

Projet Micro-Contrôleur :

Rapport - Convertisseur DC-DC.

Réalisé par :

BERKANI Oumaima – BENLAFQIH Ahmed

Sommaire :

I. Introduction

- Exploration de l'interface de Tinkercad
- Essai de différents projets Arduino pour comprendre les principes de base

II. Réalisation du convertisseur DC-DC en mode abaisseur de tension ("buck")

- Fonction PWM de l'Arduino
- Etude des différentes composantes du circuit
- Conception du circuit du convertisseur DC-DC
- Mesure de la gamme de tension en sortie du convertisseur

III. Conclusion

- Conclusion générale sur le projet

I. Introduction :

L'objectif principal de ce projet était de comprendre le fonctionnement d'un convertisseur DC-DC en mode abaisseur de tension ("buck"), d'implémenter une régulation PID pour maintenir une tension de sortie constante, et de caractériser pleinement les performances du système.

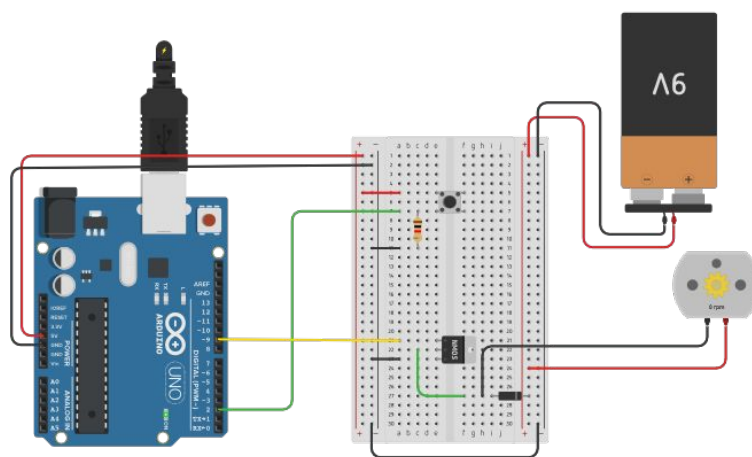
La première partie de ce rapport présente notre démarche pour prendre en main le logiciel Tinkercad et les projets Arduino, ainsi que notre exploration des différentes fonctionnalités disponibles. Ensuite, nous détaillons la conception et la mise en œuvre du convertisseur DC-DC, en mettant l'accent sur les choix de composants et d'algorithmes de contrôle. Enfin, nous présentons les résultats obtenus, y compris la gamme de fonctionnement du convertisseur et l'efficacité de la régulation PID.

Ce projet va nous permettre d'acquérir une compréhension approfondie des principes de base des convertisseurs DC-DC.

Tinkercad :

Tinkercad est une application de modélisation 3D et de simulation en ligne qui permet aux utilisateurs de créer des prototypes virtuels de différents projets, notamment des circuits électroniques. Elle leur permet aussi de les tester.

Sur Tinkercad, nous avons commencé par réaliser des circuits simples utilisant l'Arduino. Ces projets allaient du branchement simple de LED et de les faire clignoter jusqu'à des circuits plus complexes, tels qu'un circuit contrôleur de moteur utilisant un bouton et un NMOS. Voici une représentation de ce circuit :



Dans ce circuit, le bouton est utilisé comme interrupteur pour activer ou désactiver le moteur, contrôlé par le NMOS. L'Arduino est programmé pour détecter l'état du bouton et commander le NMOS en conséquence, permettant ainsi de contrôler le fonctionnement du moteur.

Tinkercad nous a permis de visualiser et de tester le fonctionnement de ces circuits de manière virtuelle avant de les réaliser physiquement.

II. Réalisation du convertisseur DC-DC :

PWM :

Pour pouvoir procéder avec notre convertisseur on doit tout d'abord faire le point sur une particularité de notre Arduino Uno essentielle au fonctionnement de ce dernier, cette fonctionnalité est bien la PWM (Pulse Width Modulation)

Le PWM (Pulse Width Modulation) est une technique utilisée pour moduler la tension délivrée par la carte Arduino. Au lieu de fournir simplement 5 volts ou rien, on crée des signaux carrés (ou créneaux) où la durée pendant laquelle le signal est à 5 volts (ou HIGH) et la durée pendant laquelle il est à 0 volts (ou LOW) peuvent être contrôlées.

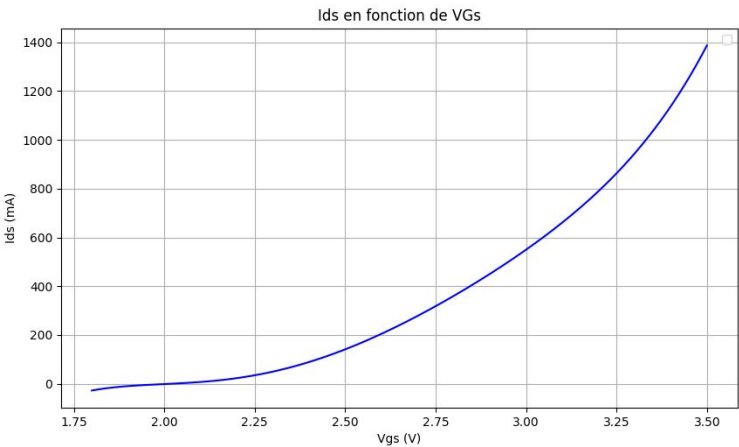
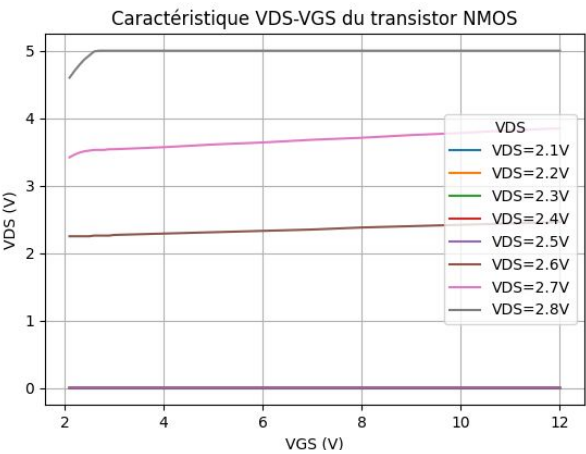
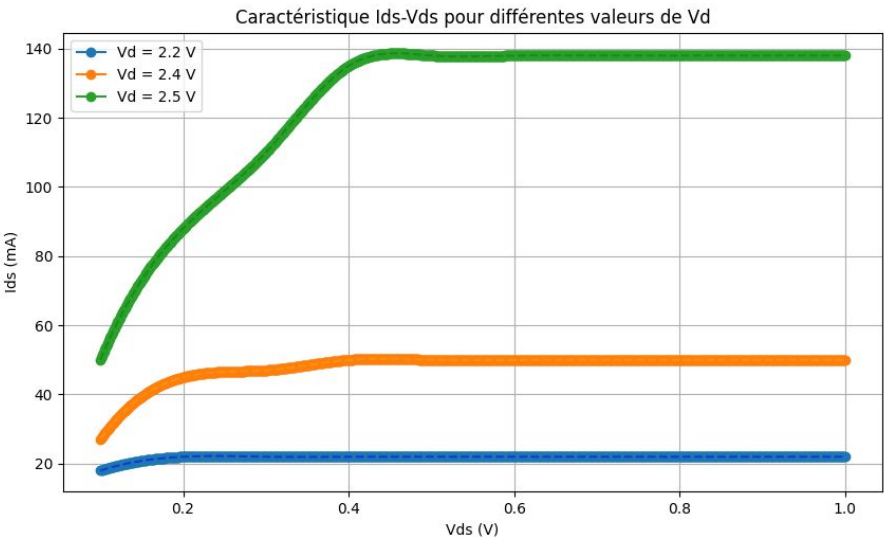
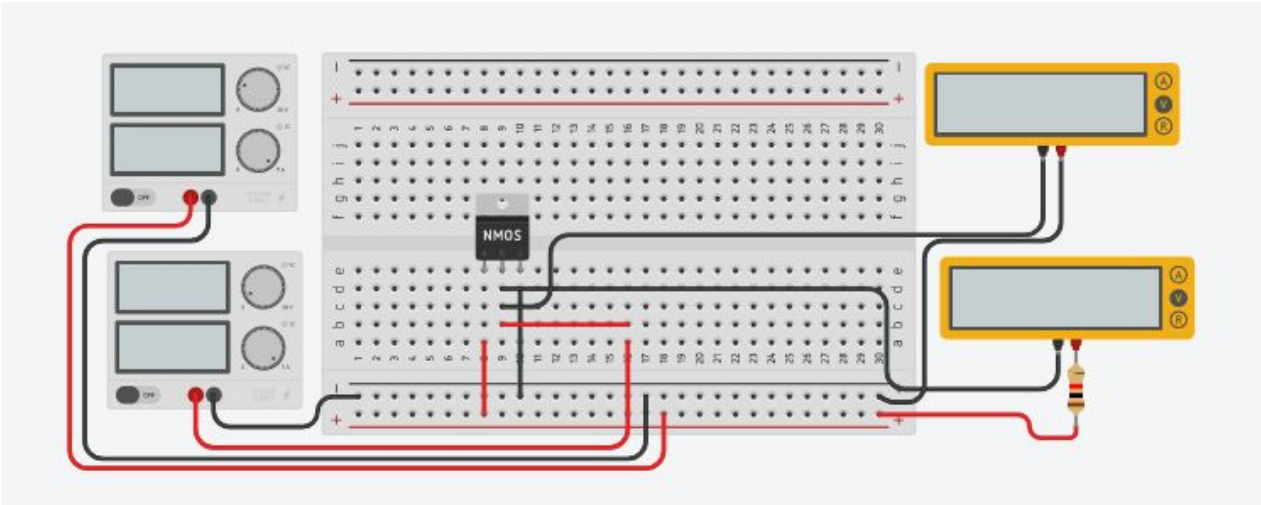
Pour cela, un pin de la Arduino est connecté au transistor, et un code est envoyé à la Arduino pour définir le rapport cyclique, également appelé duty-cycle, du signal PWM. Le rapport cyclique représente la proportion de temps pendant laquelle le signal est à l'état haut par rapport à la période totale du signal. La moyenne de ce signal PWM nous donne la nouvelle valeur de tension.

En ajustant le rapport cyclique du signal PWM, on peut contrôler la tension moyenne délivrée, ce qui permet de réguler la tension de sortie dans un convertisseur DC-DC, par exemple.

NMOS :

Un NMOS (Metal-Oxide-Semiconductor Field-Effect Transistor) est un transistor à effet de champ dans lequel le courant entre le drain et la source est contrôlé par la tension appliquée à la grille

Grâce à Tinkercad, nous avons pu représenter un transistor NMOS et mener une étude de ce dernier afin de le caractériser.

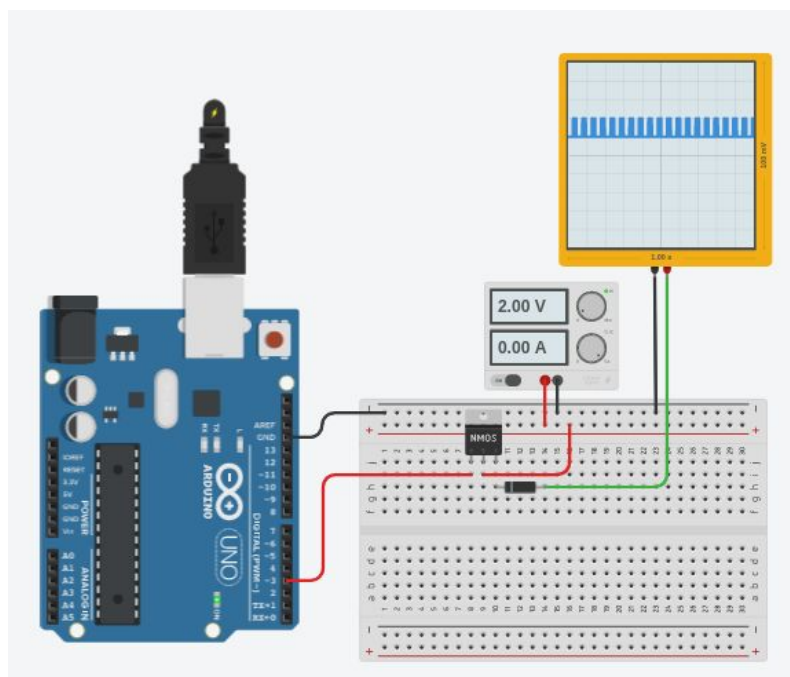


Hacheur :

Dans le cadre de notre étude, nous avons commencé par mettre en place un hacheur fonctionnel dont le but est de générer une tension créneau en sortie variant entre 0 et X Volts. Nous avons procédé par simuler ce dernier sur Tinkercad.

Nous utilisons une Arduino pour simuler le fonctionnement du hacheur. Cette Arduino produit une tension créneau de fréquence constante et de rapport cyclique variable. Cette tension est appliquée à la grille (G) de notre transistor NMOS. La grille agit comme un interrupteur pour une seconde source de tension placée au drain (D). Lorsque la tension appliquée à la grille dépasse sa tension de seuil, le courant peut circuler entre le drain et la source. Cette configuration nous permet de produire un courant créneau avec une tension définie.

Le montage sur Tinkercad nous permet de simuler et de tester ce circuit de manière efficace avant de le mettre en œuvre dans un environnement réel.



Comme on peut le remarquer sur la figure précédente, notre hacheur est bien opérationnel. En mode abaisseur de tension, il convertit un signal d'entrée continu (sur le schéma précédent 2V) et nous donne en tension de sortie un signal rectangulaire (qui ne dépasse pas 0,98V dans ce cas).

```

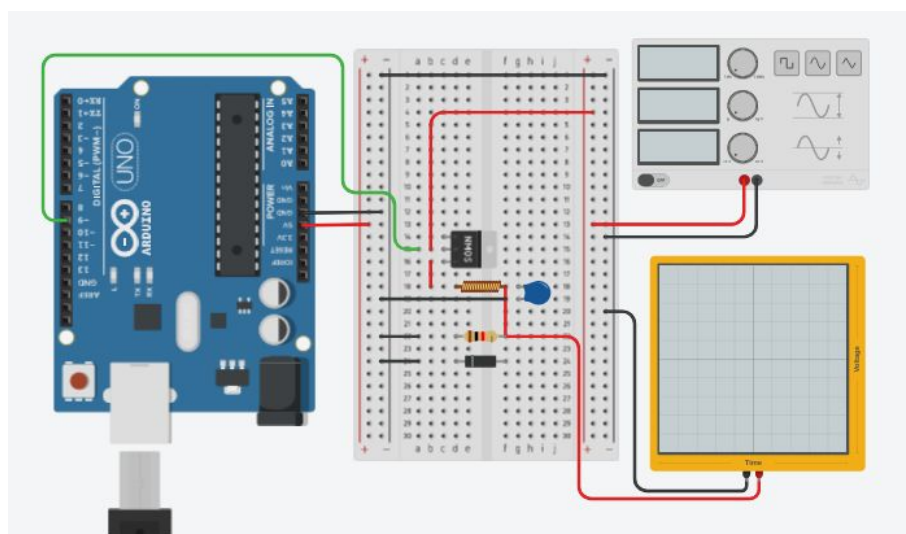
1  int pin = 3;
2  int dut = 50;
3
4  void setup()
5  {
6      pinMode(pin, OUTPUT);
7      analogWrite(pin, dut);
8  }
9
10 void loop()
11 {
12     //
13 }

```

Et nous avons sur la figure précédente le code que nous avons injecté dans notre Arduino. Ce code Arduino utilise la modulation de largeur d'impulsion (PWM) pour contrôler la sortie analogique sur le pin 3 avec un rapport cyclique de 50%.

Convertisseur buck :

Notre convertisseur buck n'est rien d'autre que notre hacheur exploité précédemment, combiné à un filtre passe-bas. Nous ajoutons le filtre passe-bas car notre commutateur n'est pas parfait. Après le passage par le hacheur, notre signal de sortie peut toujours contenir une forme de bruitage, dû aux harmoniques qui peuvent introduire d'autres formes d'ondulations non désirées. Le filtre passe-bas permet de lisser le signal de sortie en éliminant ces composantes indésirables et en ne laissant passer que la composante continue désirée.



```

1 void loop() {
2   int pin = 9;
3   int duty_cycle = 102;
4   int resistance_values[] = {100, 200, 300};
5   int num_resistances = sizeof(resistance_values) / sizeof(resistance_values[0]);
6
7   for (int i = 0; i < num_resistances; i++) {
8     int resistance = resistance_values[i];
9
10    for (int dc = 0; dc <= 255; dc += 25) {
11
12      float voltage = (dc / 255.0) * 5.0;
13      float current = voltage / resistance;
14
15
16      printf("Duty Cycle: %d, Resistance: %d ohms, Voltage: %.2f V, Current: %.2f A\n", dc, resistance, voltage, current);
17    }
18  }
19 }

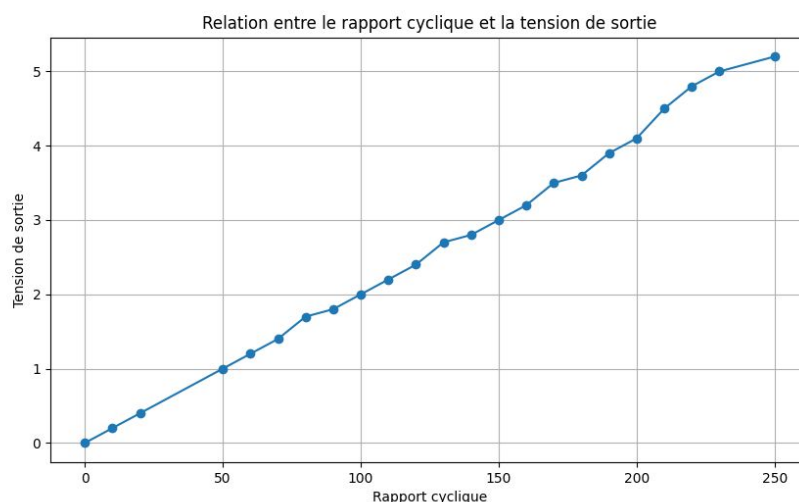
```

Ci-dessus se trouve le code injecté dans l'Arduino de notre convertisseur buck. Ce code simule des mesures de tension pour différentes valeurs de rapport cyclique et de résistance. Il utilise une boucle pour itérer à travers différentes valeurs de résistance et une autre boucle pour itérer à travers les valeurs de rapport cyclique. Pour chaque combinaison de rapport cyclique et de résistance, il calcule la tension à l'aide de la loi d'Ohm et imprime les valeurs de tension et de courant correspondantes.

Pour déterminer les bonnes valeurs de notre inductance L et du condensateur C , nous avons essayé plusieurs valeurs jusqu'à ce que les perturbations ne soient plus visibles sur l'oscilloscope de notre simulation. En fin de compte, nous nous sommes fixés sur les valeurs suivantes : $C = 100 \mu\text{F}$, $L = 50 \text{ mH}$ et nous avons pris $R = 10 \text{ k}\Omega$.

Rapport cyclique :

De retour sur Tinkercad, nous avons varié le rapport cyclique entre 10 % et 100 % en modifiant le code injecté sur l'Arduino Uno. En visualisant ensuite les résultats sur l'oscilloscope de notre simulation, nous avons obtenu la courbe suivante :



En raison des incertitudes liées à Tinkercad, nos graphiques peuvent présenter des imprécisions, comme cela a été le cas pour le graphique précédent.

D'après notre graphique, il est observé que la tension de sortie est proportionnelle au rapport cyclique. Nous avons l'intention de renforcer ces résultats obtenus par simulation en utilisant un montage théorique, où les mesures seraient plus précises. Cependant, en raison de problèmes matériels, cette étape n'a pas pu être réalisée.

III. Conclusion :

En conclusion, ce projet nous a permis de comprendre en profondeur le fonctionnement des convertisseurs DC-DC.

Tout d'abord, nous avons exploré les fonctionnalités de Tinkercad, une plateforme de simulation de circuits électroniques hors pair qui nous a permis de créer et de tester nos conceptions en toute sécurité.

Ensuite, nous avons étudié les caractéristiques du transistor NMOS ainsi que plusieurs autres composants du convertisseur buck pour finalement concevoir ce dernier en mode abaisseur de tension (buck) en utilisant une Arduino Uno. Nous avons optimisé les paramètres du convertisseur pour obtenir une gamme de fonctionnement spécifique, notamment en termes de tension de sortie et de courant.

L'analyse des résultats obtenus a confirmé la performance du convertisseur et de la régulation, validant ainsi la fiabilité et l'efficacité de la conception.

FIN.