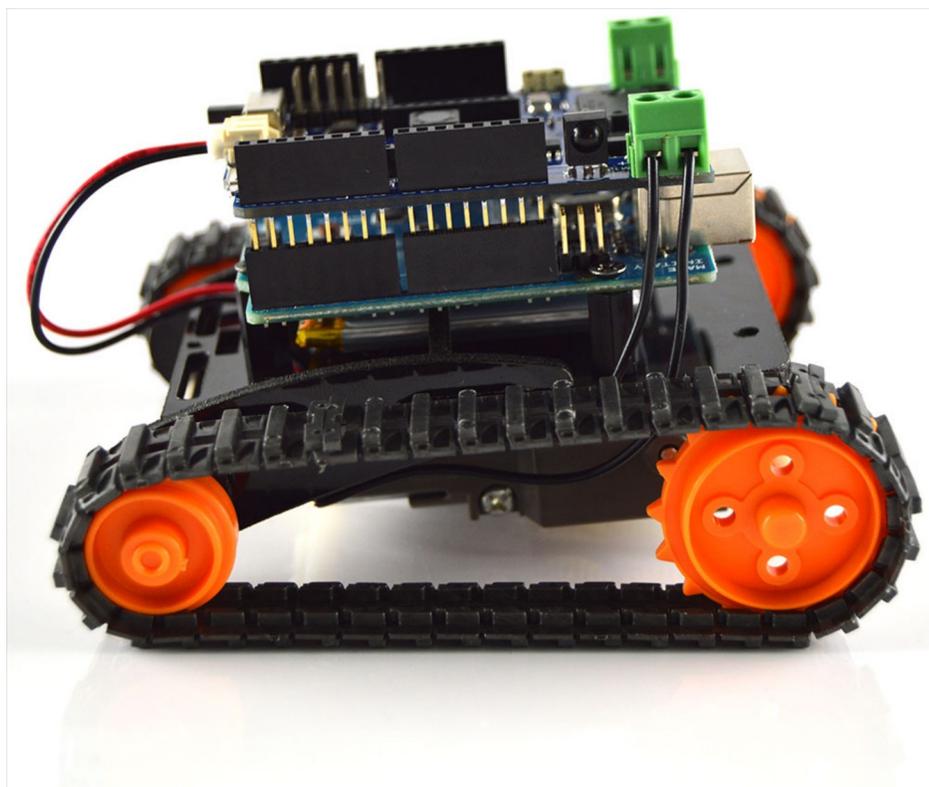




## MINI ROVER



Révision 1.1B (août 2021)

[www.RobotShop.com](http://www.RobotShop.com)

# Table des matières

<b>Table des matières</b>	<b>1</b>
<b>À propos</b>	<b>2</b>
Ce qui est inclus	2
<b>Boîte à Engrenages</b>	<b>3</b>
Assemblage de la Configuration A	3
<b>Montage du microcontrôleur</b>	<b>13</b>
Arduino UNO	13
<b>Blindage Rover DFRobotShop</b>	<b>19</b>
Principales caractéristiques	20
Brochage	24
<b>Arduino</b>	<b>25</b>
Installation logiciel & Configuration	25
<b>Exemple de Code</b>	<b>26</b>
Avance à pleine vitesse	26
Contrôle moteur W/A/S/D	26
Encodeurs (en option)	29
Détermination des codes IR	30
Exemple de code IR avec contrôle moteur	30

# À propos

Le Kit Mini Rover DFRobotShop (Arduino UNO) est une petite plateforme à chenilles idéale pour expérimenter avec la programmation Arduino et les robots mobiles. C'est un robot polyvalent qui utilise la boîte à engrenages double ainsi que le jeu de chenilles et de roues Tamiya. Il comprend un microcontrôleur Arduino UNO compatible avec les blindages, le blindage DFRobotShop Rover et une batterie au lithium.

Vous trouverez ci-dessous un guide étape par étape pour l'assemblage de la boîte à engrenages et du châssis. Notez que, bien que même si la boîte à engrenages double de Tamiya permet trois configurations différentes, elle doit être assemblée en configuration A pour ce kit.

## Ce qui est inclus

### Electronique

1x microcontrôleur Arduino UNO  
1x Blindage pour DFRobotShop Rover  
1x batterie LiPo 3.7V 1000mAh

### Matériel

8x vis à métaux à tête cylindrique bombée 4-40 x 3/8"  
4x entretoises hexagonale en nylon 4-40 x 1/2" F/F (pour le montage Arduino)  
2x entretoises hexagonale en nylon 4-40 x 3/4" F/F (pour montage Raspberry Pi)  
1x ruban adhésif double face (option de montage de batterie)  
1x 62cm de fil de branchement noir 22 AWG  
5x broches à sertir femelles

### Chenilles / Moteurs

1x boîte à Engrenages double Tamiya  
1x kit de chenilles et pignons Tamiya

### Pièces de châssis en lexan

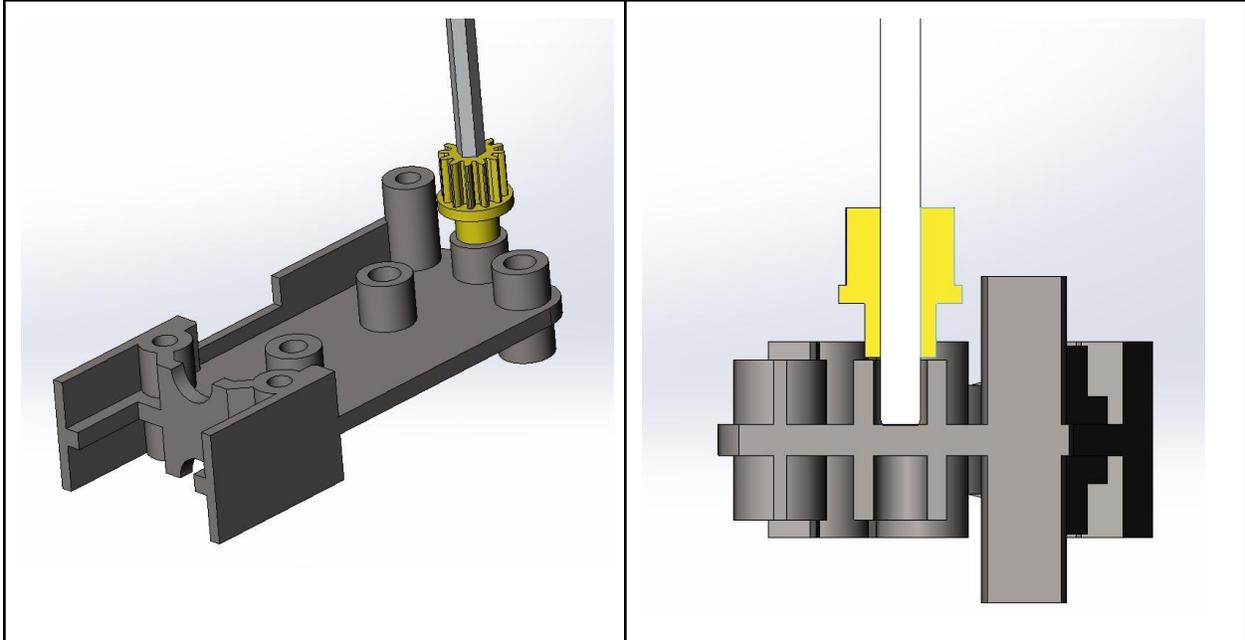
1x plaque principale  
2x plaques latérales  
1x plaque de support supérieure  
1x plaque de support inférieure

*Remarque : Le Lexan est recouvert des deux côtés d'un mince film de plastique utilisé pour protéger la surface pendant la découpe au laser. Il peut être retiré comme un autocollant.*

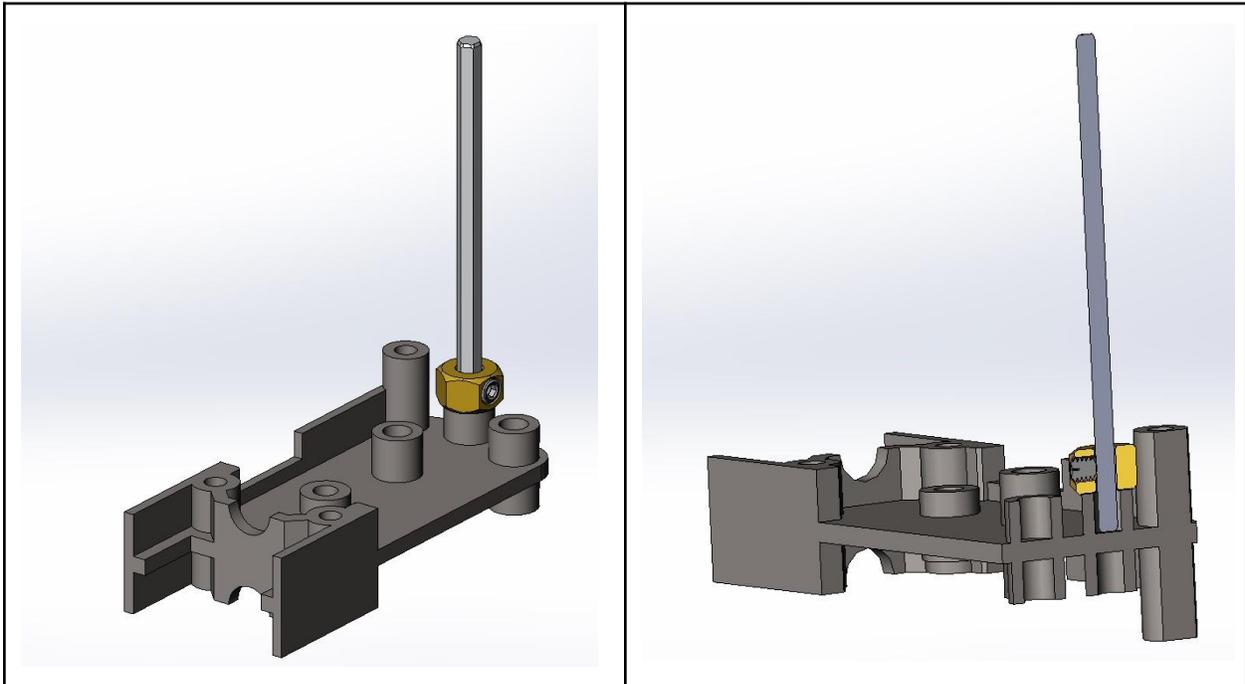
# Boîte à Engrenages

## Assemblage de la Configuration A

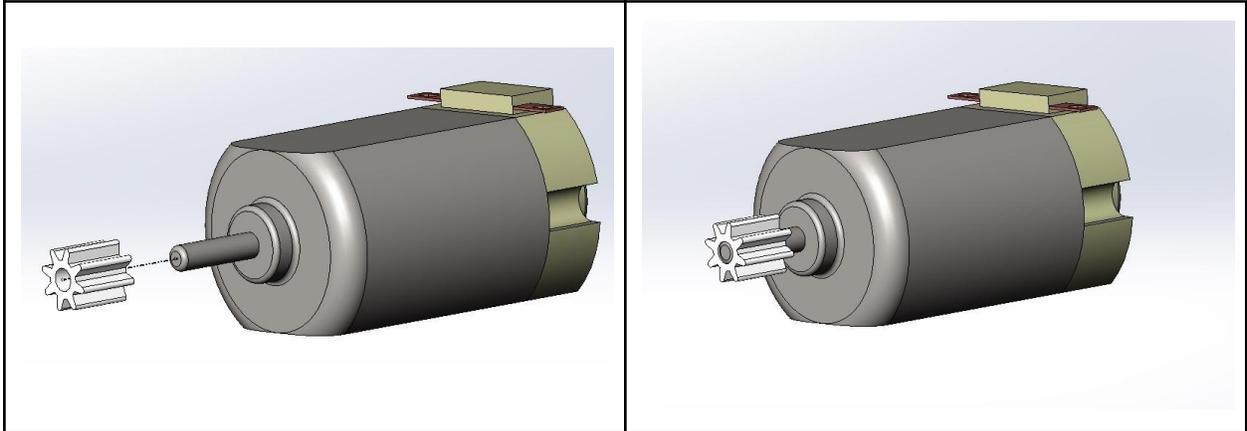
### Étape 1A (2x) : Positionnement du pignon d'entraînement



### ÉTAPE 1B : Positionnement du collier de l'essieu central (avec vis de réglage)



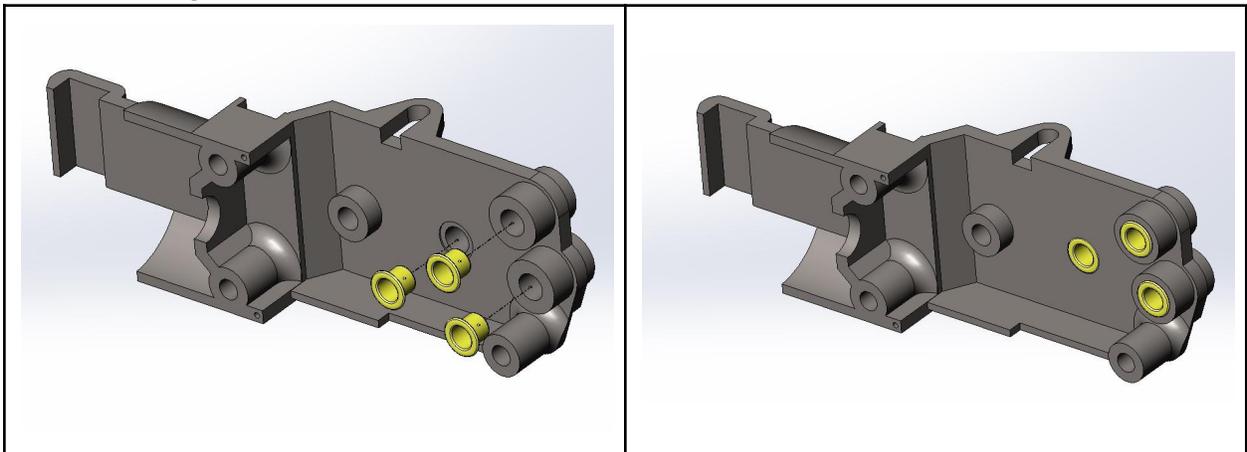
## ÉTAPE 2 (2x) : Pignon Moteur



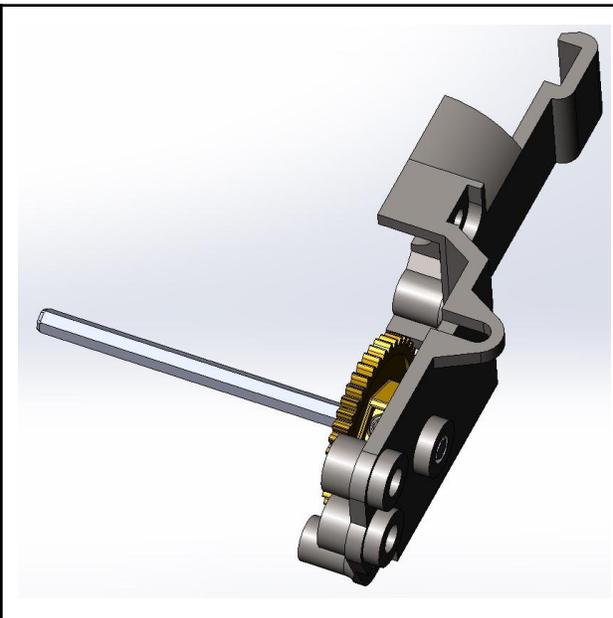
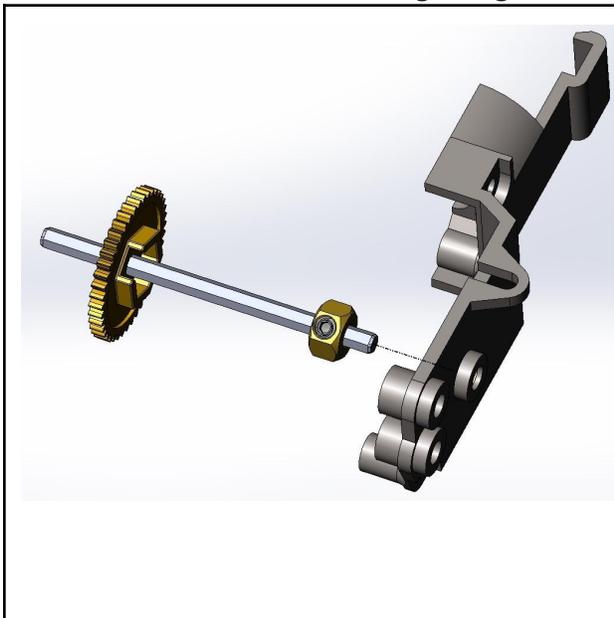
*Remarque : À ce stade, vous pouvez également attacher / souder des fils au moteur. Nous suggérons que les fils mesurent au moins 14 cm de long. Il y a cinq sertissages inclus avec le kit ainsi que 62cm de fil 22 AWG. Vous pouvez couper le fil en quatre sections égales, dénuder environ 5 mm de fil à chaque extrémité et utiliser les sertissages entre une extrémité du fil et les bornes du moteur. Alternativement, vous pouvez utiliser vos propres matériaux.*

*Les bornes du moteur sont très fragiles - ne les soumettez pas à une pression trop forte ou à des flexions répétées, car elles peuvent se briser.*

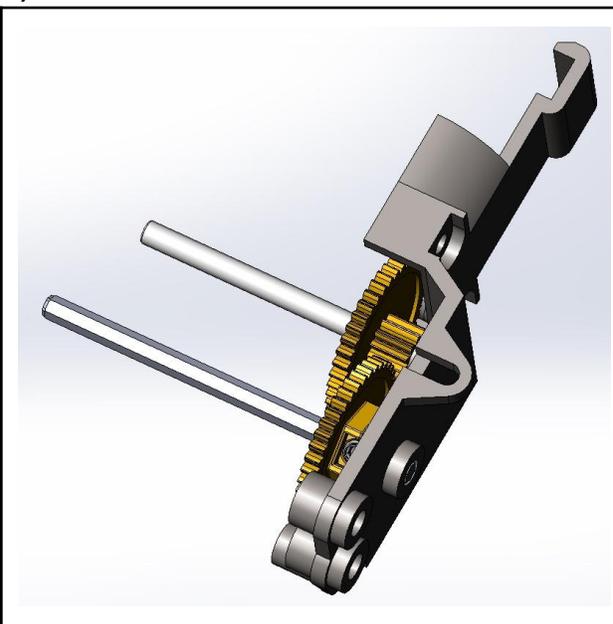
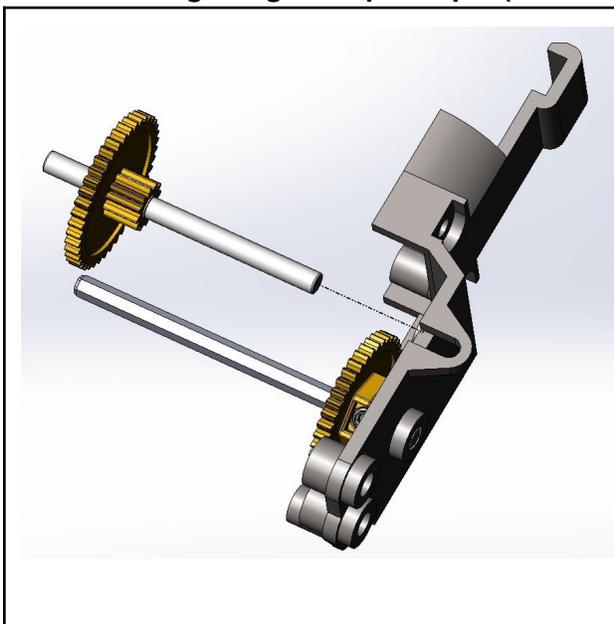
## ÉTAPE 3 : Bague en laiton



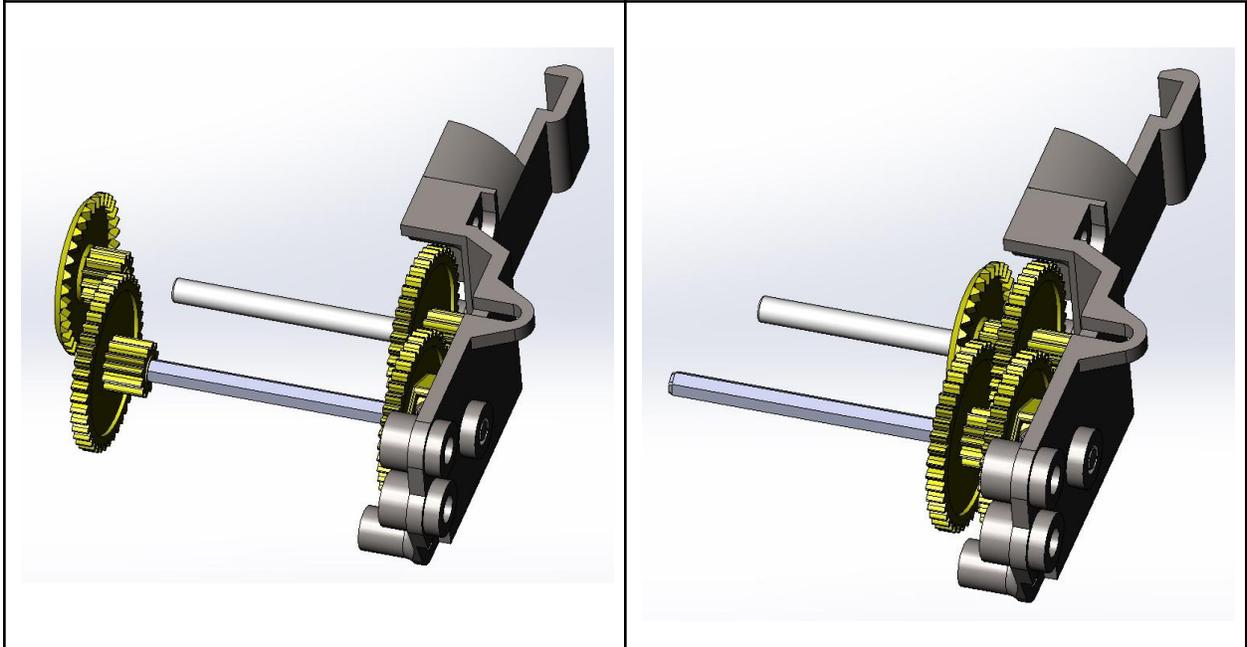
**ÉTAPE 4 : Essieu central + engrenage**



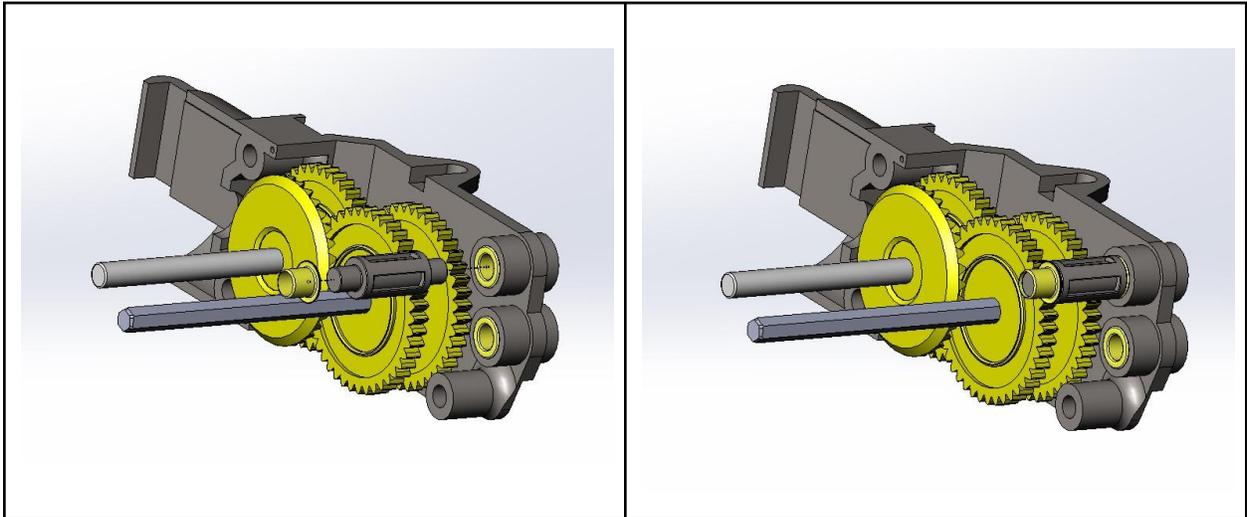
**ÉTAPE 5 : Engrenages en plastique (côté droit)**



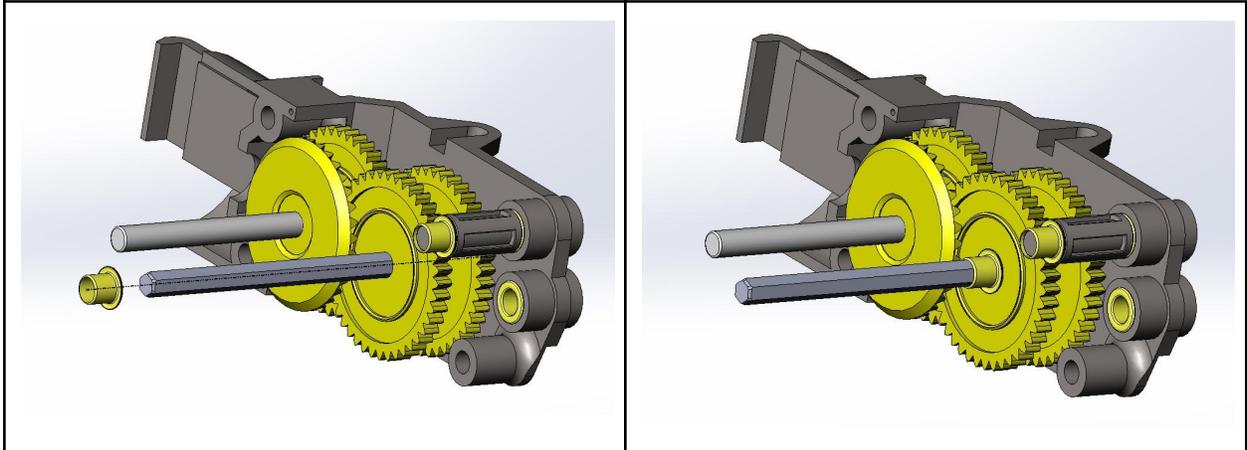
**ÉTAPE 6 : Engrenages en plastique (côté droit)**



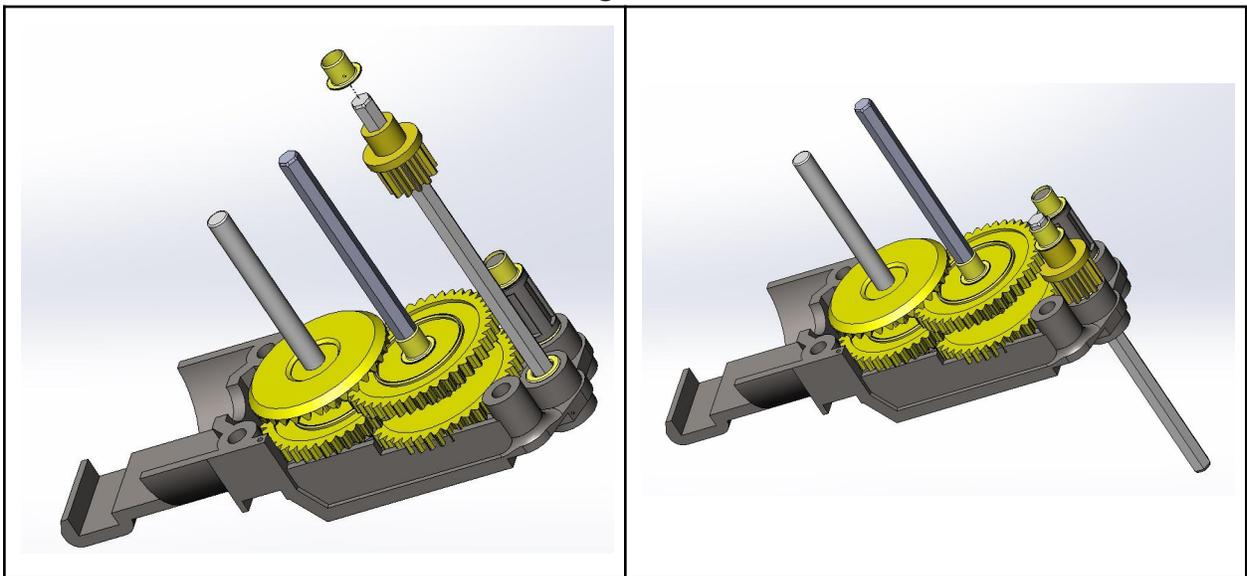
**ÉTAPE 7 : Entretoises en plastique + bague en laiton**



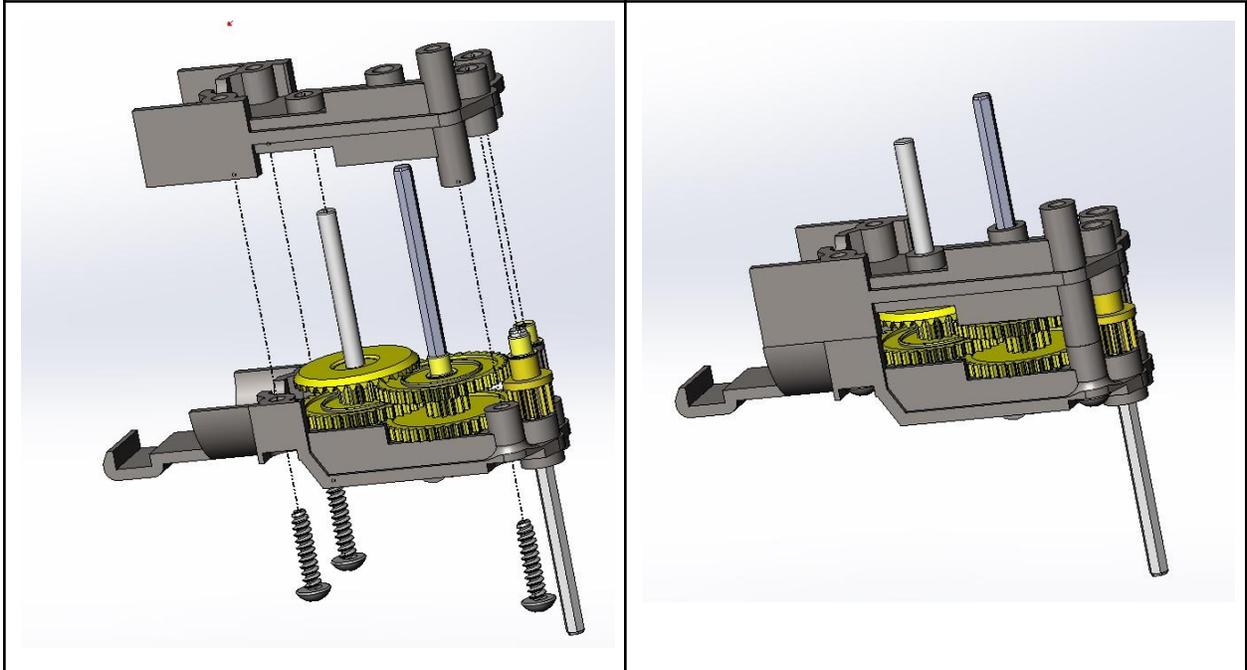
**ÉTAPE 8 : Bague en laiton**



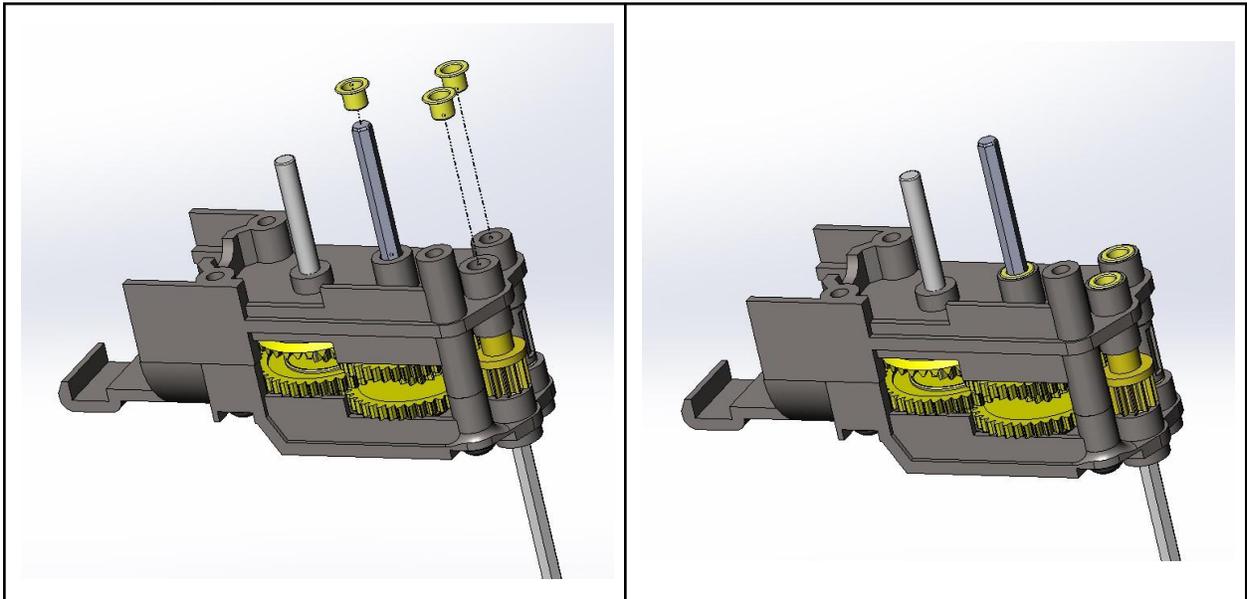
**ÉTAPE 9 : Arbre de transmission droit + bague en laiton**



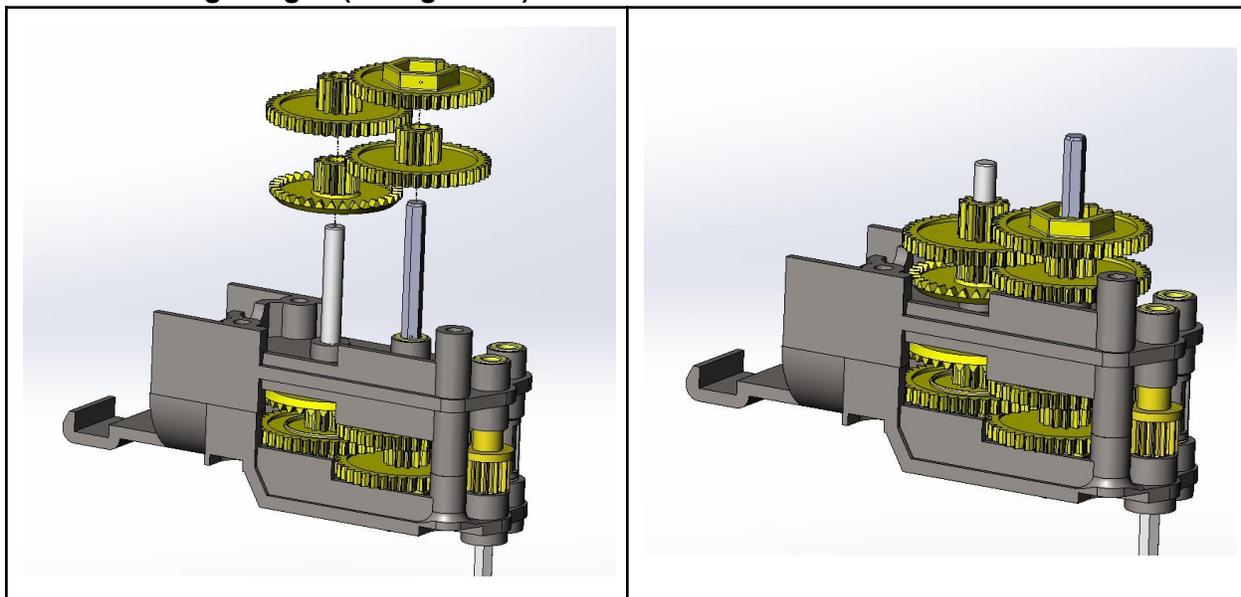
### ÉTAPE 10 : Support central



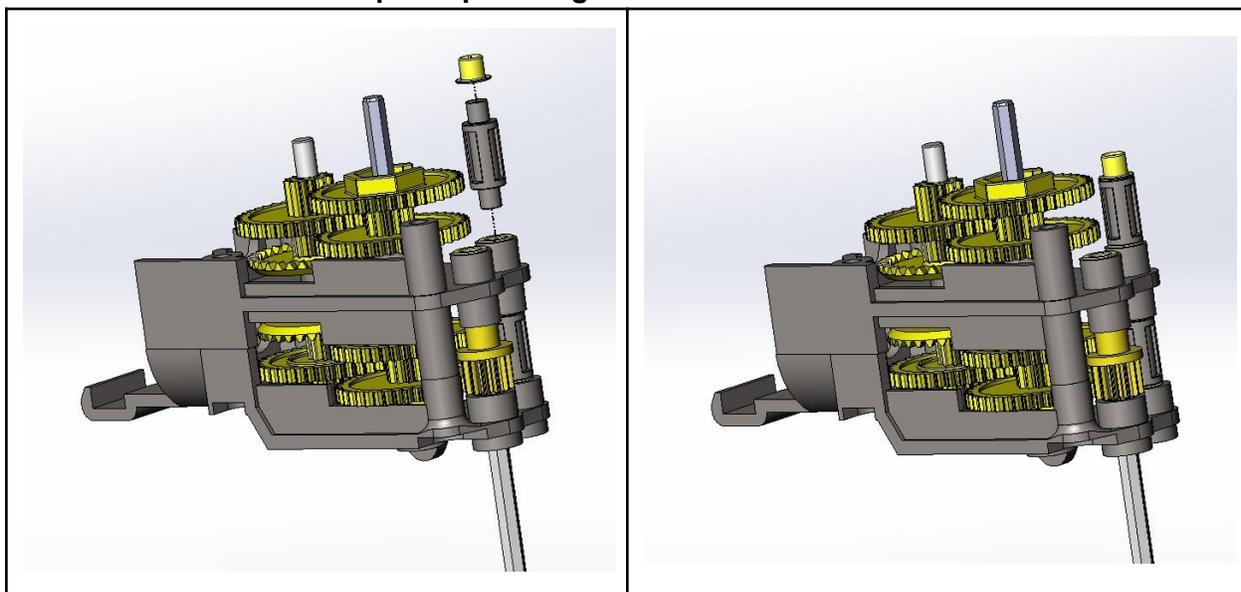
### ÉTAPE 11 : Bagues en laiton



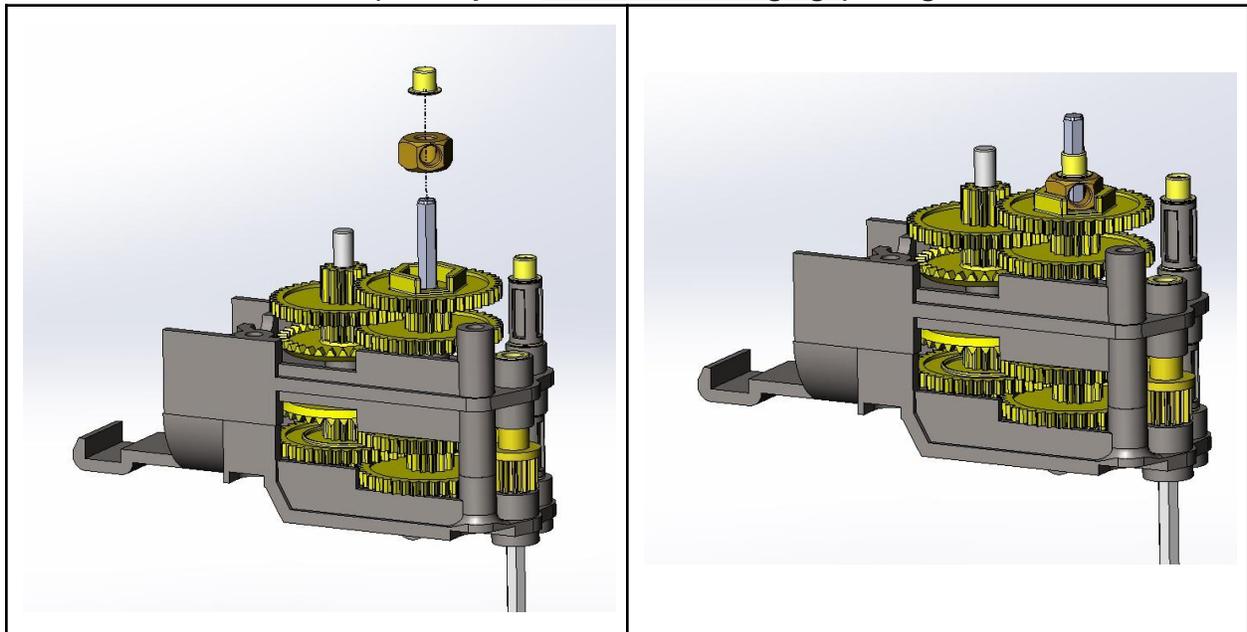
**ÉTAPE 12 : Engrenages (côté gauche)**



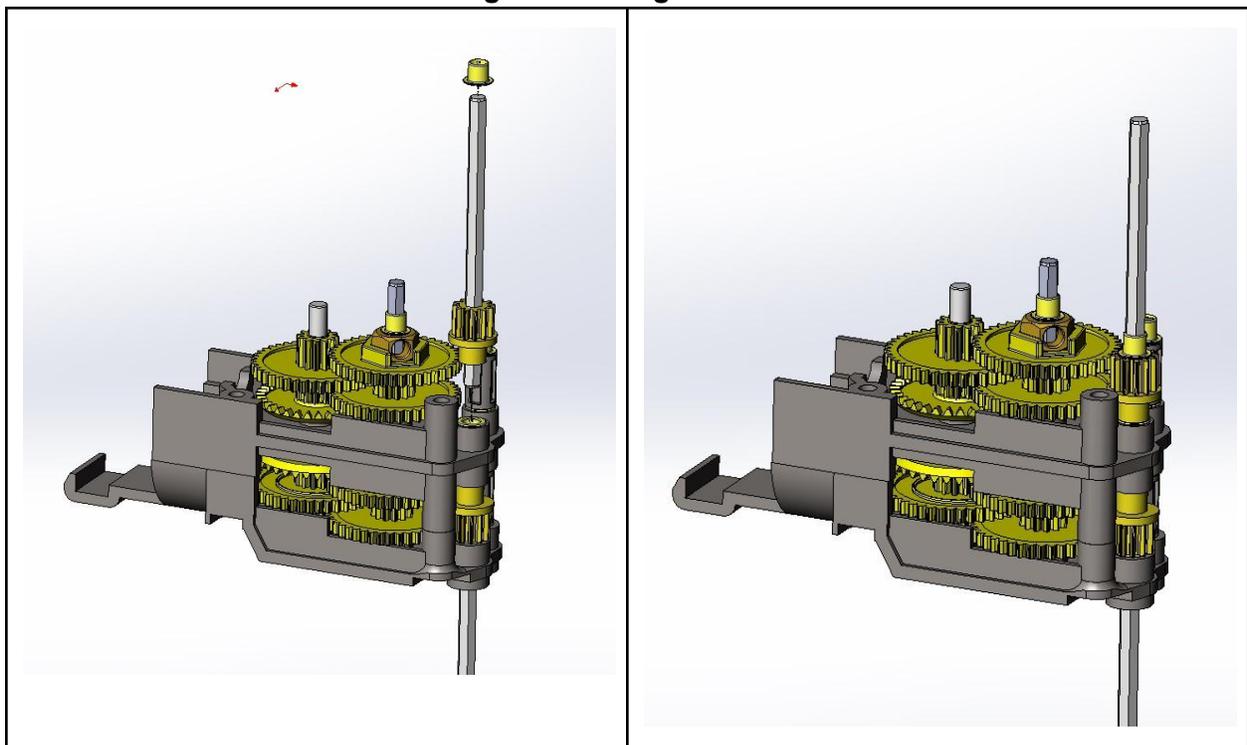
**ÉTAPE 13 : Entretoises en plastique + bague en laiton**



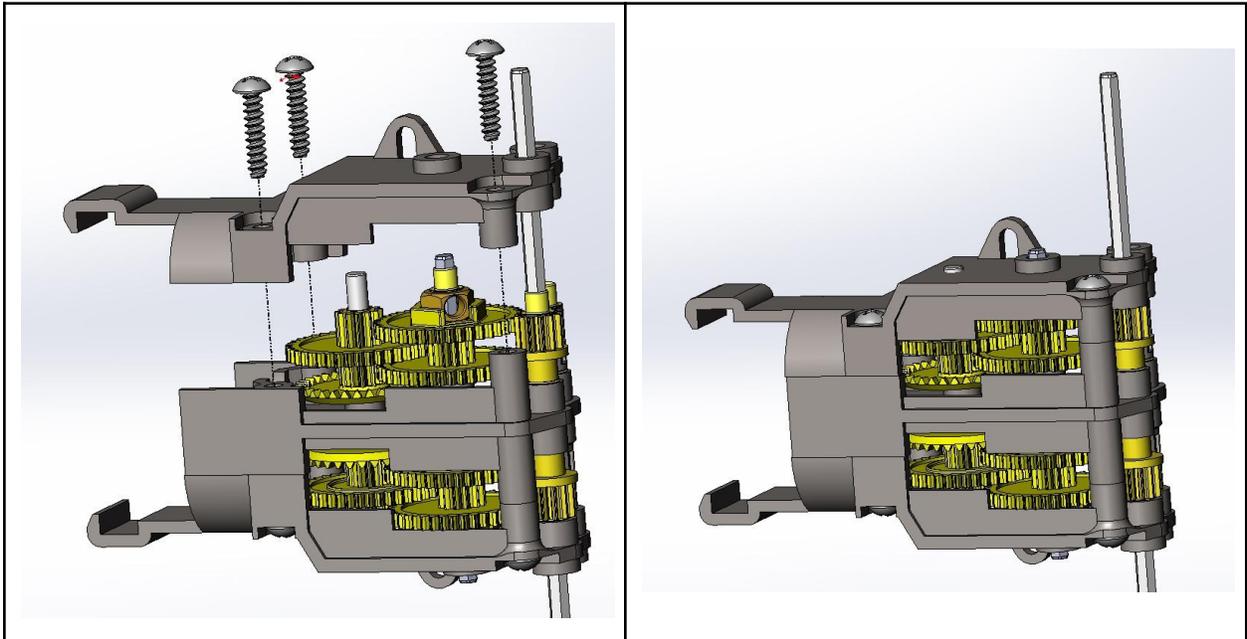
**ÉTAPE 14 : Collier d'arbre (remarque : PAS de vis de réglage) + bague en laiton**



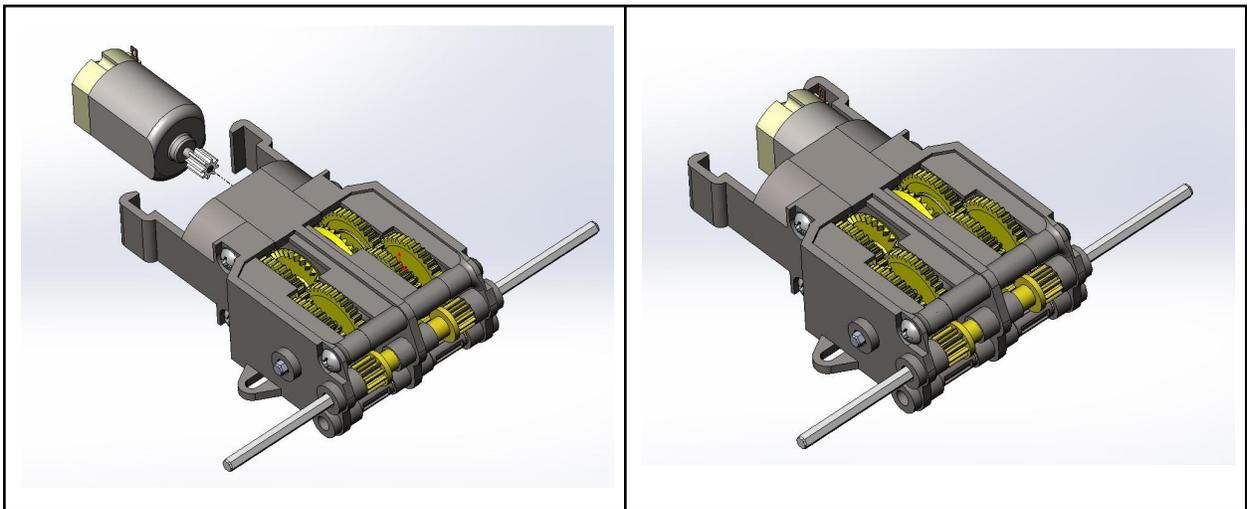
**ÉTAPE 15 : Arbre de transmission gauche + bague en laiton**



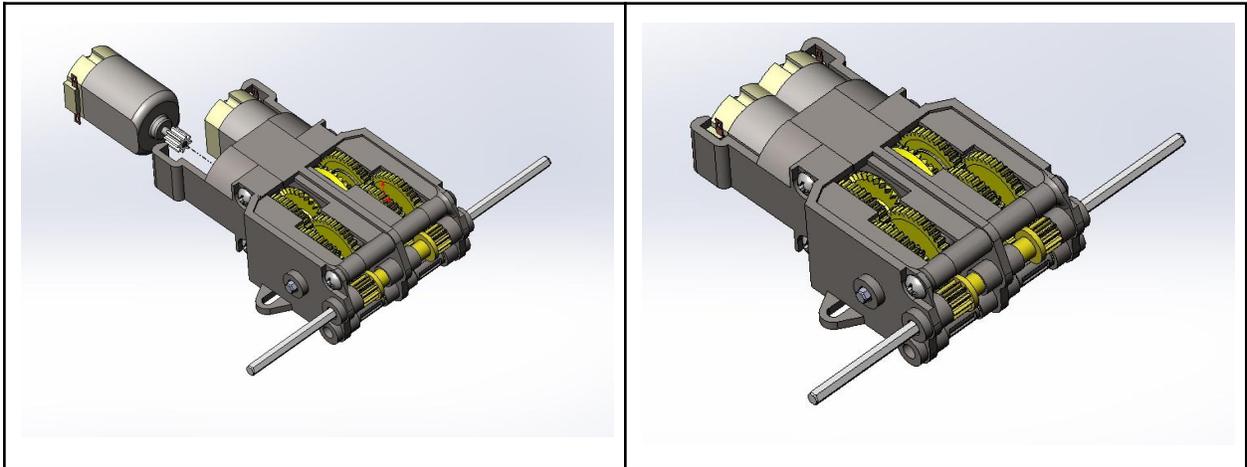
### ÉTAPE 16 : Support gauche



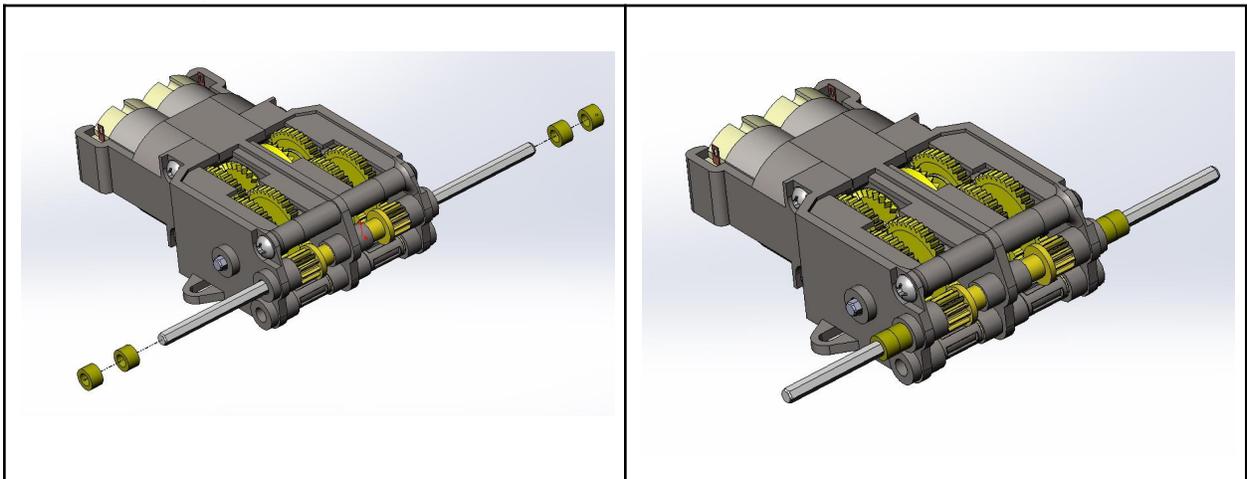
### ÉTAPE 17 : Moteurs CC



### ÉTAPE 18 : Moteurs CC

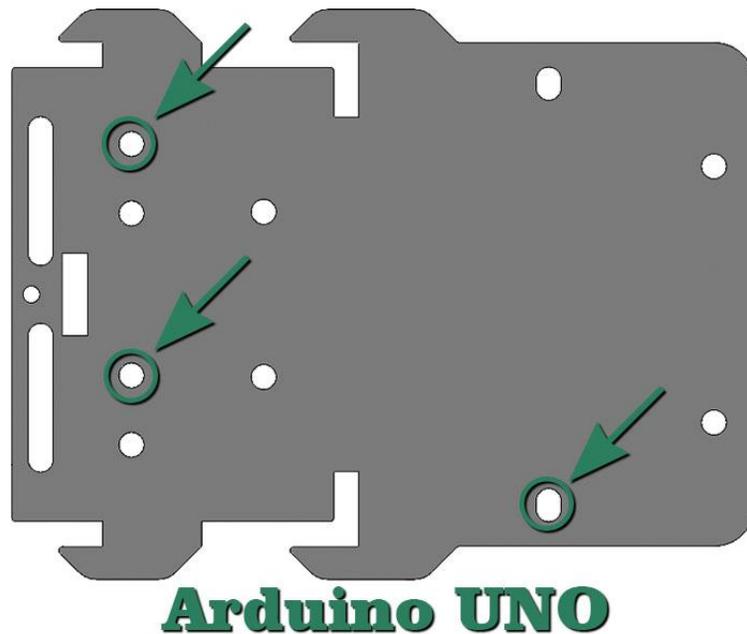


### ÉTAPE 19 : Espaceurs



## Montage du microcontrôleur

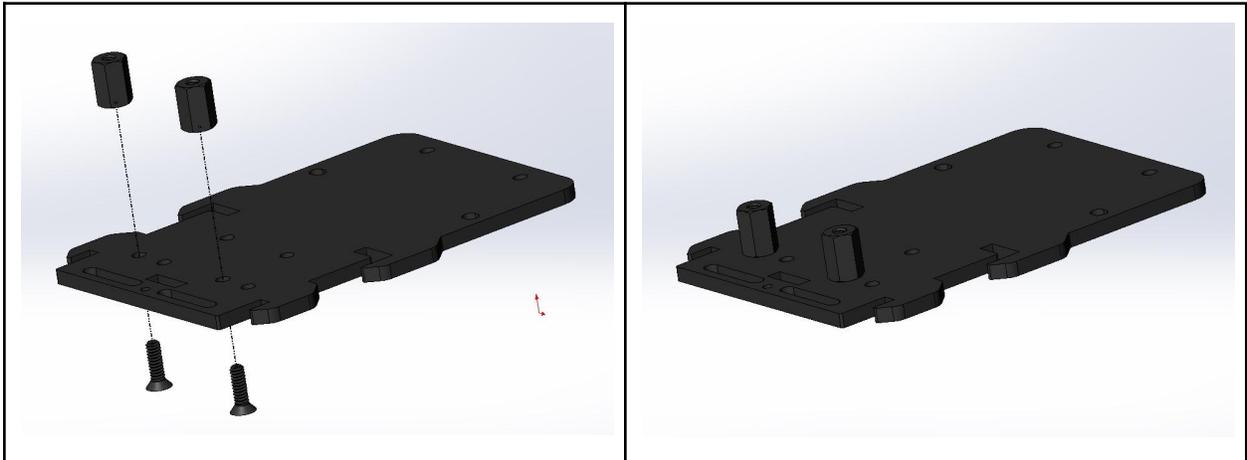
Les emplacements des entretoises hexagonales sont déterminées en fonction de la carte que vous prévoyez utiliser. Le Mini RobotShop Rover a des trous de montage pour un Arduino UNO ou un Raspberry Pi et la plaque est symétrique.



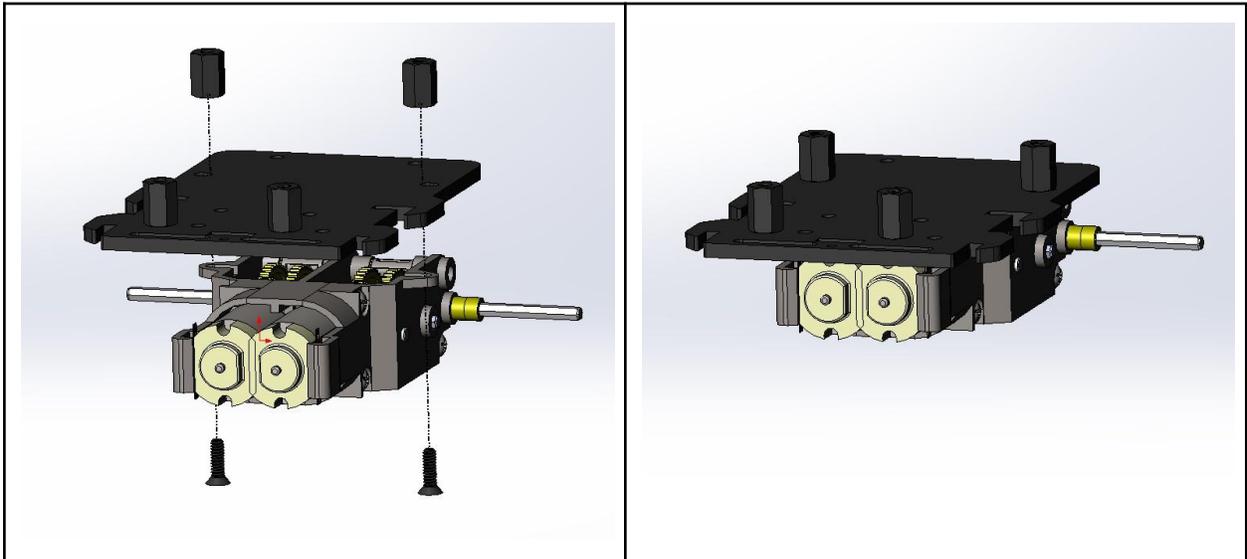
### Arduino UNO

Le montage Arduino utilise trois (ou quatre) entretoises hexagonales, dont l'une est partagée avec le montage de la boîte à engrenages double Tamiya (utilisée à la place de la vis et de l'écrou de 3 mm inclus avec la boîte à engrenages double Tamiya). Le connecteur USB / connecteur à baril fait face à l'arrière de la carte. Le guide ci-dessous montre comment installer un Arduino UNO.

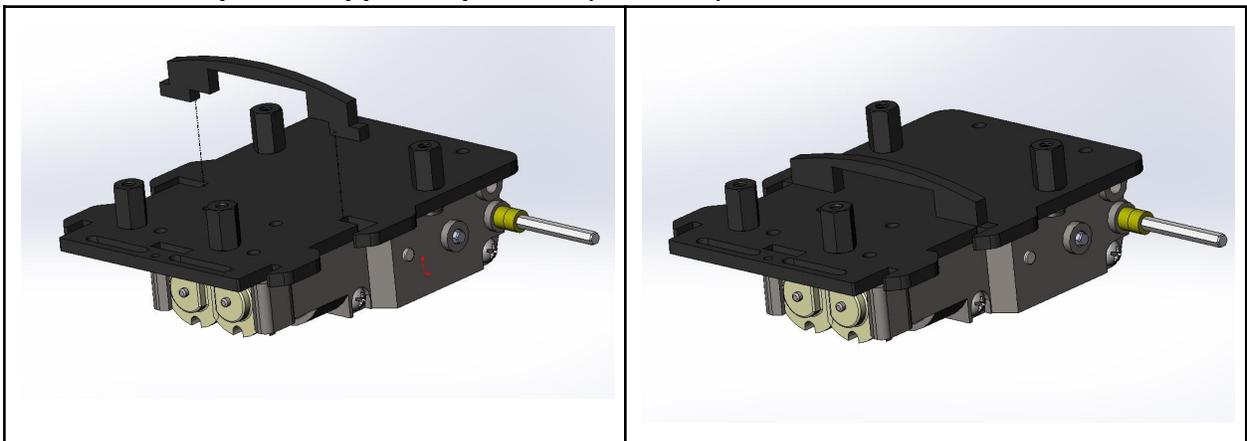
### ÉTAPE 20 : Entretoises pour Arduino



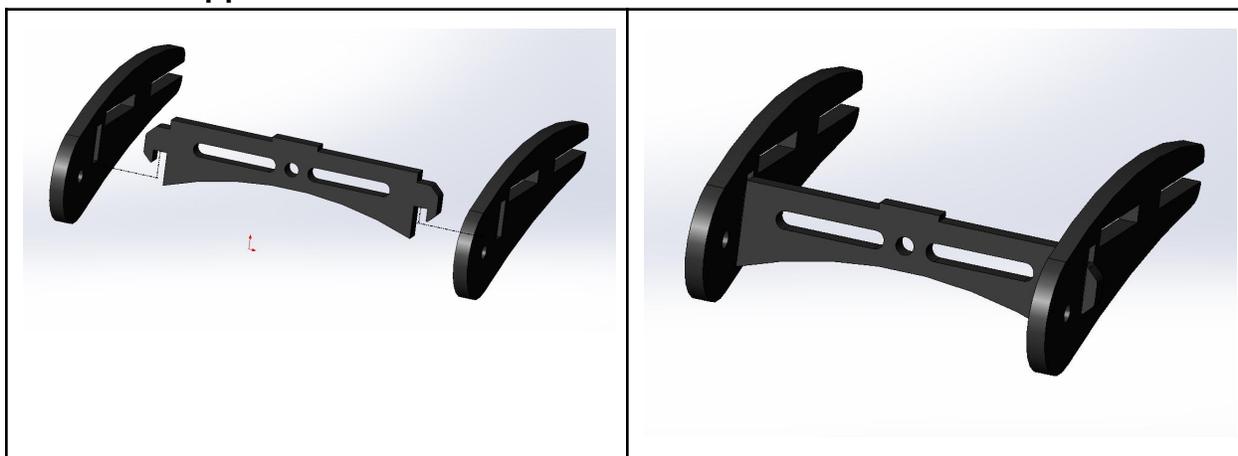
### ÉTAPE 21 : Plaque de base à la boîte à engrenages



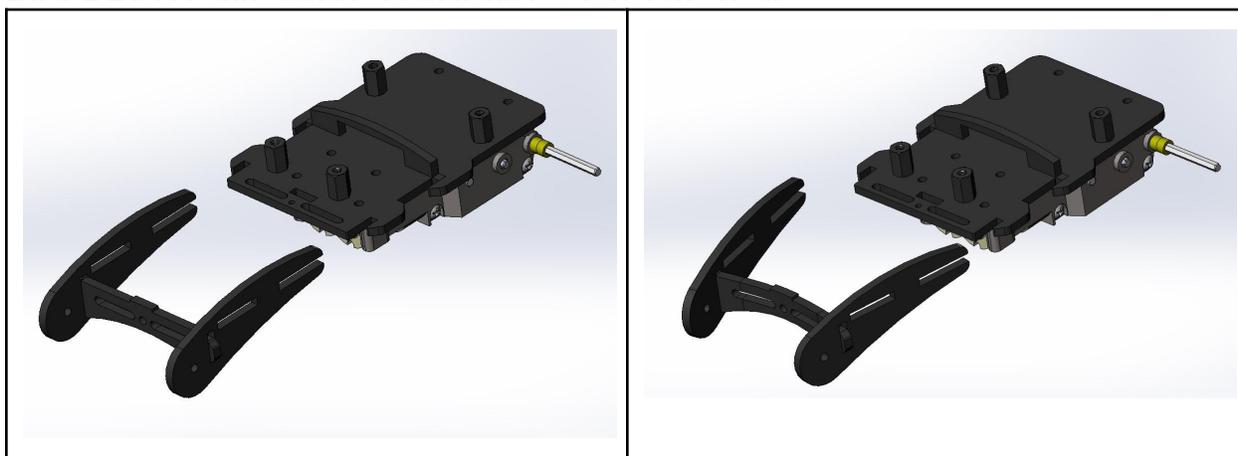
### ÉTAPE 22 : Plaque de support supérieure (facultatif)



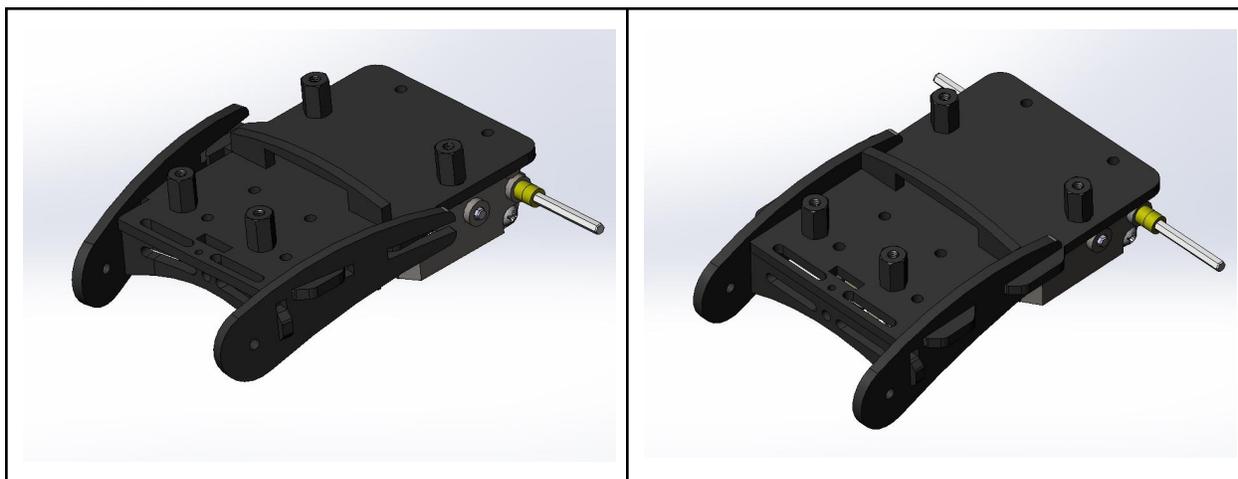
### ÉTAPE 23 : Supports



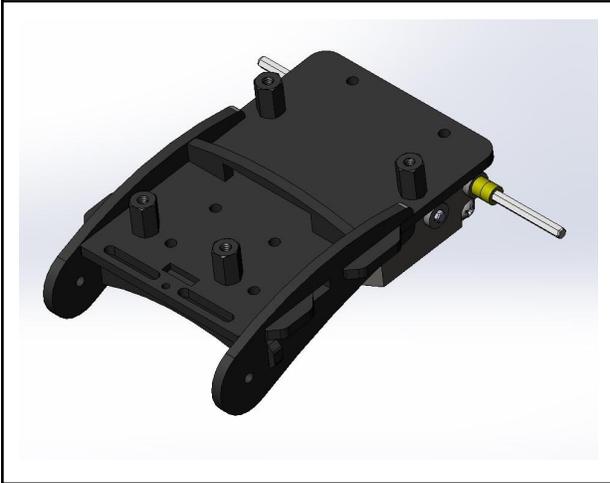
### ÉTAPE 24A : Panneaux latéraux flexibles vers l'extérieur



### ÉTAPE 24B : Faites glisser les panneaux latéraux sur les « verrous » avant, puis redressez-les

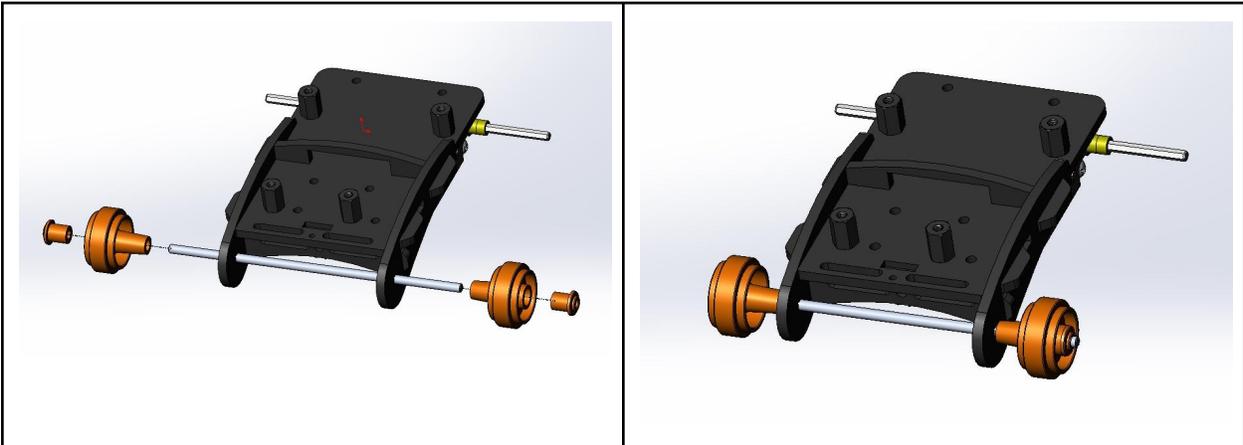


**ÉTAPE 24C : Poussez la languette de la plaque de support inférieure sous la base ; faites-la glisser vers l'arrière pour la verrouiller.**

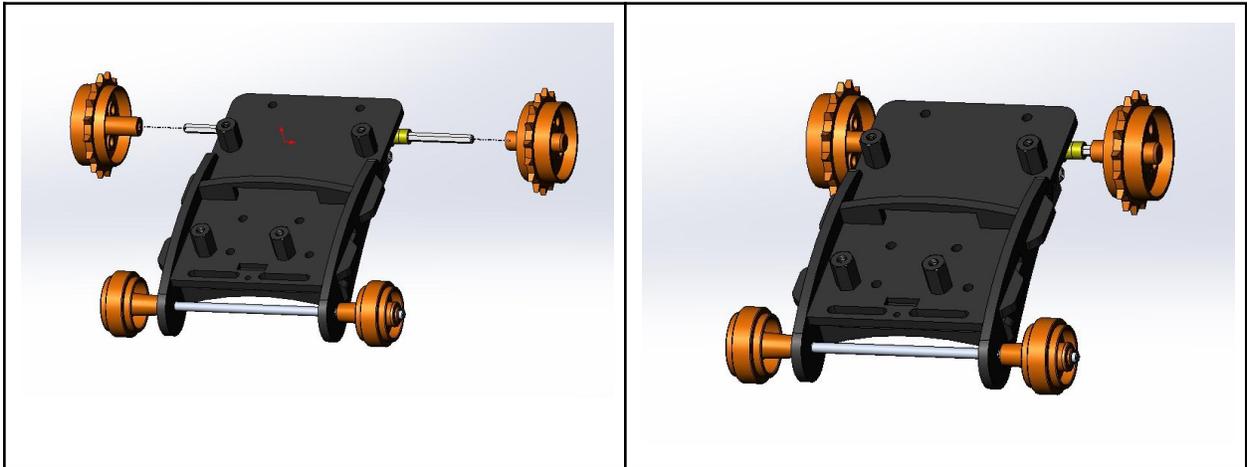


*Remarque : La plaque de support inférieure comporte une languette en haut. Cette languette s'insère dans la fente de la base et maintient toutes les pièces en lexan en place sans utiliser de vis.*

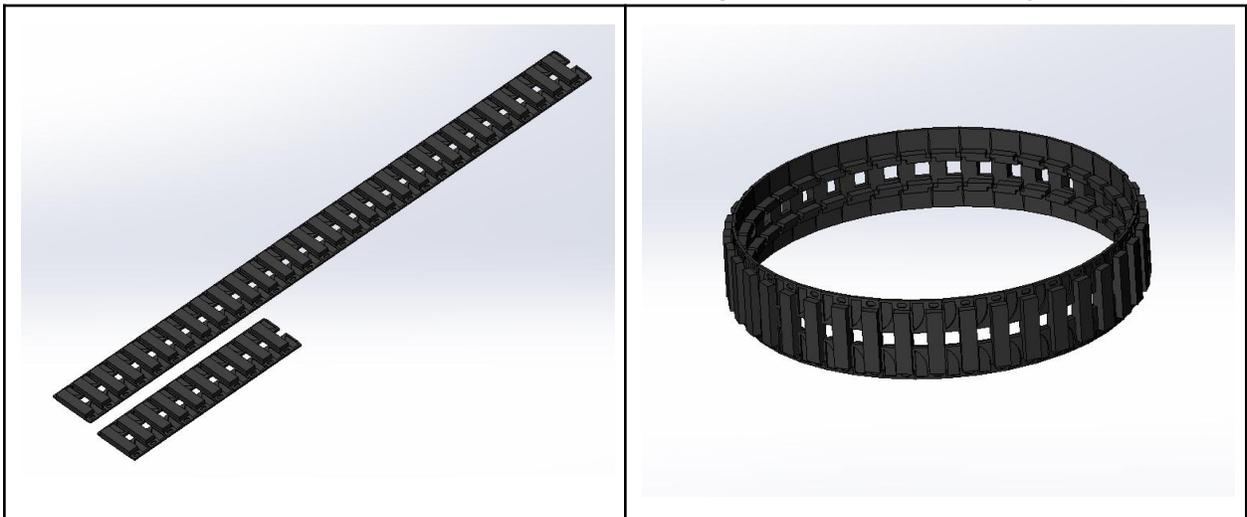
**ÉTAPE 25 : Pignons de renvoi**



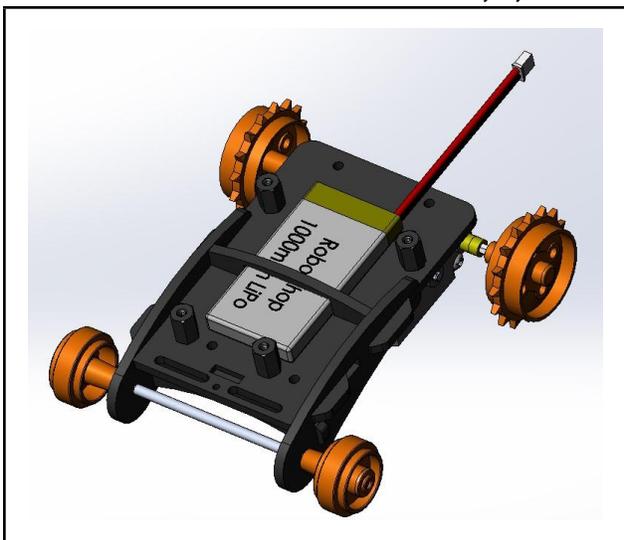
### ÉTAPE 26 : Pignons d'entraînement



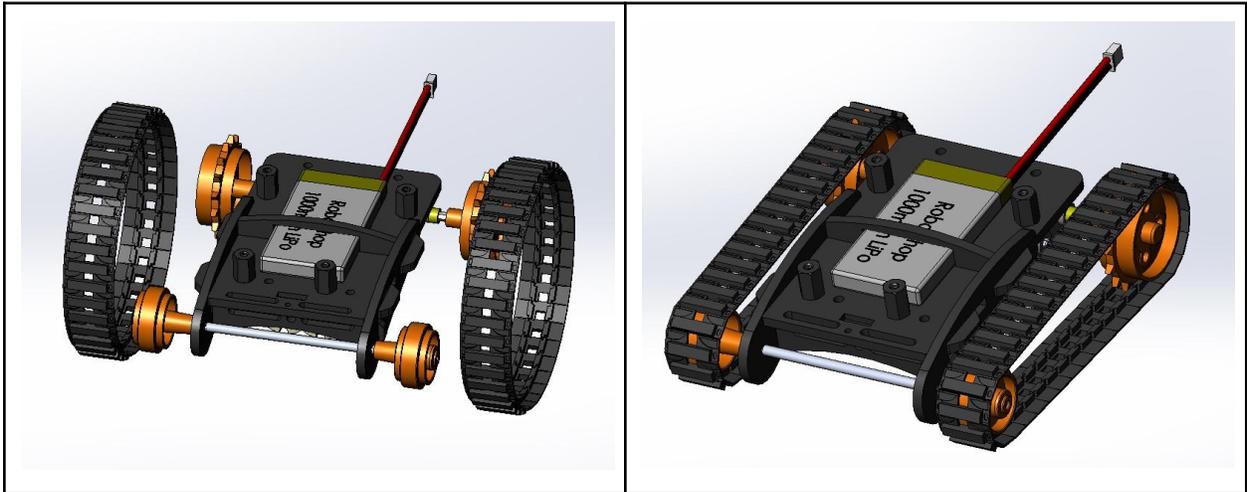
### ÉTAPE 27 (2x) : Connecter seulement 1 chenille longue + 1 chenille de moyenne par côté



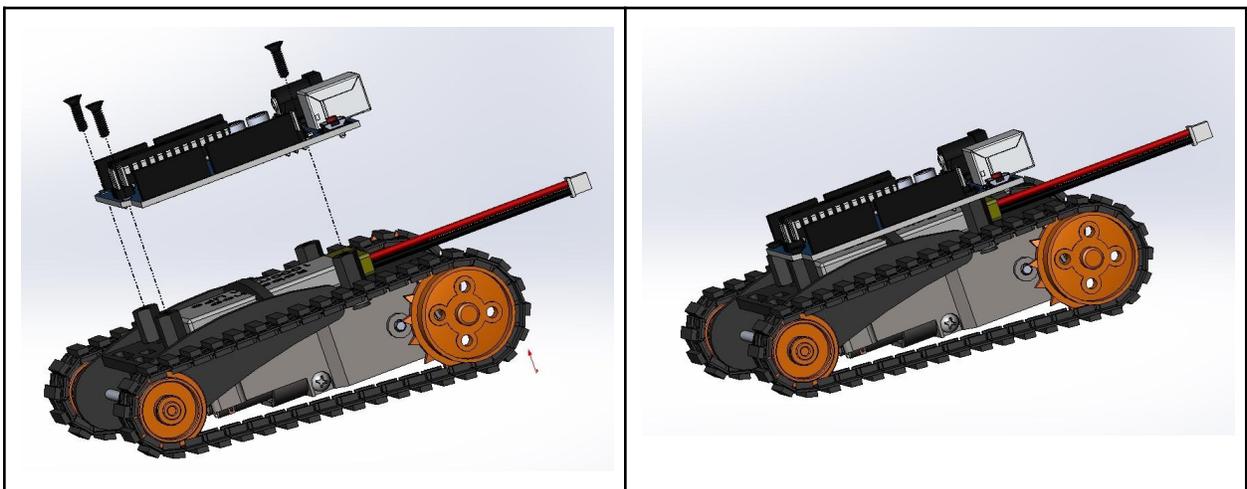
### ÉTAPE 28 : Batterie LiPo 1000mAh, 3,7V



## ÉTAPE 29 : Chenilles



## ÉTAPE 30 : Microcontrôleur

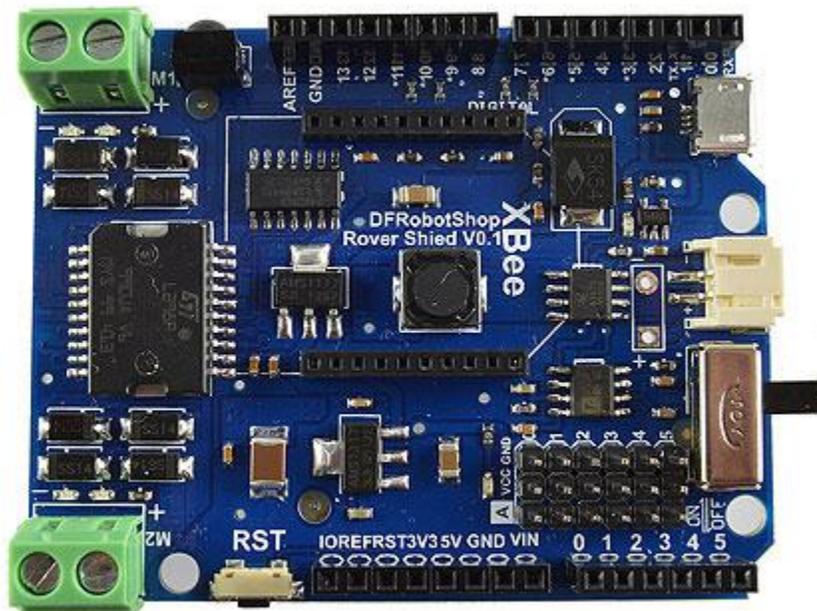


*Remarque: Seulement trois des quatre entretoises sont utilisées - la quatrième sert simplement de support.*

# Blindage Rover DFRobotShop

Le blindage pour DFRobotShop Rover est le le blindage "tout en un" idéal pour le développement de petits robots mobiles basés sur Arduino. Ce blindage vous permet d'alimenter votre Arduino à l'aide d'une batterie LiPo 3.7V et possède même un chargeur LiPo 3.7V intégré, vous permettant de charger la batterie via USB. Le pont en H L298P intégré (contrôleur de moteur CC à brosse double) est parfait pour contrôler deux petits moteurs à engrenages basse tension tels que ceux proposés par Tamiya, et permet un contrôle total de la vitesse et de la direction. Les caractéristiques supplémentaires comprennent un récepteur IR universel, des connecteurs XBee et des broches analogiques (pour une connexion facile aux capteurs). Le blindage est conçu pour être utilisé avec un microcontrôleur compatible avec le blindage, tel que l'Arduino UNO inclus dans ce kit.

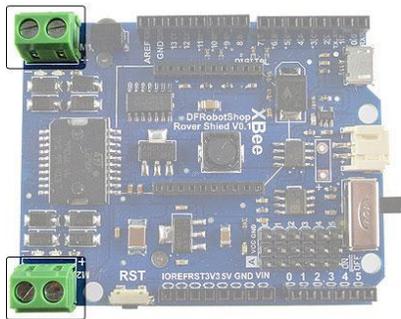
Nous ne suggérons pas d'utiliser le blindage avec une plaque d'essais sans soudure ou avec d'autres microcontrôleurs compatibles avec le blindage qui peuvent ne pas avoir le même brochage ou la même tension qu'un Arduino UNO standard. Lorsque vous empilez plusieurs blindages, ou lorsque vous utilisez le blindage avec une carte basée sur Arduino qui a des fonctionnalités supplémentaires intégrées, assurez-vous de vérifier les broches qui sont utilisées, car certaines peuvent interférer les unes avec les autres.



## Principales caractéristiques

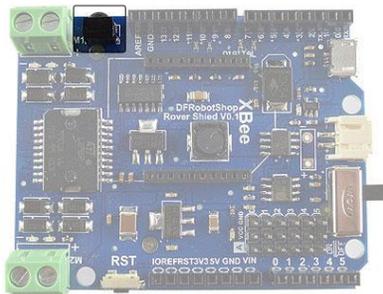
- Contrôleur de moteur CC double (3,5 à 9 V et jusqu'à 2 A par canal)
- Régulateur de tension intégré
- Chargeur de batterie LiPo 3,7 V (via connecteur micro USB)
- Broches compatible XBee
- Récepteur IR universel 38Khz (portée non obstruée ~10m)
- Interrupteur marche / arrêt
- Broches analogiques

### Contrôleur de moteur à courant continu double / pont en H



Ces deux paires de bornes à vis permettent de connecter deux petits moteurs à courant continu (CC). M1 et M2 sont sérigraphiés sur le blindage. La puce utilisée, la L298P, est destinée à être utilisée avec des moteurs CC à balais et peut fournir au maximum 2A par canal. Lorsque vous utilisez des moteurs CC à balais, assurez-vous de leur laisser le temps de refroidir car le cycle de service (temps de fonctionnement vs temps de refroidissement) est d'environ 25 %.

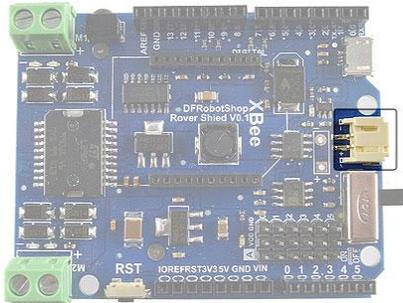
### Récepteur infrarouge



Voici le récepteur infrarouge. L'exemple de code ci-dessous fonctionne avec de nombreuses télécommandes infrarouges (mais pas toutes). Nous suggérons quelques télécommandes IR (telles qu'une télécommande TV universelle) pour voir si vous pouvez lire le signal avec le code fourni ci-dessous. Notez que le récepteur IR subira des interférences provenant des lumières

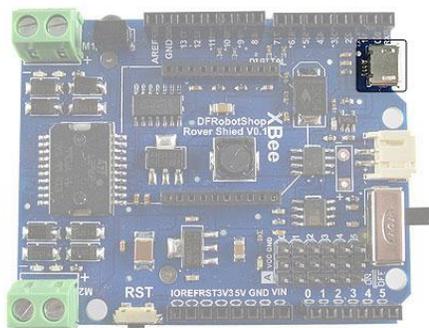
normales, du soleil et d'autres sources de lumière infrarouge. Ces sources créent elles-mêmes des signaux et peuvent interférer avec les signaux des appareils IR. L'infrarouge fonctionne mieux lorsqu'une télécommande IR est pointée directement sur la surface arrondie (réceptrice) du récepteur IR, bien qu'elle détecte les signaux IR réfléchis par d'autres surfaces. Lorsque vous l'utilisez pour contrôler un robot mobile, nous vous suggérons fortement de le faire dans une pièce avec le moins d'autres sources IR possible.

### Connecteur JST LiPo



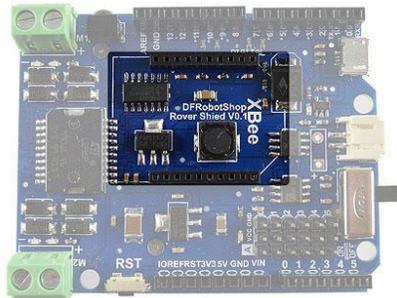
Le connecteur mini JST est destiné à être utilisé avec une batterie LiPo 3,7V, mais vous pouvez l'utiliser avec d'autres sources d'alimentation telles que NiMh, Alcaline ou autre. Le régulateur de tension intégré fournira cette tension directement aux moteurs (non régulé) et régulera (augmentation ou diminution) la tension à utiliser par le microcontrôleur Arduino et d'autres appareils électroniques. Avant de connecter une batterie ou une source d'alimentation, assurez-vous que le fil rouge (positif) va à la borne « + » et que le fil noir (négatif) va à la borne « - » (comme sérigraphié sur la carte). Certaines batteries LiPo ont les fils dans l'autre sens et la connecter sans modification endommagera les cartes. Si vous utilisez des batteries autres qu'un LiPo 3,7 V, NE branchez PAS le câble micro USB car il tentera de charger les autres batteries comme s'il s'agissait de LiPo 3,7 V.

### Port de chargement USB



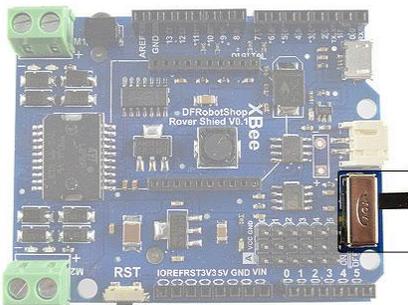
Le port USB sur le blindage est spécifiquement destiné à charger une batterie LiPo de 3,7 V et n'est pas utilisé pour les données/communications ni comme source d'alimentation secondaire. Le connecteur utilisé est le micro USB, que l'on trouve souvent sur les téléphones intelligents. Pour charger un LiPo 3,7 V, connectez simplement la batterie au connecteur JST et le port USB à un appareil pouvant fournir de l'énergie (idéalement un ordinateur ou un adaptateur mural fournissant 5 V à 500 mA ou plus). Si vous utilisez des batteries autres qu'un LiPo 3,7 V, NE branchez PAS le câble micro USB car il tentera de charger les autres batteries comme s'il s'agissait de LiPo 3,7 V.

## Broches XBee



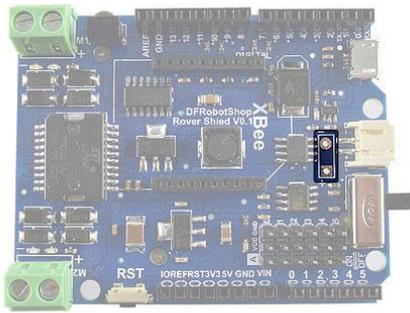
Les broches XBee inclus sur la carte sont idéalement utilisées pour la communication sans fil. Ils sont compatibles avec les modules XBee des séries 1 et 2, ainsi qu'avec de nombreux modules Bluetooth et WiFi qui ont une empreinte XBee. Veillez à installer le module dans le même sens que la sérigraphie sur la carte. Lors du téléchargement du code sur le microcontrôleur, le module XBee ou le blindage doivent être retirés.

## Interrupteur Marche / Arrêt



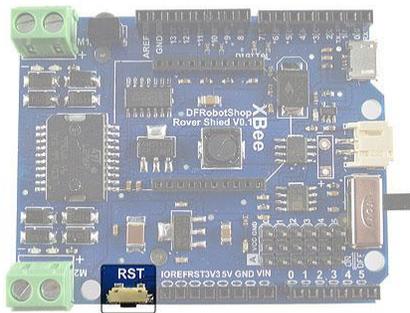
L'interrupteur marche / arrêt connecte essentiellement la batterie au reste de la carte. Plutôt que de déconnecter la batterie pour alimenter le robot, il vous suffit maintenant d'actionner un interrupteur.

## Broches d'alimentation



En plus du connecteur JST blanc, il existe deux connexions (non populé) qui vous permet de souder votre propre connecteur ou câble.

### Réinitialisation



Le bouton de réinitialisation offre la même fonctionnalité que le bouton de réinitialisation trouvé sur le microcontrôleur Arduino, mais vous évite d'avoir à retirer le blindage pour accéder à celui-ci. Éteindre puis rallumer la carte a le même effet.

## Brochage

Les broches suivantes sont connectées en permanence. Si vous souhaitez empiler d'autres cartes, assurez-vous de vérifier que les mêmes broches ne sont pas utilisées pour chaque blindage. Si vous choisissez de ne pas utiliser de fonctionnalités spécifiques, telles que le contrôleur de moteur intégré, vous pouvez toujours envoyer des signaux via les broches d'E/S aux autres blindages.

D0 : connectée à la broche XBee Tx (communication série)

D1 : Connectée à la broche XBee Rx (communication série)

D5 : contrôle de vitesse E2

D6 : contrôle de vitesse E1

D7 : contrôle de direction M2

D8 : contrôle de direction M1

D9 : broche du récepteur IR (le capteur est un récepteur uniquement, il ne peut pas émettre).

*Remarque: Lors du téléchargement du code sur la carte Arduino, il est important que le Xbee ou un autre module sans fil associé ne soit pas connecté, libérant ainsi les lignes de communication série (broches numériques 0 et 1 sur l'Arduino).*

# Arduino

*“Arduino est une plate-forme de prototypage électronique open source basée sur du matériel et des logiciels flexibles et faciles à utiliser. Il est destiné aux artistes, designers, amateurs et à toute personne intéressée par la création d'objets ou d'environnements interactifs.”*

[www.arduino.cc](http://www.arduino.cc)

Il existe de nombreuses façons d'apprendre à coder dans Arduino. Nous suggérons un ou plusieurs des éléments suivants :

[www.arduino.cc](http://www.arduino.cc) -> Catégories « Apprentissage » et « Référence »

Livres sur Arduino <http://www.robotshop.com/books.html>

## Installation logiciel & Configuration

- Installez le logiciel Arduino IDE sur [Arduino.cc](http://www.arduino.cc)
- Téléchargez la bibliothèque infrarouge :  
<http://www.robotshop.com/content/ZIP/arduino-library-dfr0107.zip>
- Décompressez / extrayez le dossier IRemote vers Arduino -> Bibliothèques
- Ouvrez le logiciel Arduino et importez la bibliothèque IR comme suit :
- Sketch -> Import Library -> Add Library (recherchez le dossier IRemote)

Vous n'êtes pas limité à l'utilisation de la bibliothèque IR, bien qu'il s'agisse du seul code “officiel” pris en charge à l'heure actuelle par DFRobot et RobotShop.

## Exemple de Code

Lorsqu'il est connecté à un microcontrôleur Arduino, le blindage DFRobotShop Rover ajoute le matériel de base nécessaire pour faire fonctionner un petit robot mobile. Nous proposons ci-dessous des exemples de code de base pour vous aider à démarrer dans la programmation de robots mobiles et d'Arduino. Le code ci-dessous n'est pas "optimisé" mais est destiné à vous aider à démarrer avec chacune des principales fonctionnalités du blindage.

## Avancer à pleine vitesse

Le code ci-dessous fera tourner les deux moteurs connectés au pont en H intégré à pleine vitesse.

```
/* Copy and paste the code below into the Arduino software */
int E1 = 6; //M1 Speed Control
int E2 = 5; //M2 Speed Control
int M1 = 8; //M1 Direction Control
int M2 = 7; //M2 Direction Control

void setup(){
  int i;
  for(i=5;i<=8;i++)
    pinMode(i, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  int leftspeed = 255; //255 is maximum speed
  int rightspeed = 255;
  analogWrite (E1,255);
  digitalWrite(M1,LOW);
  analogWrite (E2,255);
  digitalWrite(M2,LOW);
  delay(100);
}
```

## Contrôle moteur W/A/S/D

Le code ci-dessous permet d'envoyer des commandes série au DFRobotShop Rover par une connexion filaire (USB) ou sans fil (Bluetooth, XBee). Si vous envisagez de créer votre propre

code, vous pouvez l'utiliser pour acquérir une compréhension de base de la façon de faire fonctionner les moteurs.

Les commandes clavier sont :

- "w" : avancer
- "a" : tourner dans le sens des aiguilles d'une montre (c'est-à-dire vers la gauche)
- "s" : tourner dans le sens antihoraire (c'est-à-dire à droite)
- "d" : reculer

Le rover exécutera la commande qui lui est envoyée jusqu'à ce qu'on lui dise de s'arrêter en appuyant sur n'importe quelle autre touche du clavier. Les commandes peuvent être envoyées via le terminal série Arduino, bien que vous deviez appuyer sur la touche "Entrée" pour que la commande soit envoyée. Des interfaces telles que Hyperterminal pour Windows ou Putty peuvent également être utilisées.

Assurez-vous de sélectionner le bon port COM et le bon débit de 9600 bauds. Le débit en bauds du module Bluetooth peut être de 9600 ou 115 200 selon le module - consultez le manuel pour en être certain. Notez qu'il est probable que les deux moteurs ne soient pas identiques et que vous devrez ajuster légèrement la vitesse pour que le robot aille à peu près droit. Notez que les encodeurs aident le robot à se déplacer en ligne droite.

```
/*To control the rover, copy and paste the code below into the Arduino
software*/
int E1 = 6; //M1 Speed Control
int E2 = 5; //M2 Speed Control
int M1 = 8; //M1 Direction Control
int M2 = 7; //M2 Direction Control

void setup(void){
  int i;
  for(i=5;i<=8;i++)
    pinMode(i, OUTPUT);
  Serial.begin(9600); // Therefore if you send serial commands, they should
be at 9600 baud, Xon/Xoff
}

void loop(void){
  while (Serial.available() < 1) {} // Wait until a character is received
  char val = Serial.read();
  int leftspeed = 255; //255 is maximum speed
  int rightspeed = 255;
  switch(val) { // Perform an action depending on the command
    case 'w': //Move Forward
      forward (leftspeed,rightspeed);
      break;
```

```

    case 's': //Move Backwards
        reverse (leftspeed,rightspeed);
        break;
    case 'a': //Turn Left
        left (leftspeed,rightspeed);
        break;
    case 'd': //Turn Right
        right (leftspeed,rightspeed);
        break;
    default:
        stop();
        break;
}
}
void stop(void) {
    digitalWrite(E1,LOW);
    digitalWrite(E2,LOW);
}

void forward(char a,char b){
    analogWrite (E1,a);
    digitalWrite(M1,LOW);
    analogWrite (E2,b);
    digitalWrite(M2,LOW);
}

void reverse (char a,char b){
    analogWrite (E1,a);
    digitalWrite(M1,HIGH);
    analogWrite (E2,b);
    digitalWrite(M2,HIGH);
}

void left (char a,char b){
    analogWrite (E1,a);
    digitalWrite(M1,HIGH);
    analogWrite (E2,b);
    digitalWrite(M2,LOW);
}
void right (char a,char b){
    analogWrite (E1,a);
    digitalWrite(M1,LOW);
    analogWrite (E2,b);
    digitalWrite(M2,HIGH);
}
}

```

## Encodeurs (en option)

Le code ci-dessous compte simplement le nombre de changements tels qu'ils sont visualisés par un encodeur analogique à canal unique. L'encodeur suggéré (Tamiya Encoder Kit) utilise un disque avec 8x sections blanches et 8x découpes. Cela fournira un compte de 16 par rotation de 360 degrés. À vous de l'optimiser et de l'intégrer à votre code. Le code envoie également des commandes aux moteurs pour les faire tourner.

```
int rawsensorValue = 0; // Store the value coming from the sensor
int sensorcount0 = 0;
int sensorcount1 = 0;
long count = 0;

void setup() {
  int i;
  for(i=5;i<=8;i++)
    pinMode(i, OUTPUT);
  Serial.begin(9600);
  int leftspeed = 255; //255 is maximum speed
  int rightspeed = 255;
}

void loop() {
  analogWrite (10,255);
  digitalWrite(12,LOW);
  analogWrite (11,255);
  digitalWrite(13,LOW);
  delay(20);
  rawsensorValue = analogRead(0);
  if (rawsensorValue < 600){ //Min value is 400 and max value is 800, so
state chance can be done at 600.
    sensorcount1 = 1;
  }
  else {
    sensorcount1 = 0;
  }
  if (sensorcount1 != sensorcount0){
    count ++;
  }
  sensorcount0 = sensorcount1;
  Serial.println(count);
}
```

## Détermination des codes IR

Le code ci-dessous vous aidera à déterminer les valeurs des signaux envoyés depuis une télécommande IR. Notez que le code ne fonctionne pas pour toutes les télécommandes IR et peut devoir être personnalisé. Copiez et collez tout le code ci-dessous dans la fenêtre Arduino. Notez que ce code ne fonctionnera que si la bibliothèque IRemote est correctement installée (voir les instructions ci-dessus).

```
#include <IRremote.h>
#include <IRremoteInt.h>

IRrecv irrecv(9); // D9 on the shield is permanently connected to the IR
receiver
decode_results results;

void setup()
{
  Serial.begin(9600); // This determines the baud rate
  irrecv.enableIRIn(); // Enable the receiver
}

void loop(){
  if (irrecv.decode(&results)){
    Serial.println(results.value, DEC); // alternatively use HEX rather than
DEC
    irrecv.resume(); // Receive the next value
  }
}
```

Une fois que vous avez vérifié et téléchargé le code, ouvrez le terminal série dans l'IDE Arduino, en gardant l'Arduino connecté à l'ordinateur. Couvrez le récepteur IR avec votre main de tous les côtés sauf celui avec la partie ronde tournée vers l'extérieur (pour limiter les interférences IR des lumières, du soleil et d'autres sources). Dirigez votre télécommande infrarouge vers le récepteur et appuyez sur un bouton. Vous devriez voir une valeur apparaître dans la fenêtre du terminal série uniquement lorsque vous appuyez sur un bouton. Appuyez plusieurs fois sur le bouton pour vous assurer qu'il s'agit de la bonne valeur et notez-le dans un fichier (le bouton et le code correspondant).

## Exemple de code IR avec contrôle moteur

Le code suivant fait avancer le rover pendant un temps spécifique lorsqu'un code IR spécifique est détecté. Nous vous encourageons à comprendre le code WASD fourni ci-dessus et à utiliser l'exemple de code ci-dessous comme modèle afin de créer un rover contrôlé par infrarouge.

```

#include <IRremote.h>
#include <IRremoteInt.h>

int RECV_PIN = 9;
IRrecv irrecv(RECV_PIN);
decode_results results;

int E1 = 6; //M1 Speed Control
int E2 = 5; //M2 Speed Control
int M1 = 8; //M1 Direction Control
int M2 = 7; //M2 Direction Control

void setup(){
  int i;
  for(i=5;i<=8;i++)
    pinMode(i, OUTPUT);
  Serial.begin(9600);
  irrecv.enableIRIn(); // Start the receiver
}

void loop(){
  int leftspeed = 255; //255 is maximum speed
  int rightspeed = 255;
  if (irrecv.decode(&results)){
    if(results.value == 16593103){ // This code will vary based on your
remote control
      analogWrite (E1,255);
      digitalWrite(M1,HIGH);
      analogWrite (E2,255);
      digitalWrite(M2,HIGH);
      delay(1000);
    }
    digitalWrite(E1,LOW);
    digitalWrite(E2,LOW);
    irrecv.resume(); // Receive the next value
  }
}

```