

RAPPORT DE PROJET

Microcontrôleur

Projet réalisé par :
Alice FOUILLE
Glen ROGER

Table des matières

Introduction.....	2
- Abstract	2
I. Utilisation des transistors.....	3
- PMOS.....	3
II. Fonctionnement du PWM	3
- Simulation d'un signal PWM	4
III. Étude de la partie hacheur	4
IV. Étude du filtre aux bornes de la résistance de charge	5
- Fonctionnement de la diode	5
- Fonctionnement de la bobine :	5
- Dimensionnement de la bobine :	5
- Fonctionnement du condensateur.....	6
V. Circuit complet	6
- Montage du circuit	6
VI. Vérification et amélioration du montage	7
- Tension de sortie en fonction du rapport cyclique	7
- Tension de sortie en fonction de la résistance de charge.....	7
- Utilisation d'un signal PWM avec un correcteur proportionnel	7
- Mise en place d'un potentiomètre.....	8
Conclusion	8
Annexe	9

Introduction

L'objectif de ce projet est de réaliser et caractériser un convertisseur DC-DC de type buck. Ce type de convertisseur permet de convertir une tension continue en une autre tension continue de plus faible valeur.

Le convertisseur est constitué d'une partie hacheur et d'un filtre. Il permet de couper périodiquement la tension continue en entrée du circuit. Il est le plus souvent constitué d'un transistor à effet de champ MOSFET, actionné par un signal PWM, fonctionnalité permise notamment par Arduino. Dans cette étude, nous devons dans un premier temps tracer les caractéristiques du transistor afin de comprendre son fonctionnement. Le transistor PMOS agit comme un interrupteur et est piloté par un signal PWM et un transistor NPN.

Le filtre quant à lui est constitué d'une bobine, d'un condensateur, d'une résistance de charge et d'une diode qui agit comme un interrupteur en état opposé à celui du PMOS. C'est un filtre passe-bas qui permet de transformer la tension d'entrée en une tension continue moyennée, dont la valeur est contrôlée par le rapport cyclique du PWM.

En fin, une boucle de rétroaction permet d'auto-réguler la tension afin d'obtenir la tension souhaitée.

- Abstract

Les convertisseurs buck ont de nombreuses applications en électronique car ils permettent d'abaisser une tension à toute valeur choisie en faisant varier uniquement le rapport cyclique du PWM, tout en ayant un très bon rendement, bien meilleur que celui d'un diviseur de tension par exemple.

L'objectif de notre projet est de réaliser et caractériser un convertisseur abaisseur de tension. Pour cela, nous avons utilisé des simulations faites sur les logiciels Tinkercard et LTspice, et conçu le système en pratique pour vérifier que les valeurs théoriques établies en première approche, aboutissent bien à la tension de sortie désirée.

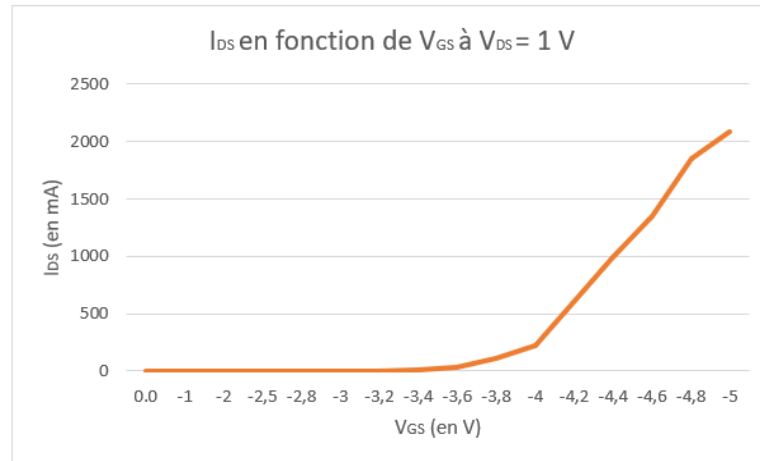
En étudiant les différentes parties du convertisseur (PWM seul, partie hacheur, composants du filtre un à un, filtre complet, convertisseur complet), Nous avons pu caractériser le convertisseur DC-DC en termes de fréquence, de tensions haute et basse du PWM, des valeurs de R, L et C pour obtenir un abaisseur de tension. Nous avons également pu réaliser un correcteur proportionnel, c'est-à-dire une boucle de rétroaction qui permet au système de s'auto-réguler pour obtenir la tension souhaitée.

I. Utilisation des transistors

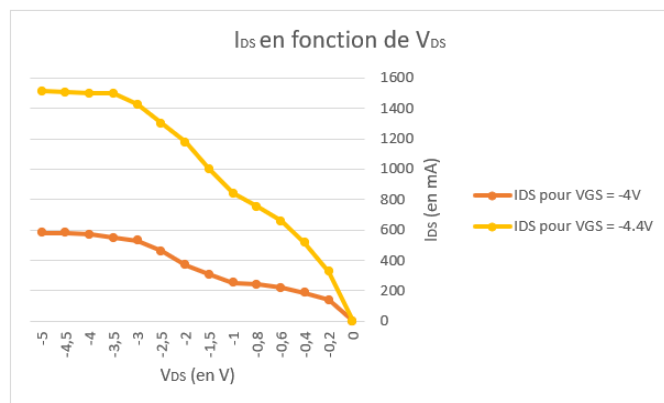
Dans le cadre de ce projet, nous avons deux types de transistors à effet de champ. Le transistor NMOS est construit avec une source et un drain de type N et un substrat de type P, tandis que le PMOS est construit avec une source et un drain de type P et un substrat de type N. Une des différences majeures du NMOS et du PMOS est qu'à haute tension, le NMOS conduit, contrairement au PMOS. Et à basse tension le PMOS conduit, contrairement au NMOS. Ici, nous décidons de prendre le transistor PMO

- PMOS

$V_{GS}(V)$	$I_{DS}(mA)$
0.0	1
-1	1
-2	1
-2,5	1
-2,8	1
-3	1
-3,2	3
-3,4	8
-3,6	37
-3,8	108
-4	225
-4,2	610
-4,4	990
-4,6	1350
-4,8	1850
-5	2085



$V_{DS}(V)$	I_{DS} pour $V_{GS} = -4V$	I_{DS} pour $V_{GS} = -4.4V$
0	0	0
-0,2	140	325
-0,4	188	515
-0,6	221	659
-0,8	242	754
-1	254	842
-1,5	307	1001
-2	370	1176
-2,5	460	1300
-3	530	1425
-3,5	550	1500
-4	570	1500
-4,5	580	1504
-5	580	1511



On remarque donc que le PMOS fonctionne avec des tensions négatives, comme on considère le courant $V_{GS} = -V_{SG}$. Cette propriété du PMOS nous sera utile plus tard.

En analysant les courbes, on voit que le PMOS FQP1706 a une valeur seuil de -3.6 V pour $V_{DS} = 1$ V. Cette tension de seuil nous indique la plage de fonctionnement de notre transistor comme on peut voir sur le second graphique qui montre qu'à partir de cette valeur seuil, nous rencontrons un plateau de valeur.

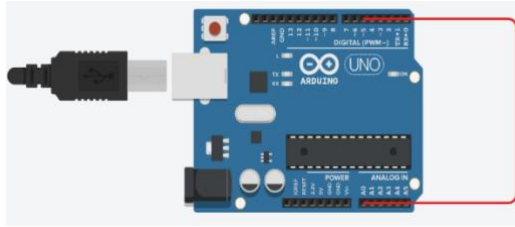
II. Fonctionnement du PWM

Le PWM ou le « Pulse Width Modulation » nous permet d'obtenir un signal numérique en faisant varier le rapport cyclique, c'est-à-dire qu'on veut que l'état haut du signal n'ait pas le même temps que l'état bas du signal sauf pour un rapport cycle de 50%. Pour créer ce signal, nous utilisons une carte Arduino.

Nous avons, tout au long du projet, utiliser la PIN 5 de l'Arduino car sa fréquence d'écriture est de 980 Hz et qui est la plus élevée sur la carte.

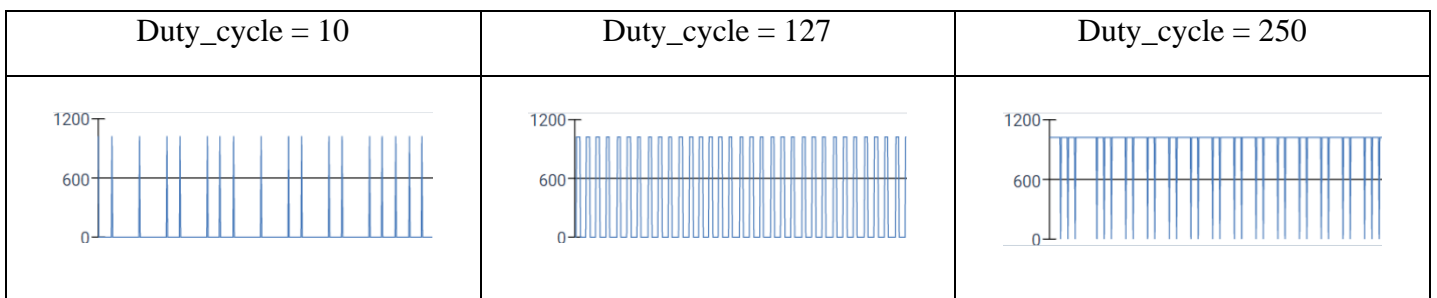
- Simulation d'un signal PWM

Pour cette simulation, nous avons utilisé le logiciel Tinkercad en faisant le schéma du montage comme on peut le voir ci-dessous. Ici, nous avons utilisé deux bibliothèques précises. La bibliothèque « `analogWrite()` » qui permet de créer un signal créneau avec une tension comprise entre 0 et 5 V et pouvant faire varier le rapport cyclique entre 0 et 255. La bibliothèque « `analogRead()` » quant à elle fait une lecture de la valeur du signal qu'elle reçoit, avec des valeurs comprises entre 0 et 1023. Dans notre cas ci-dessous par exemple, 0 correspond à une tension de 0 V et 1023 correspond à une tension de 5 V.



```
void setup()
{
  Serial.begin(9600);
  pinMode(5, OUTPUT);
  pinMode(A0, INPUT);
  analogWrite(5,Duty_cycle);
}

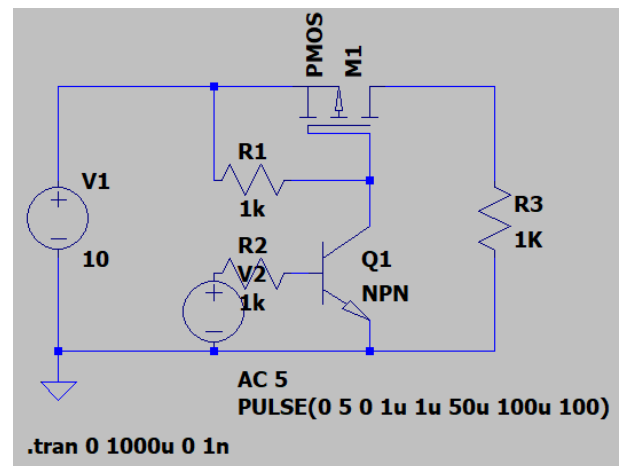
void loop()
{
  Serial.println(analogRead(A0));
}
```



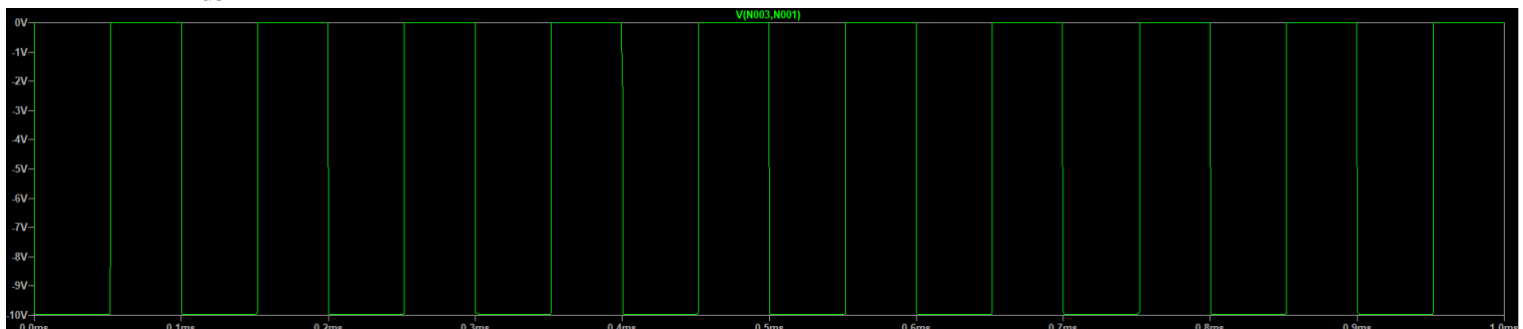
On a donc ici réussi à créer un signal PWM avec un rapport cyclique que l'on peut faire varier.

III. Étude de la partie hacheur

La partie hacheur nous permet de modéliser un interrupteur électronique. Dans notre cas, notre hacheur est composé d'un signal PWM, associé à deux transistors comme on peut le voir sur le schéma ci-dessous effectué grâce au logiciel LTspice :



On a sur V_{GS} la courbe suivante :

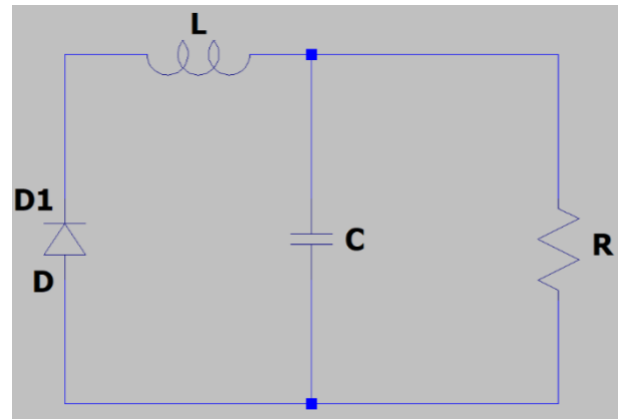


Ceci s'explique par le fait que lorsque le PWM est à 0 V, $V_{GS} = 0$ V et se comporte comme un interrupteur ouvert et lorsque le PWM est à 5 V, $V_{GS} = -10$ V. On a donc créé un interrupteur électronique.

IV. Étude du filtre aux bornes de la résistance de charge

A la sortie V_{out} , nous avons un filtre RLC associé à une diode comme ci-dessous :

Dans toute cette étude nous utilisons une valeur de résistance $R = 100 \Omega$



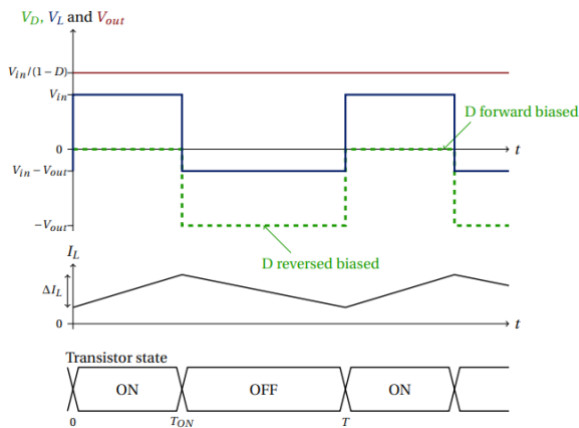
- Fonctionnement de la diode

La diode fonctionne selon deux états, ON et OFF, qui dépend du sens de circulation du courant à travers la diode. La diode est passante, c'est-à-dire à l'état ON si le courant circule du pôle positif vers le pôle négatif. Si le courant circule dans le sens contraire, la diode se comporte comme un circuit ouvert et est dans l'état OFF.

Lorsque le transistor est passant, la diode est à l'état OFF. Lorsque le transistor est bloquant, le courant induit du fait que l'énergie magnétique emmagasinée dans la bobine est alors relâchée, le courant circule de la cathode vers l'anode et la diode est donc à l'état ON.

L'utilisation de la diode est ainsi très utile pour pouvoir faire circuler le courant lorsque le transistor est bloquant, sans avoir d'impact lorsque le transistor est passant.

- Fonctionnement de la bobine :



Dans la bobine, l'intensité est faite d'une valeur moyenne $\langle I_L \rangle = \frac{D \cdot V_{in}}{R_L}$. Ainsi, le courant qui traverse la bobine varie linéairement en fonction du temps avec deux phases :

- État haut : Le courant dans l'inductance croît.
- État bas : Le courant dans l'inductance décroît.

La variation d'intensité dans la bobine $\Delta I_L = \frac{V_{in}(1-D)D}{L \cdot f}$ avec f la fréquence de commutation du PWM. Ainsi, plus L est grand, plus on tendra vers la situation idéale d'une tension avec des ondulations négligeables.

- Dimensionnement de la bobine :

Dans un premier temps, pour dimensionner la bobine, il faut donc faire l'étude du circuit sans le condensateur.

$L = 5 \text{ mH}$	$L = 12 \text{ mH}$	$L = 25 \text{ mH}$

Expérimentalement, on détermine cette valeur de L, pour laquelle on obtient un signal triangle. Si L est trop petit, la bobine n'a pas le temps de se décharger complètement ainsi on voit l'apparition des paliers, ce qui ne correspond pas à ce qu'on cherche.

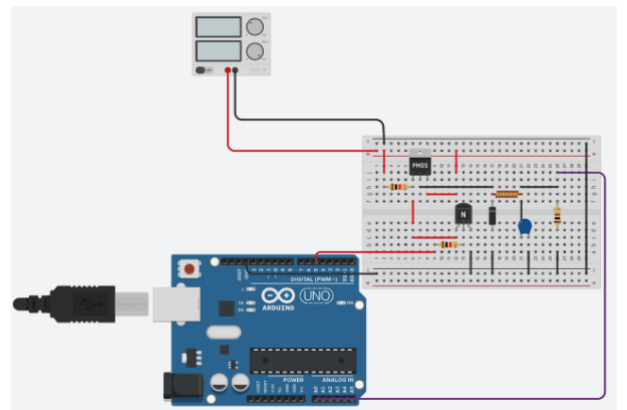
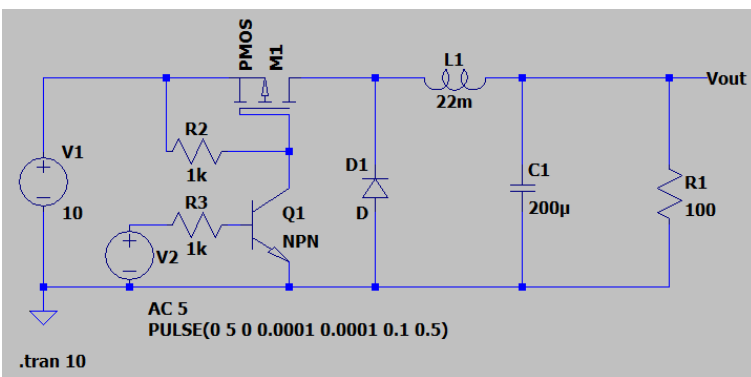
A partir de cette simulation on peut donc que pour $L = 5 \text{ mH}$, on est dans un cas « instable ». Pour $L = 12 \text{ mH}$, on est à la limite de l'instabilité ce qui fait de cette valeur, la valeur minimale qu'on puisse imposer. Et on remarque qu'à $L = 25 \text{ mH}$, on a un signal triangle sans palier.

- Fonctionnement du condensateur

Le condensateur a une influence directe sur la stabilité et l'ondulation de tension. Il permet de transformer la tension qui a une forme de triangle en une tension moyennée et d'obtenir ainsi une tension V_{out} quasi constante et valant $V_{out} = D * V_{in}$.

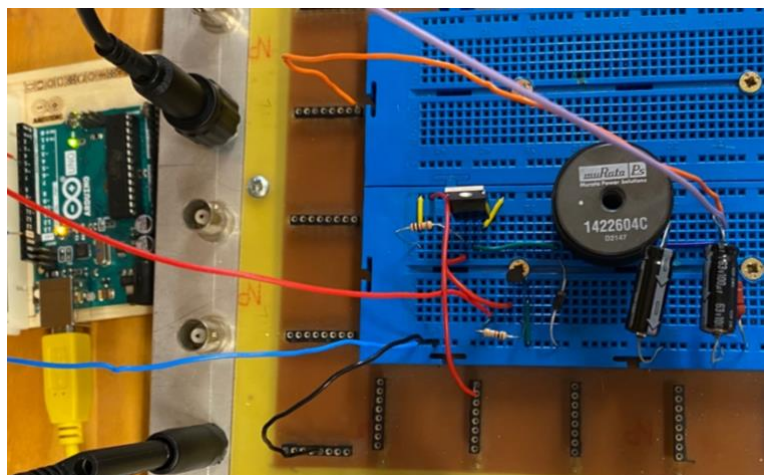
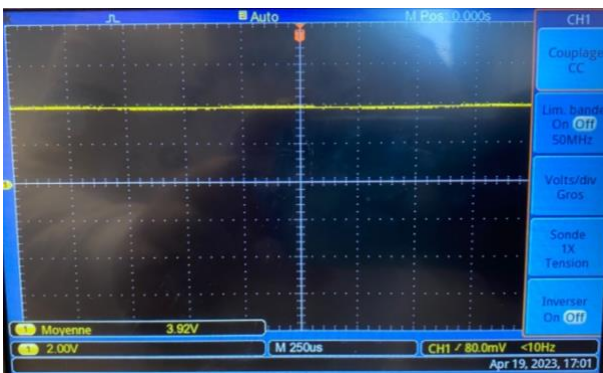
Le couple de composants L et C forment un filtre passe-bas, dont la fréquence de coupure $f_c = \frac{1}{2\pi\sqrt{LC}}$ est volontairement nettement inférieure à la fréquence de commutation des interrupteurs. En conséquence, l'onde fondamentale et les harmoniques constituant la tension de sortie sont assez atténués pour devenir d'amplitudes négligeable devant la valeur moyenne de cette tension. On obtient alors la tension de sortie voulue, continue et constante. En fait, $\Delta V_{out} = \frac{V_{in}(1-D)D}{8*L*C*f^2}$ qui est très négligeable devant $V_{out} = D * V_{in}$. Le choix de C se fait donc sur le critère de la comparaison entre fréquence de coupure du filtre LC et fréquence de commutation du PWM.

V. Circuit complet



- Montage du circuit

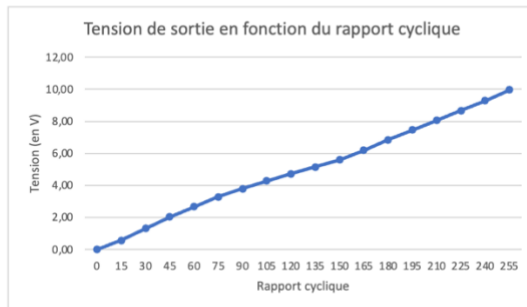
Ici on peut voir à partir de l'oscilloscope que le signal de la tension de sortie à une ondulation presque négligeable, que l'on peut assimiler à un signal continue comme on le voulait. Dans ce montage on a utilisé deux condensateurs de $100 \mu\text{F}$ pour lisser au plus le signal avec un rapport cyclique d'environ 100 soit un ratio de 39,2%.



VI. Vérification et amélioration du montage

- Tension de sortie en fonction du rapport cyclique

Tension de sortie (V)	Rapport cyclique
0,00	0
0,57	15
1,32	30
2,04	45
2,66	60
3,28	75
3,80	90
4,29	105
4,72	120
5,14	135
5,60	150
6,19	165
6,84	180
7,46	195
8,06	210
8,66	225
9,29	240
9,96	255

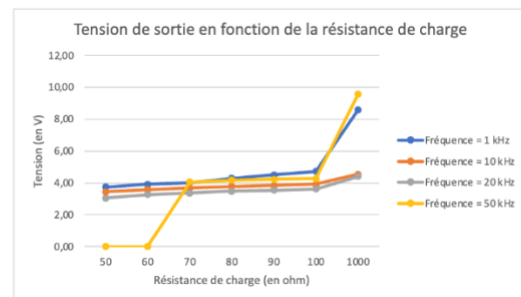


On remarque ici que la droite représentant la tension de sortie en fonction du rapport cyclique est linéaire. Ceci nous montre que selon le rapport cyclique qu'on choisit, la tension de sortie varie proportionnellement. Ce qui est mieux même c'est qu'on peut donc viser une tension de sortie bien précise en faisant varier le rapport cyclique.

- Tension de sortie en fonction de la résistance de charge

Dans cette partie, on se place dans le cas où notre rapport cyclique est de 50% soit de 127 sur Arduino. On remarque donc que plus la fréquence augmente plus les valeurs tension à la sortie sont rapprochées pour des résistances comprises entre 70 et 100 Ω . Ce qui n'est pas cohérent dans ces résultats c'est le fait que pour 50 kHz, la tension augmente alors qu'on a un passe-bas ce qui devrait plus atténuer à haute fréquence à moyenne fréquence.

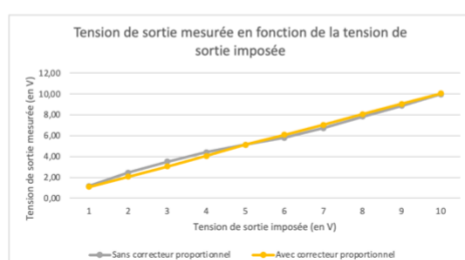
Tension de sortie (V)				Résistance de charge (ohm)
Fréquence = 1 kHz	Fréquence = 10 kHz	Fréquence = 20 kHz	Fréquence = 50 kHz	
3,74	3,45	3,05	0,00	50
3,91	3,58	3,26	0,00	60
4,02	3,68	3,37	4,05	70
4,29	3,77	3,48	4,17	80
4,50	3,84	3,54	4,22	90
4,71	3,91	3,61	4,27	100
8.60	4.54	4.40	9.58	1000



- Utilisation d'un signal PWM avec un correcteur proportionnel

Le but dans cette partie est de créer un correcteur proportionnel afin de mettre une boucle de rétroaction sur notre système est de corriger les erreurs qui peuvent survenir. Ce correcteur va pouvoir faire varier la valeur du rapport cyclique dans le PWM, pour cela il faut comparer la sortie du système à la valeur de sortie imposée. Pour cela nous allons utiliser le code dans l'annexe.

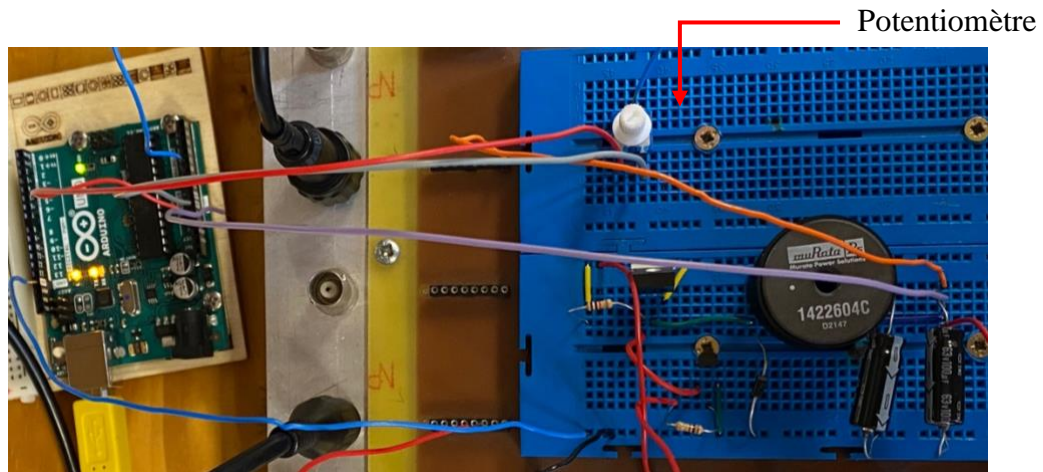
Tension de sortie imposée (en V)	Rapport cyclique pour avoir la sortie imposée	Tension de sortie mesurée (en V)	Tension de sortie mesurée (en V)
		Sans correcteur proportionnel	Avec correcteur proportionnel
1	25	1,16	1,08
2	51	2,45	2,07
3	76	3,50	3,04
4	102	4,41	4,04
5	127	5,14	5,14
6	153	5,81	6,08
7	178	6,74	7,04
8	204	7,83	8,07
9	229	8,86	9,04
10	255	9,96	



On remarque ici que l'erreur en la valeur imposée et la valeur mesurée est plus petite avec un correcteur proportionnel que sans. On obtient donc une droite plus linéaire ce qui est mieux en termes de précision de système.

- Mise en place d'un potentiomètre

Le potentiomètre nous permet de faire varier la tension de sortie que l'on désire sans toucher au code Arduino, il n'est donc juste là que pour l'aspect pratique de l'utilisation d'un convertisseur buck et n'a pas d'aspect technique. L'utilisation du potentiomètre impose quand même un code supplémentaire que l'on trouvera dans l'annexe.



Conclusion

En conclusion, dans ce projet nous avons dû nous familiariser avec la carte d'acquisition Arduino et utiliser les logiciels qui sont fournis avec, entre autres le code Arduino et le logiciel de simulation Tinkercad. Notre but était de pouvoir abaisser une tension continue, nous avons donc créé un convertisseur abaisseur de tension pour atteindre la tension qu'on voulait. Au début du projet, on avait une tension continue d'entrée de 10 V que l'on voulait abaisser à 5 V. On a pu voir tout au long du projet que le convertisseur abaisseur de tension à une gamme de tension de sortie comprise entre 1 et 9V.

Annexe

```
int PWM = 5;
int v_in = 10;
int v_out;

void setup()
{
  Serial.begin(9600);
  pinMode(PWM, OUTPUT);
  pinMode(A0, INPUT);
  analogWrite(PWM, 127);
}

void loop()
{
  Serial.println(analogRead(A0));
  delay(100);
}
```

Titre : Code Arduino pour créer un signal PWM avec lecteur de la tension de sortie du convertisseur

```
int PWM = 5;
int v_in = 10;
int v_out_max = 4;
int v_out;
int rc=128;
float dv=0.1;
float error;
float p=0.1;

void setup()
{
  Serial.begin(9600);
  pinMode(PWM, OUTPUT);
  pinMode(A0, INPUT);
  analogWrite(PWM, rc);
  delay(100);
}

void loop()
{
  v_out = analogRead(A0);
  float v_out_map = (5.0/1023)*v_out;
  error = v_out_map - v_out_max;
  if(abs(error)>dv){
    rc = int(rc - p*error*255/v_in);
    analogWrite(PWM, rc);
  }
  Serial.println(v_out_map);
  delay(100);
}
```

Titre : Code Arduino pour créer un signal PWM avec un correcteur proportionnel

```
int PWM = 5;
int v_in = 10;
//int v_out_max = 4;
int duty_cycle=128;
float dv=0.1;
float p=0.1;

void setup()
{
  Serial.begin(9600);
  pinMode(PWM, OUTPUT);
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  analogWrite(PWM, duty_cycle);
  delay(100);
}

void loop()
{
  float v_out_max = (5.0/1023)*analogRead(A1);
  int v_out = analogRead(A0);
  float v_out_map = (5.0/1023)*v_out;
  float error = v_out_map - v_out_max;
  if(abs(error)>dv){
    duty_cycle = int(duty_cycle - p*error*255/v_in);
    analogWrite(PWM, duty_cycle);
  }
  Serial.println(v_out_max);
  delay(100);
}
```

Titre : Code Arduino pour l'implémentation d'un potentiomètre au système