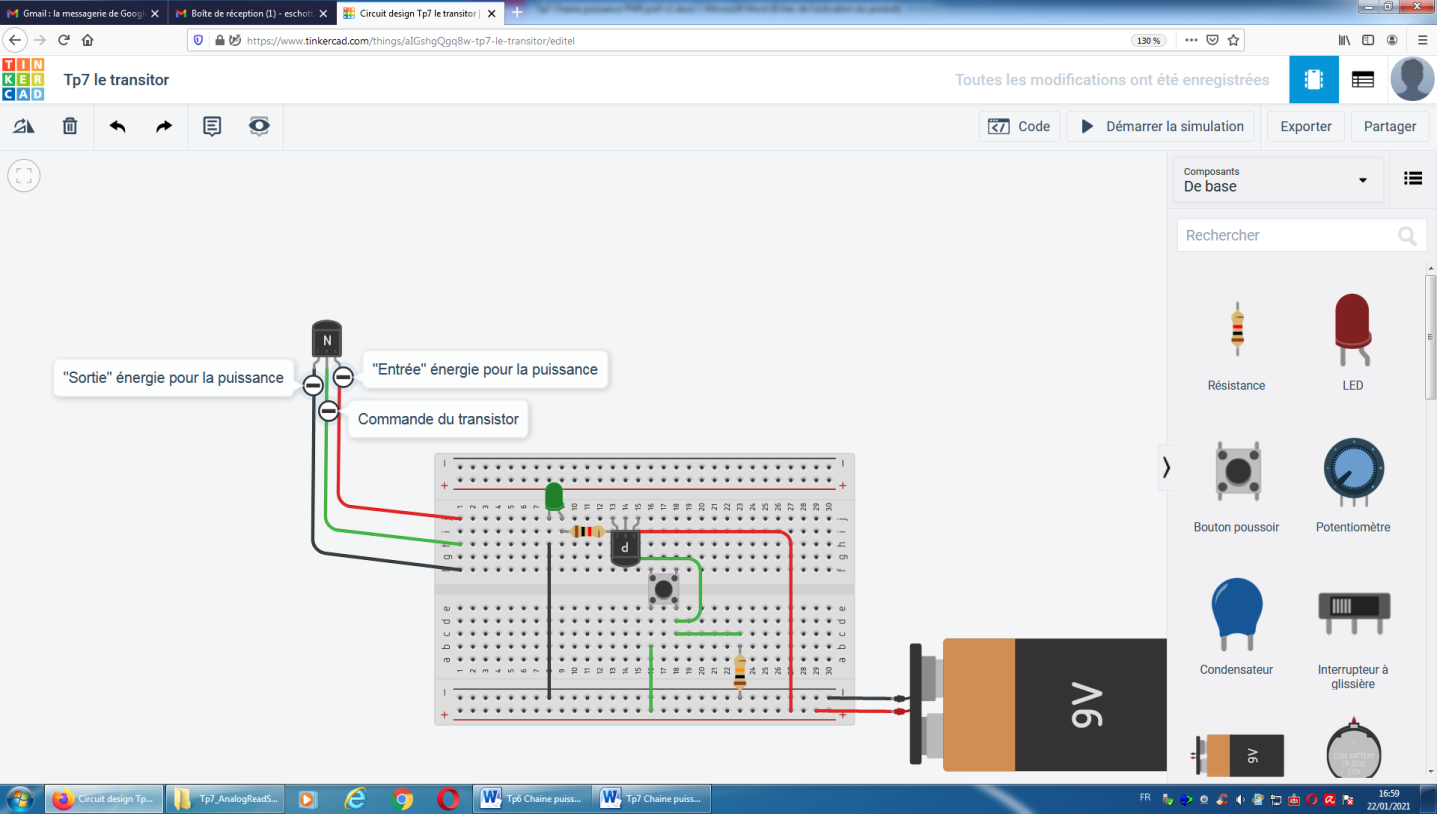
**GND**

**+5 V**

.

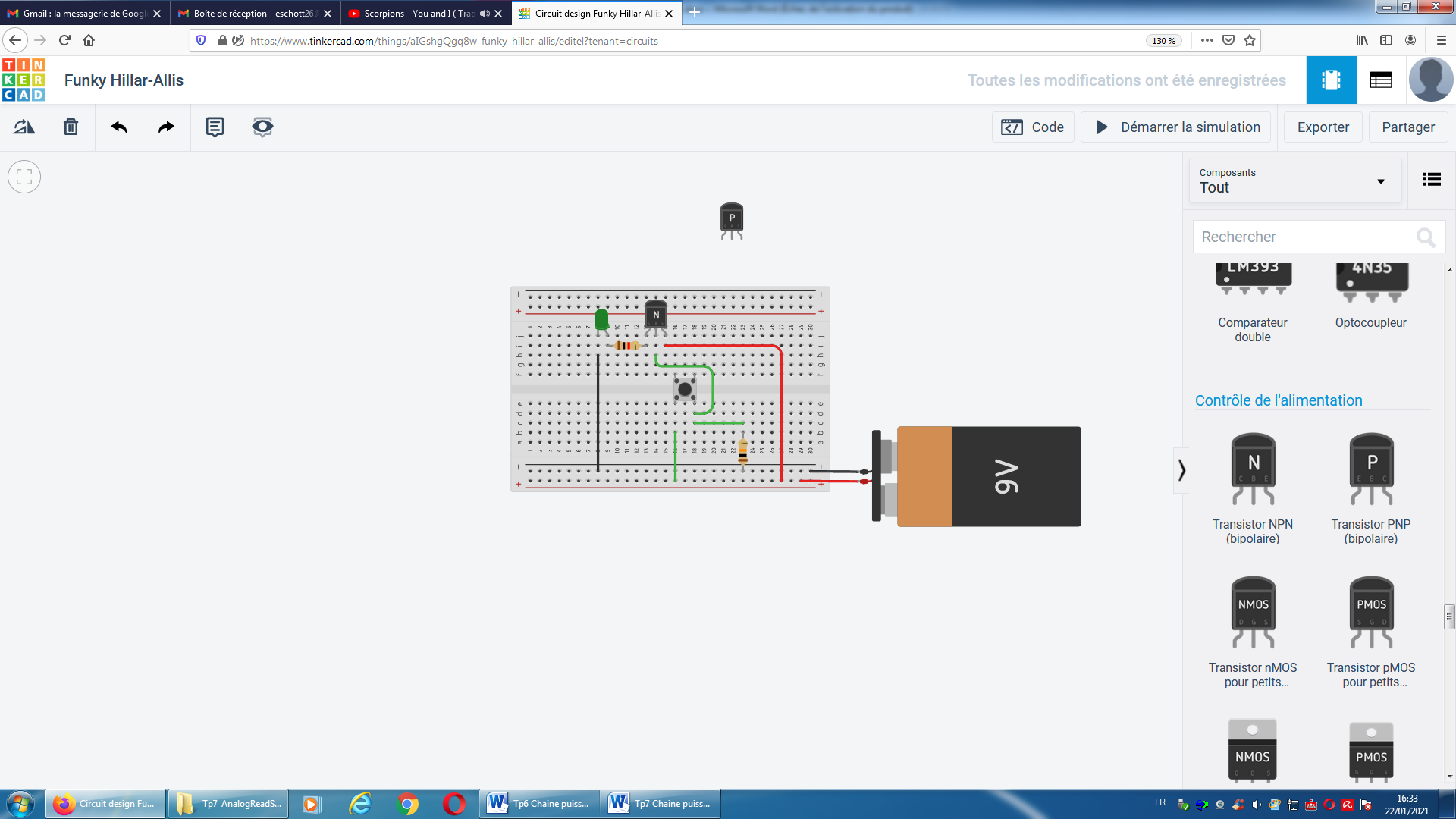
1. **Rappel :**

* **Ne jamais brancher le moteur électrique directement, sans autres composants intermédiaires sur la carte ARDUINO au risque de la griller. (Car** la carte ne doit pas débiter plus de 40 mA par port de sortie soit 200mA au total).

**Donc NE PAS FAIRE LE CABLAGE CI-DESSUS**

1. **Des infos faciles à comprendre sur le transistor**

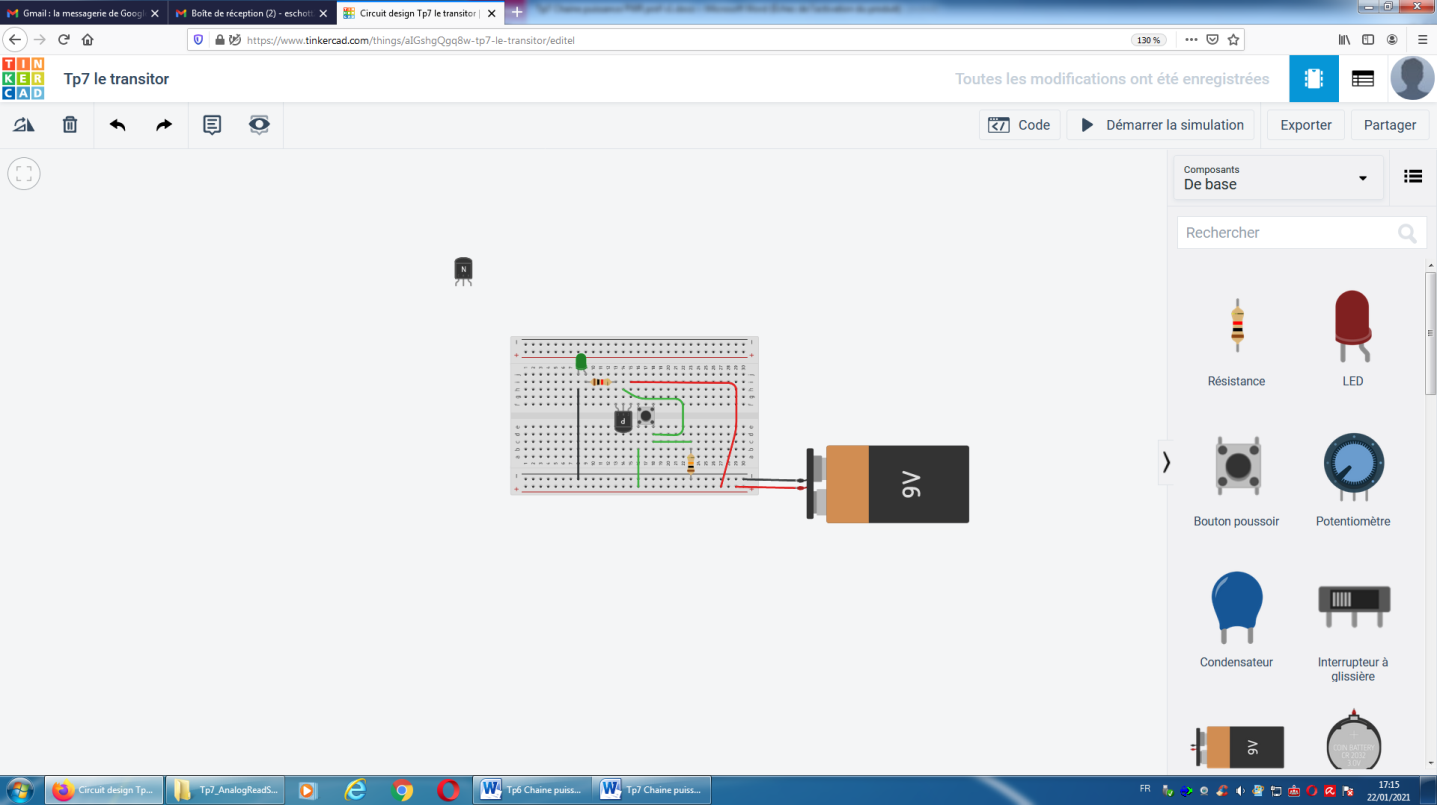
Mais quelle différence pouvez-vous observer dans le fonctionnement de ces deux montages quand on utilise transistor NPN ou PNP ?



Pour le comprendre utilisez Tinkercad, **refaire le câblage ci-contre**, avec un transistor NPN et l’essayer.

Ecrire vos remarques ci-dessous :

Lorsqu’on appuie sur le bouton la LED

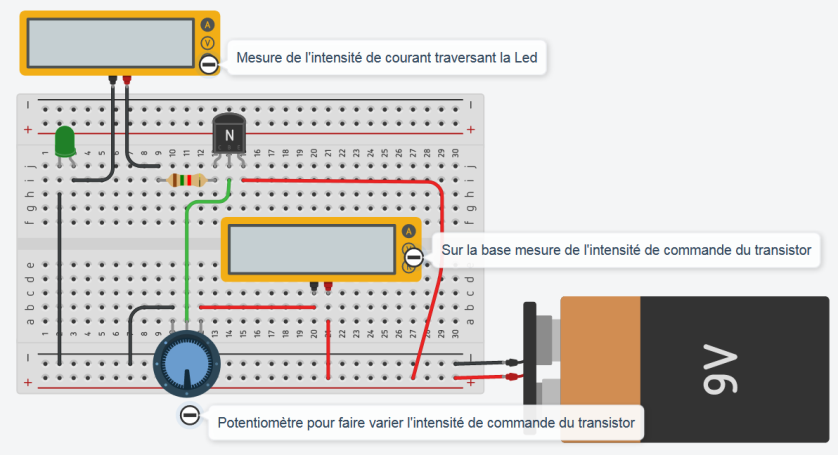
s’allume.

**Remplacez dans** le précèdent câblage le transistor NPN par un PNP et l’essayer.

Ecrire vos remarques ci-dessous :

Si on appuie sur le bouton poussoir

la LED déjà allumé s’éteint.

1. Travail facultatif seulement pour ce paragraphe, c’est pour ceux qui sont en avance et qui veulent approfondir leurs connaissances sur le transistor, ici NPN.

A partir du montage précèdent, **le modifier, en rajoutant deux ampèremètres le réaliser** et **l’essayer.**

**Relevez** les valeurs de l’intensité traversant la Led en sortie du transistor sur la borne (c).

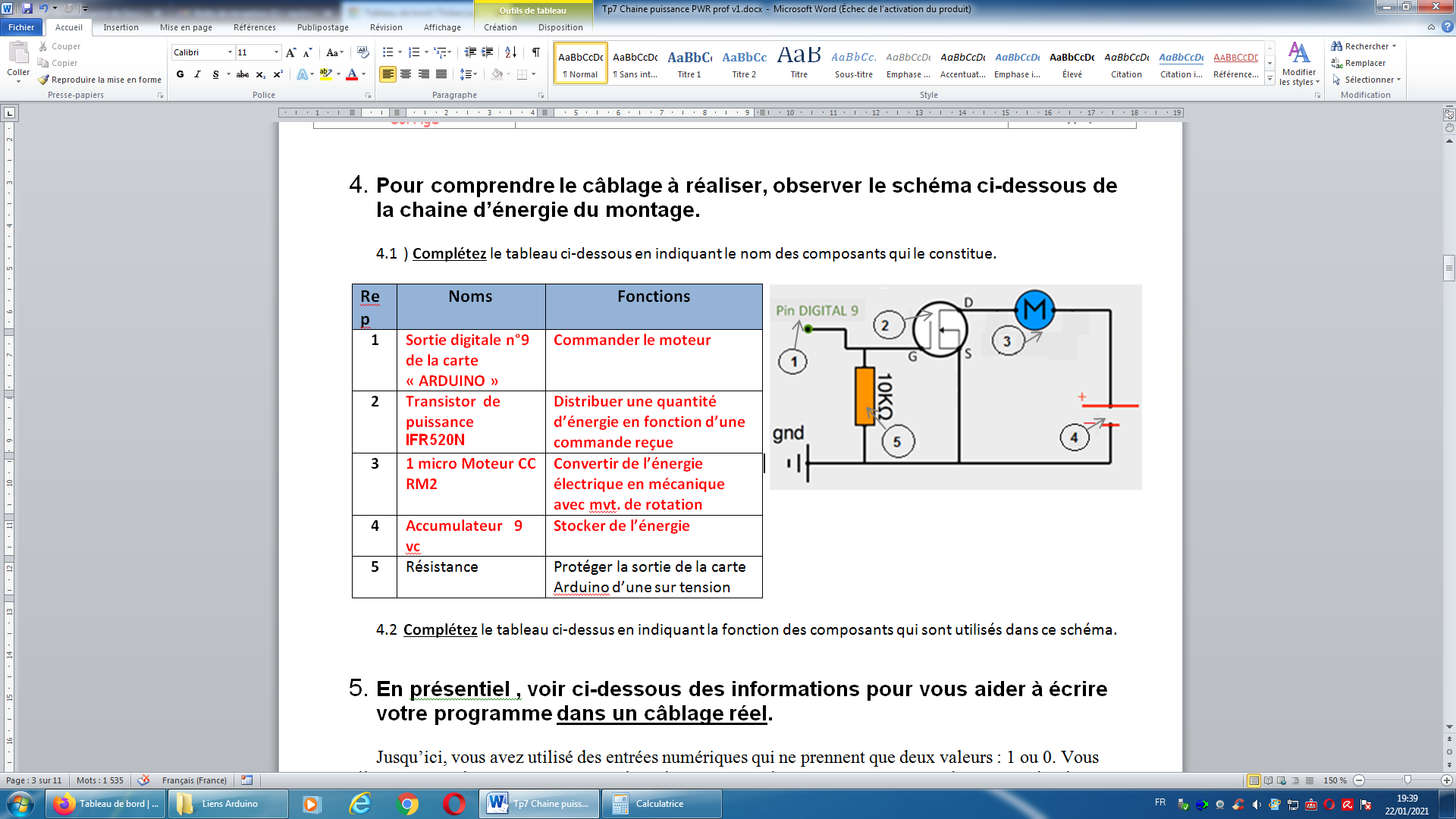


**Complétez** alors le tableau ci-dessous avec les valeurs relevées sur votre câblage.

Sur un tableur **tracez** les valeurs de sortie d’intensité traversant la Led en fonction des valeurs d’intensité de commande du relais.

Vous devez **réaliser** une capture d’écran et la coller ci-dessous.

Qu’en déduisez-vous ? …………………………………………………………………………….

1. **Pour comprendre le câblage à réaliser, observez le schéma ci-dessous de la chaine d’énergie du montage.**
   1. ) **Complétez** le tableau ci-dessous en indiquant le nom des composants qui le constitue.

|  |  |  |
| --- | --- | --- |
| **Rep** | **Noms** | **Fonctions** |
| **1** | **Pin 9 de la carte arduino** | **D’envoyer du signal au pin 9** |
| **2** | **Transistor NMOS (MOSFET)** | **Sert a alimenter des charges de communication** |
| **3** | **Moteur** | **Transforment l’énergie électrique en énergie électrique mécanique de rotation** |
| **4** | **Pile** | **Fournir de l’électricité** |
| **5** | Résistance | Protéger la sortie de la carte Arduino d’une sur tension |

* 1. **Complétez** le tableau ci-dessus en indiquant la fonction des composants qui sont utilisés dans ce schéma.

1. **En présentiel , voir ci-dessous des informations pour vous aider à écrire votre programme dans un câblage réel.**

Jusqu’ici, vous avez utilisé des entrées numériques qui ne prennent que deux valeurs : 1 ou Vous allez à présent découvrir comment utiliser des entrées . travers la prise en main d’un nouveau composant : le potentiomètre analogique. Vous allez apprendre à vous en servir pour créer toute une nouvelle gamme de réactions en traitant des données qui ne sont pas uniquement des 1 et des 0.

Cela vous amènera à enrichir votre vocabulaire de programmation avec la notion **de mappage de valeurs.**

Sur la carte ARDUINO, vous voyez 6 connexions possibles, notées A0, A1, A2, A3, A4, et A5. C'est ici que vous pouvez connecter un fil pour recevoir un signal analogique.

Un des 6 CAN (Convertisseur analogique numérique) le transformera ensuite en signal numérique compris entre 0 et 1023.

C’est Très intéressant car la carte Arduino ne peut fournir seulement qu’une tension de 0 ou de 5 v, mais par contre elle peut "simuler" des tensions variables comprises entre 0 et 5V.

|  |  |  |
| --- | --- | --- |
| Nom | Description graphique | Codes de programmation |
| 1 Entrée Potentiomètre rotatif de1 MΩ avec le point milieu relié en A0 | **A0**  **GND**  **+5 V** | const int bouton1 = 1; //attribution de la broche 4 à la variable bouton1.  **void setup**() {  }  **void loop**() {  } |
| 1 sortie **transistor** | **IFR520N** | #define RELAY\_PIN 8  **void setup**() {  }  **void loop**() {  } |

**Le mappage de valeurs :** Il faut savoir que le langage de l'ARDUINO propose directement une fonction qui va nous éviter tous ces calculs ! Elle se présente sous cette forme :

map(valeur,min,max,transMin,transMax);

avec :

* valeur : c'est la valeur que vous voulez transformer (l’intensité du potentiomètre dans notre cas)
* min et  max : c'est la plage de valeurs. (le minimum et le maximum qu'elle peut prendre, soit ici 0 et 1023 qui correspond à la plage de résolution du CAN de l'Arduino).
* transMin et  transMax : c'est la plage de valeurs dans laquelle on doit transformer la  valeur (ici 1 et 5, car nous attendons 5 positions différentes)

 On peut donc utiliser la fonction map ainsi dans notre exemple :

int valInter=map(valPot,0,1023,1,5);

1. **Mais quel signal est donné sur la sortie digitale pour faire varier la vitesse du moteur ? (à lire)**

C’est le microcontrôleur qui va produire un signal carré dont le rapport cyclique est variable. Et grâce à cela, vous obtenez une tension moyenne (comprise entre 0 et 5V) en sortie de la carte Arduino. Il faut juste bien penser à utiliser les sorties adéquates, à savoir : **3, 5, 6, 9, 10 ou 1**. (Pour les repèrer un petit symbole alternatif est noté à coté de chacune de ces bornes).

Je résume : en utilisant la PWM, on va générer une tension par impulsions plus ou moins grandes. Ce signal va commander le transistor qui va à son tour commander le moteur. Le moteur va donc être alimenté par intermittences à cause des impulsions de la PWM. Ce qui aura pour effet de modifier la vitesse de rotation du moteur.

« Pas de stress » le moteur garde une inertie de rotation et comme la PWM est un signal qui va trop vite pour que le moteur ait le temps de s’arrêter puis de redémarrer, on va ne voir qu’un moteur qui tourne à une vitesse réduite.

Finalement, nous allons donc pouvoir modifier la vitesse de rotation de notre moteur en modifiant le rapport cyclique de la PWM.

Plus il est faible (un état BAS plus long qu’un état HAUT), plus le moteur ira doucement. Inversement, plus le rapport cyclique sera élevé (état HAUT plus long que l’état BAS), plus le moteur ira vite.

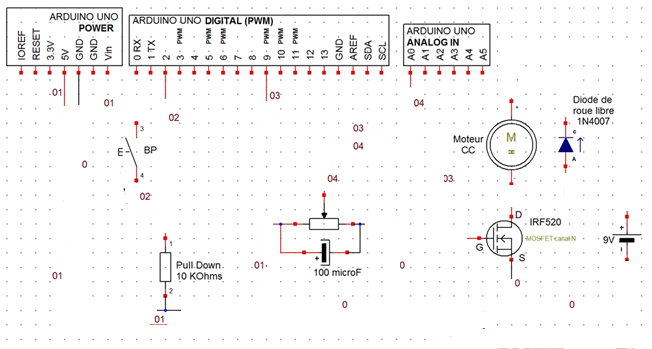
Tout cela couplé à un transistor pour ajuster la puissance (et utiliser la tension d’utilisation adaptée au moteur) et donc vous pouvez faire tourner le moteur à la vitesse que vous voulez.

1. **Voici des informations pour vous aider dans un câblage virtuel sur TINKERCAD.**

|  |  |  |
| --- | --- | --- |
| Eléments nécessaires pour réaliser le câblage virtuel dans la bibliothèque | Photo d’un composant réel. | Pour le connecter |
| **Potentiomètre** |  |  |
| **Transistor MOSFET pour un fonctionnement « proportionnel » de l’élément piloté.** |  | \_ Câble noir « GND » sur la borne « S » du Transistor.  \_ Câble violet  pour alimenter le moteur sur la borne « D ».  \_ Câble orange « Sortie 3 PWM » sur la borne « G ». |
|  |  |  |

1. **Ecrire votre programme ci-dessous dans le logiciel Arduino et dans Tinkercad.**
2. **Modéliser votre câblage**

Si vous êtes en présentiel :

**Complétez** le schéma du montage à réaliser

Après avoir **fait valider** le schéma, **câbler votre montage** en étant hors tension.

**Obtenir l’accord de votre enseignant pour le mettre sous tension**.

Si vous êtes en distanciel :

Dans Tinkercad, tracer votre schéma en dupliquant le TP6 et en modifiant celui-ci.

1. **Valider le Tp en faisant des copies d’écran, car le compte rendu de cette activité sera ramassée et notée.**