

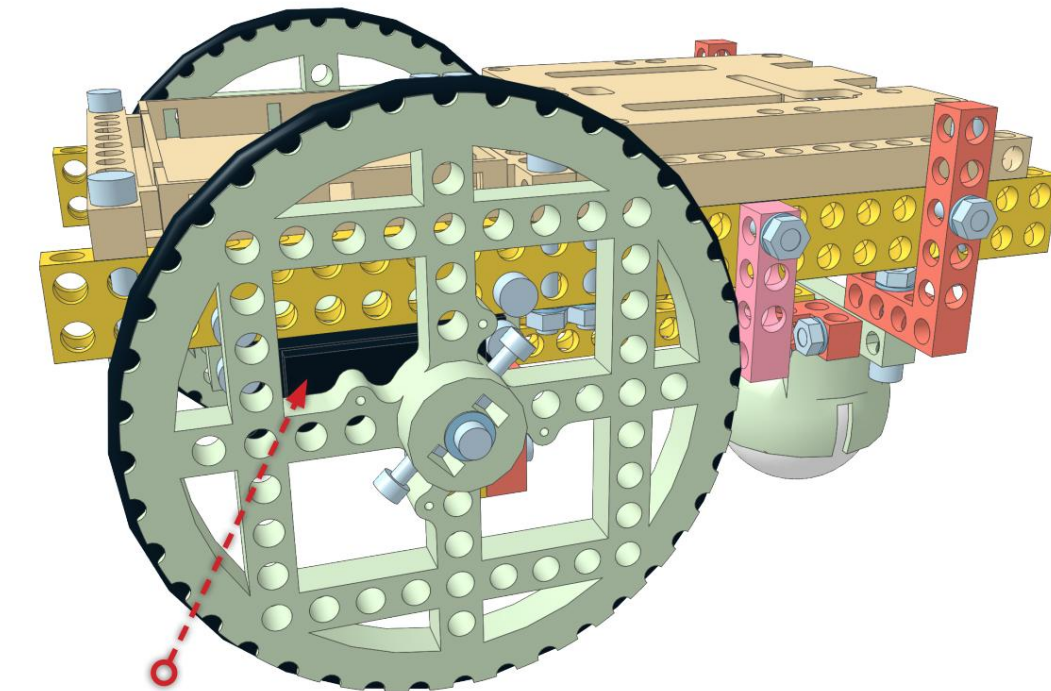
construction d'un robot mobile à partir du kit de construction Open source  
imprimé en 3D BITBEAM.  
Un Arduino UNO R3 est utilisé comme microcontrôleur programmable de  
contrôle.

## le robot FERDA

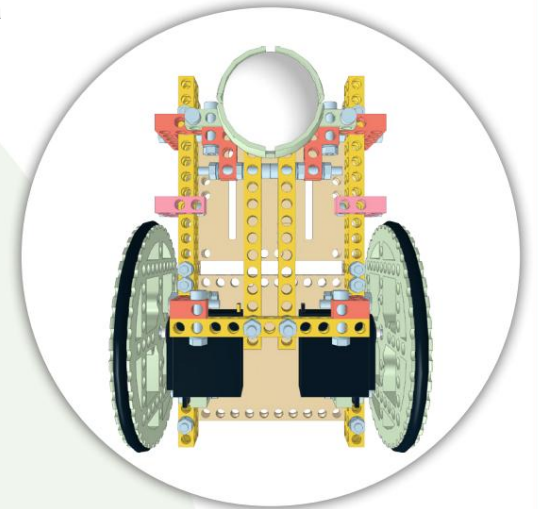
version 1.1 bêta

Champ sans soudure 400

Arduino UNO R3



Servomoteurs continus



## Vous pouvez distribuer librement ce fichier PDF !

Le matériel sur lequel vous avez mis la main a été créé sur la base de l'expérience de l'enseignement des enfants (10-14 ans) dans le cercle éducatif de programmation et de robotique JAOS ([www.policka.evangelnet.cz/roboti](http://www.policka.evangelnet.cz/roboti)).

J'aimerais que ce matériel atteigne tous ceux pour qui il peut être d'une certaine utilité. Il ne doit pas s'agir uniquement d'enseignants et de dirigeants de cercles de robotique, cela peut être n'importe qui qui construit avec enthousiasme un m-Bitbeam à partir d'un kit de construction et programme un Arduino...

Vous pouvez distribuer le fichier PDF sur Internet ou sur n'importe quel support de données. Vous pouvez également le rendre disponible en téléchargement à partir de votre serveur ou de votre site Web. La seule condition est que la distribution du fichier PDF soit toujours complète et entièrement gratuite. Dans le même temps, le téléchargement du fichier ne doit pas être conditionné à la connexion/inscription de l'utilisateur.

## L'utilisation de ce matériel n'est pas gratuite, une redevance doit être payée...

La conception et la préparation du matériel ont nécessité de nombreuses heures de travail. Si vous comptez utiliser le matériel de manière pratique, veuillez payer une redevance de 39 CZK pour l'utilisation de ce matériel numérique. Votre contribution financière sera utilisée pour la préparation d'autres matériels pédagogiques, qui seront ensuite à nouveau disponibles pour toutes les parties intéressées.

Si vous êtes encore un enfant et que vous ne pouvez pas payer le montant indiqué en ligne, demandez à vos parents d'effectuer le paiement. Si pour une raison quelconque vous ne pouvez pas organiser le paiement, ne vous inquiétez pas. N'hésitez pas à utiliser les instructions. Je crois que lorsque vous serez plus âgé, vous me paierez le montant en plus.

Si vous souhaitez utiliser le matériel comme matériel pédagogique pour vos élèves/étudiants à l'école ou dans un club, veuillez payer le montant indiqué pour chaque élève/étudiant.

Plus d'informations sur m-Bitbeam, y compris les informations de paiement, peuvent être trouvées sur :

<http://www.tfsoft.cz/m-bitbeam>

A la même adresse, vous trouverez également un aperçu de tous les matériaux préparés.

- Si vous placez le téléchargement sur votre site Web/serveur, veuillez me le faire savoir.  
Je vous informerai dès qu'une nouvelle version du guide sera disponible.
- Le manuel PDF est prêt à être imprimé sur du papier A4. Lors de l'impression du PDF sur du papier d'un format différent, n'oubliez pas de définir le réglage de taille correct (par exemple, dans la boîte de dialogue d'impression d'Adobe Reader : Options de taille -> Ajuster).
- Veuillez ne pas altérer le fichier PDF. Si vous rencontrez une erreur, écrivez-moi à ce sujet et je la corrigerai.

Que c'est juste "jouer avec un jeu de construction" ?

## Avec conception programmable ce n'est jamais un kit juste pour "jouer"...

C'est doublement vrai pour le kit open source Bitbeam !

C'est pourquoi nous avons commencé à utiliser Bitbeams avec les enfants du cercle éducatif JAOS...

Travailler avec des blocs de construction développe généralement la pensée logique, soutient une approche systématique et des compétences techniques. Lors de la résolution de problèmes spécifiques et plus complexes, travailler avec ce kit conduit à une coopération d'équipe de manière absolument non violente. De plus, le kit lui-même exerce non seulement la motricité fine, mais aussi la patience, l'imagination spatiale et le travail avec diverses instructions (changer le sens du texte, des images, des graphiques, etc.). Bien sûr, il y a les mathématiques et la physique omniprésentes (mais souvent aussi d'autres sciences naturelles). Enfin, les enfants auront une idée concrète de la programmation.



Tomáš Feltl

[skolniprojekty@gmail.com](mailto:skolniprojekty@gmail.com)

Actuellement, je me concentre principalement sur la question de l'enseignement avec l'utilisation de technologies et d'outils modernes - systèmes de laboratoire (PASCO, Vernier, ...), tableaux blancs interactifs, systèmes de réponse, kits LEGO Mindstorms, kits Bitbeams, Arduino, ... Comme conférencier axé sur l'enseignement interactif, je collabore avec divers organismes et écoles.

## Le robot Ferda se présente...

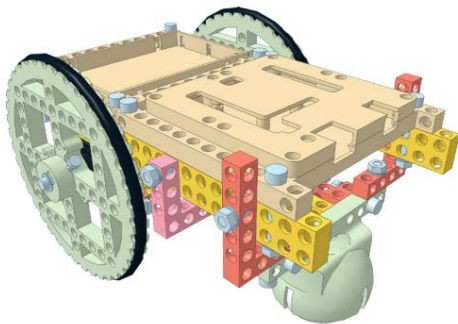
Ce robot a été créé sur la base de l'expérience de l'enseignement des enfants (10-14 ans) dans le cercle éducatif de programmation et de robotique JAOS (www.policka.evangnet.cz/roboti).

Lors de la résolution de différentes tâches et problèmes, il est parfois pratique d'inventer votre robot "sur mesure" pour une certaine tâche. La solution est souvent la "solution de conception" du robot. Mais parfois, la situation est telle que nous devons nous concentrer uniquement sur la solution logicielle au problème. Et c'est là que le robot universel FERDA entre en scène...

Avec FERDA vous pouvez :

- rouler (principalement sur une surface plane),
- dessiner sur le passe-partout (possibilité de coller un feutre dans l'axe de rotation du robot),
- combiner différents capteurs selon les besoins (système modulaire, connexion module rapide),
- déconnecter facilement tout capteur du robot,
- incliner progressivement le module avec le capteur de lumière (une position extrême vers le bas, l'autre vers l'avant),
- remplacer les piles sans démonter le robot,
- ajouter quelque chose d'autre à l'avant (peut-être son propre module d'extension d'origine),
- rouler et conduire à l'aide de la télécommande IR,
- rouler et conduire avec un smartphone.

Le robot universel FERDA peut aussi bien servir comme « explorateur » autonome d'une pièce.



## Qu'est-ce que Bitbeam ?

Bitbeam est un kit de conception et de prototypage disponible entièrement gratuitement (Open Source). Il est dimensionnellement compatible avec LEGO Technics/Mindstorms, de sorte que les deux blocs de construction peuvent être combinés les uns avec les autres. Les pièces Bitbeam de base sont conçues pour pouvoir être facilement produites "à la maison" à partir de différents matériaux en utilisant différentes technologies (impression 3D, fraisage CNC, découpe laser).

Surtout, en combinaison avec l'impression 3D en expansion, c'est une alternative bon marché intéressante aux divers kits de construction commerciaux. Plus d'informations peuvent être trouvées sur

[www.bitbeam.org](http://www.bitbeam.org).

## Pourquoi m-Bitbeam ?

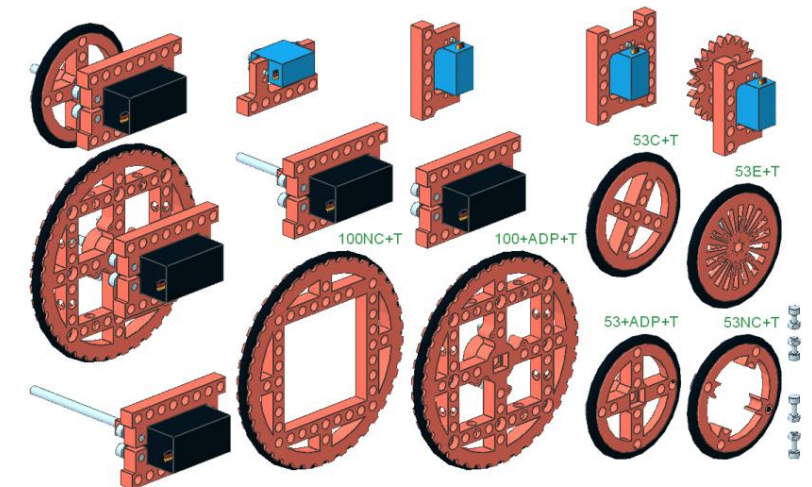
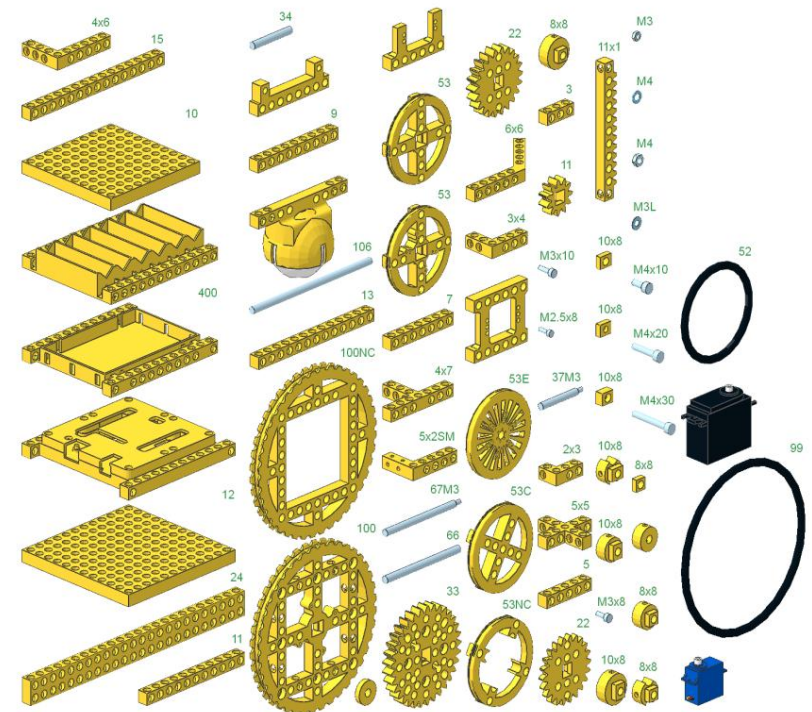
m-Bitbeam est un kit de construction imprimé en 3D qui est basé sur le concept Bitbeam et ajoute un certain nombre de pièces spéciales, telles que diverses roues, supports, boîtes, etc. La partie électronique programmable du kit est basée sur l'Arduino couramment disponible et son grand "écosystème" de divers capteurs et modules.

Il y a trois raisons principales pour la création de m-Bitbeam.

La première raison est le prix élevé des kits programmables universels tels que LEGO Mindstorms. Grâce à son faible prix d'achat, m-Bitbeam est accessible à tous (chaque enfant a son propre kit à la maison). La deuxième raison est la variabilité beaucoup plus grande d'Arduino du point de vue des capteurs, des moteurs, des modules d'extension, etc. Ainsi, les enfants sont exposés à des possibilités "insoupçonnées" et leurs projets peuvent être non seulement beaucoup plus variés, mais aussi avec un chevauchement réel dans la vie réelle et la pratique.

Il suffit de regarder sur Internet pour voir ce qu'Arduino "alimente" aujourd'hui (imprimantes 3D, machines CNC, systèmes de sécurité, systèmes de contrôle, systèmes de mesure, satellites, ...). La dernière raison de la création de m-Bitbeam est de faire connaître une autre technologie - l'impression 3D. Après avoir appris les bases de la modélisation 3D, les enfants peuvent

concevoir et fabriquer facilement vos propres pièces modulaires exactement selon les besoins de votre projet.

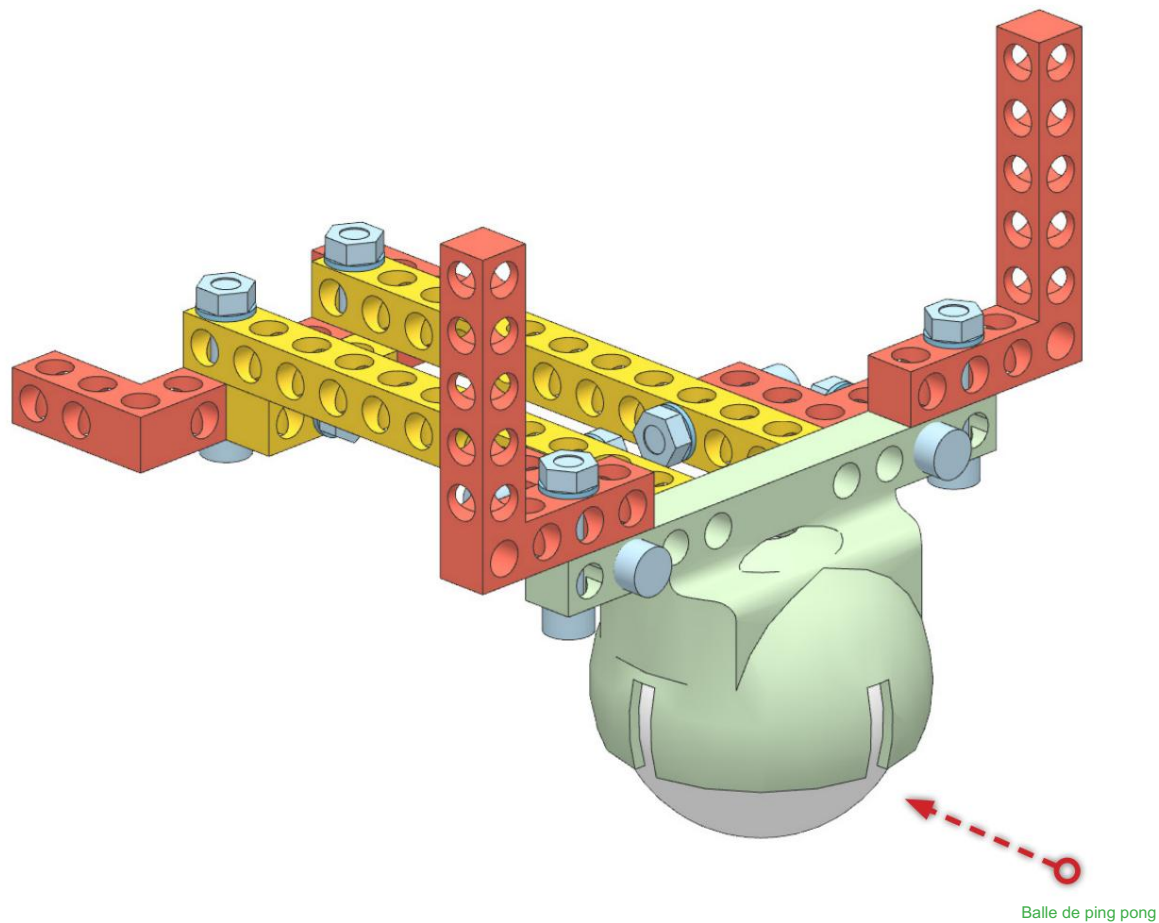


Référentiel de modèles 3D au format STL (pour l'impression 3D de pièces) : [https://github.com/e-Mole/m-Bitbeam\\_Parts\\_for\\_3Dprint](https://github.com/e-Mole/m-Bitbeam_Parts_for_3Dprint)

Référentiel de modèles 3D au format DAT (pour préparer des tutoriels, par exemple MLCad + LDraw) : [https://github.com/e-Mole/m-Bitbeam\\_Parts\\_for\\_LDraw](https://github.com/e-Mole/m-Bitbeam_Parts_for_LDraw)

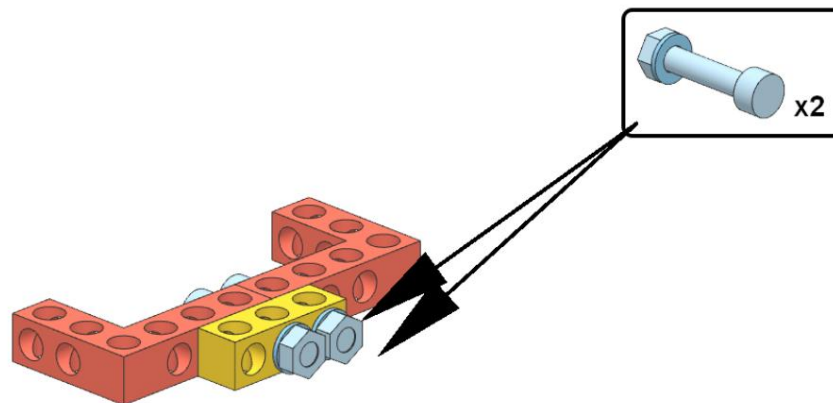
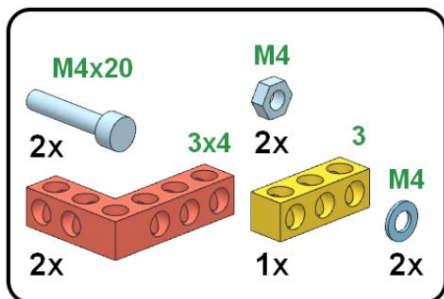


Commencez par la partie inférieure, assemblée, elle ressemblera à ceci :

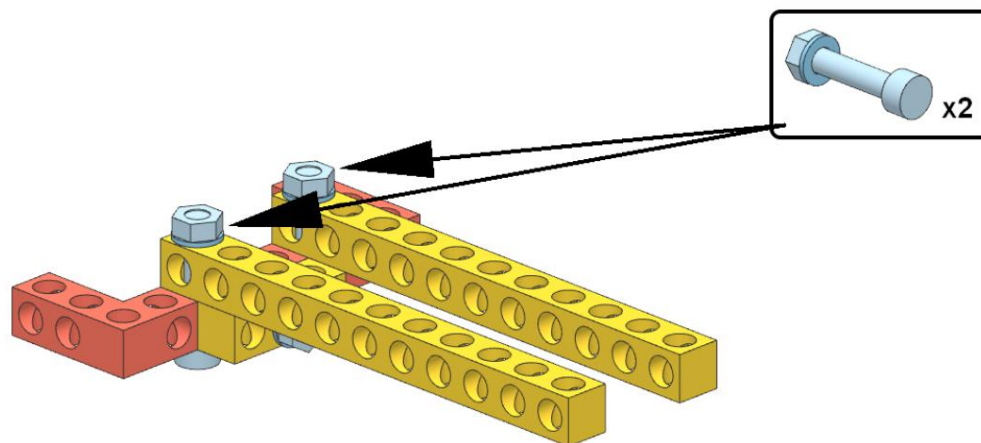
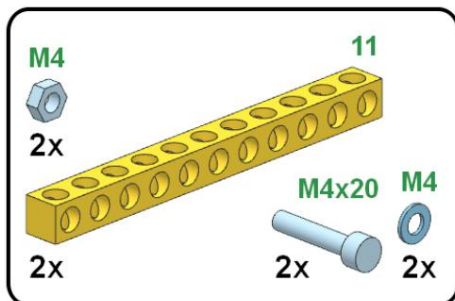


Balle de ping pong

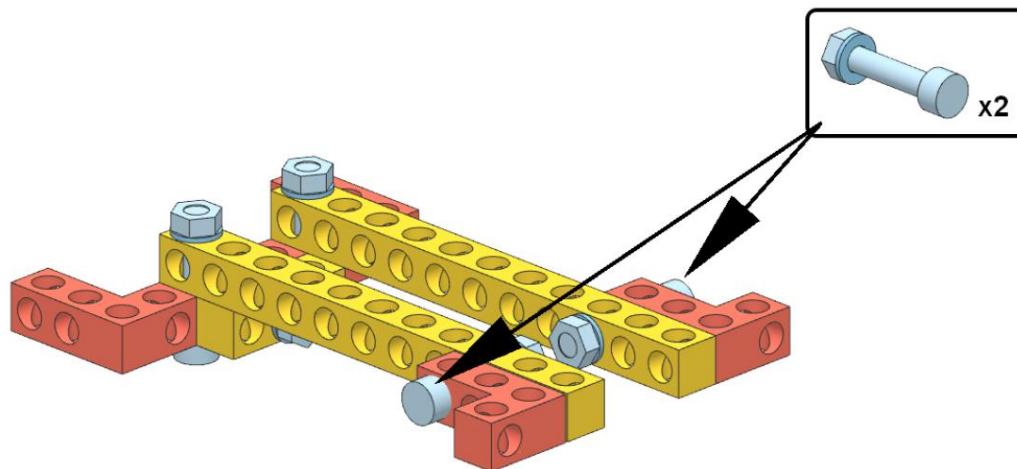
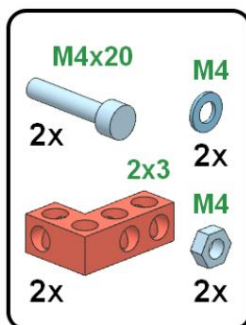
1



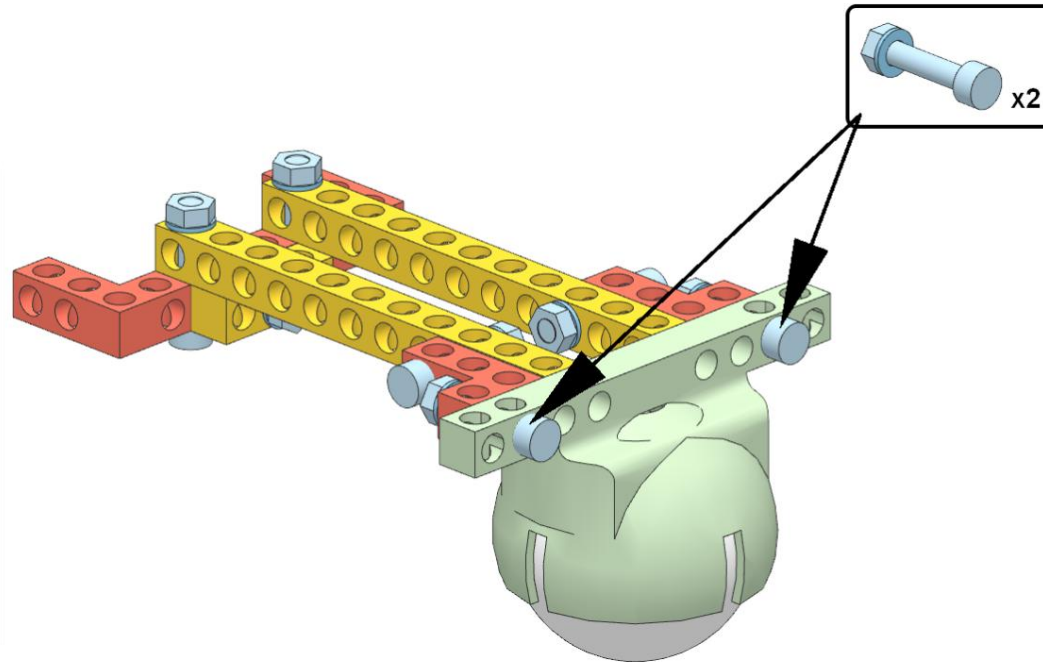
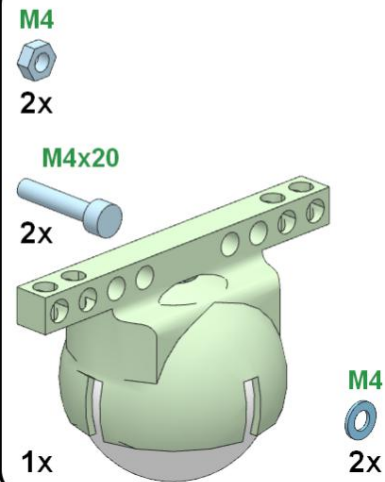
2



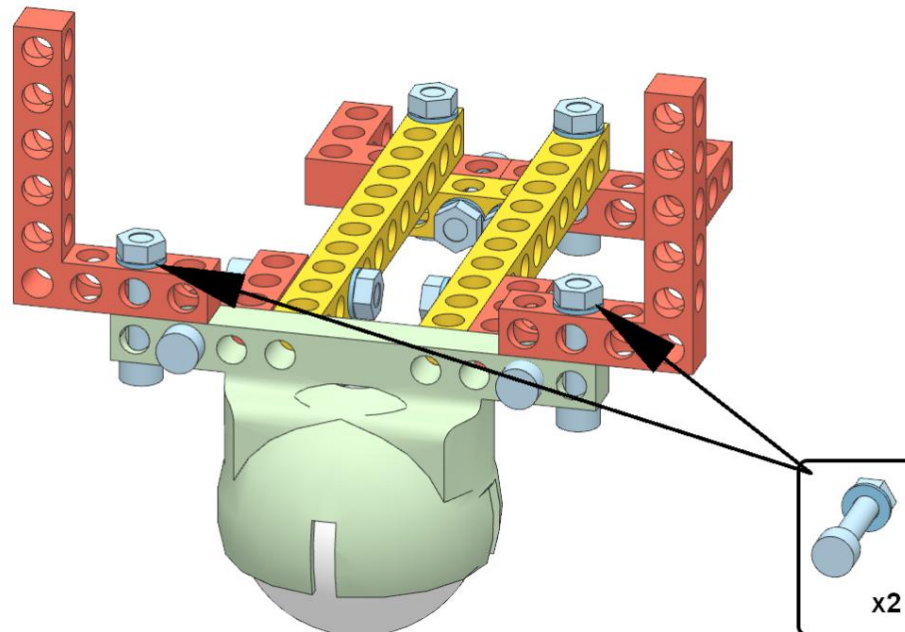
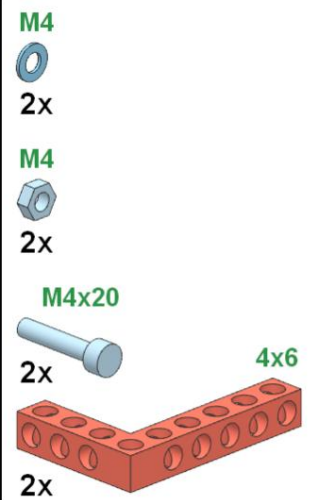
3



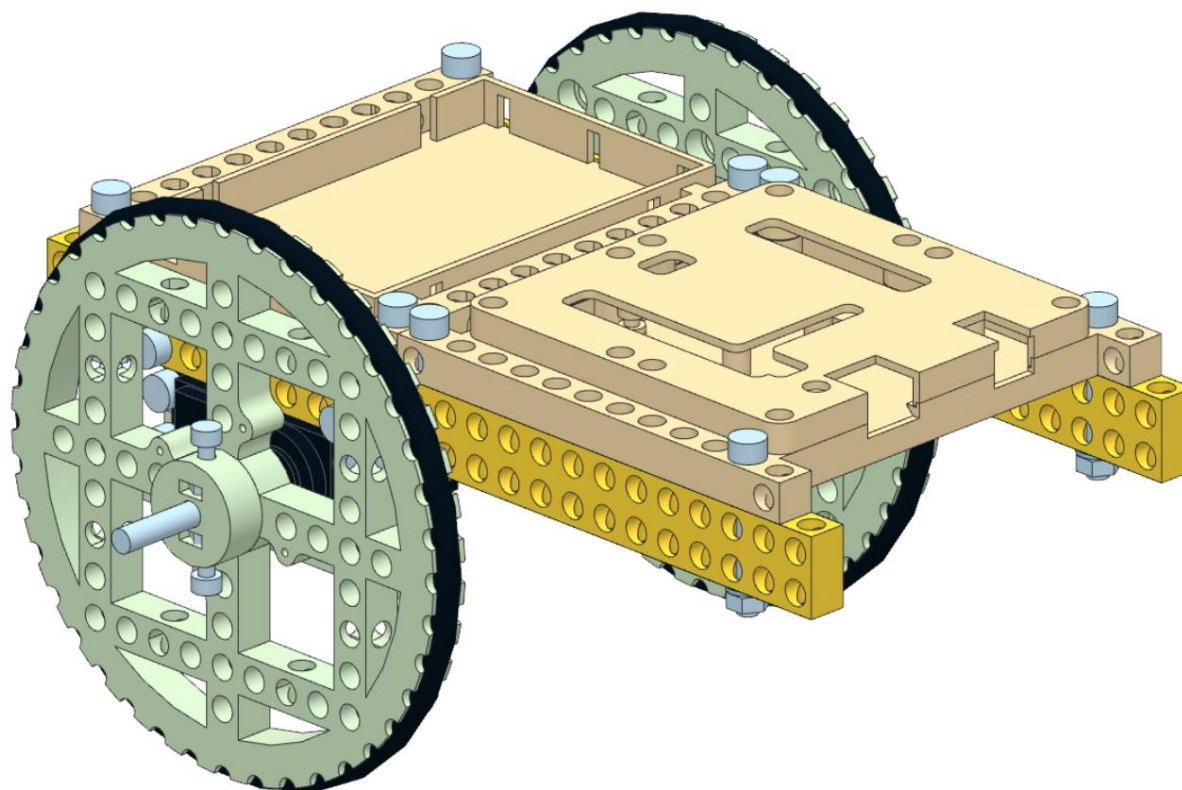
4



5

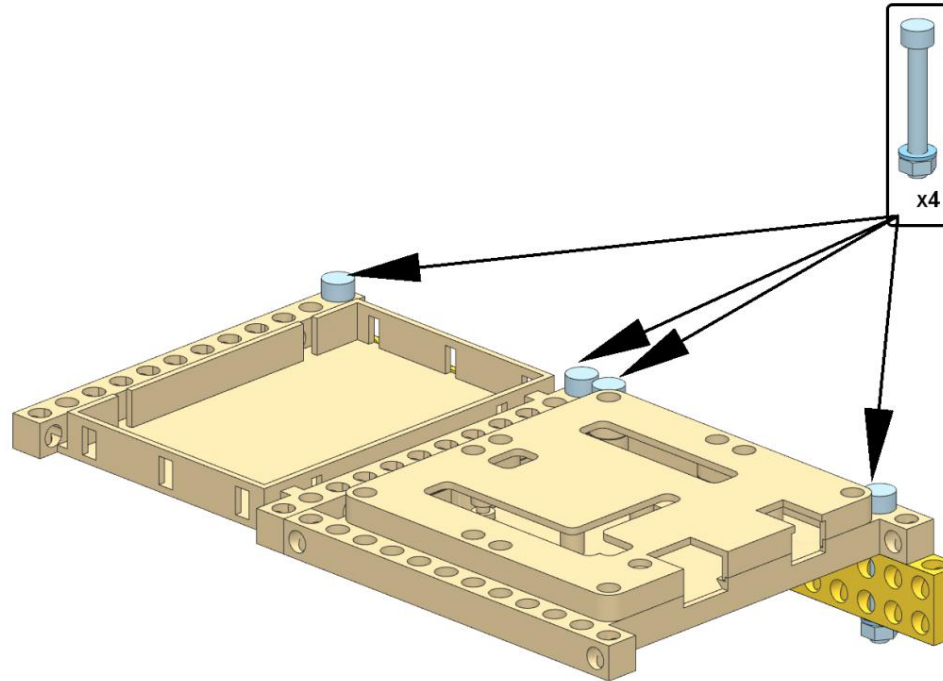
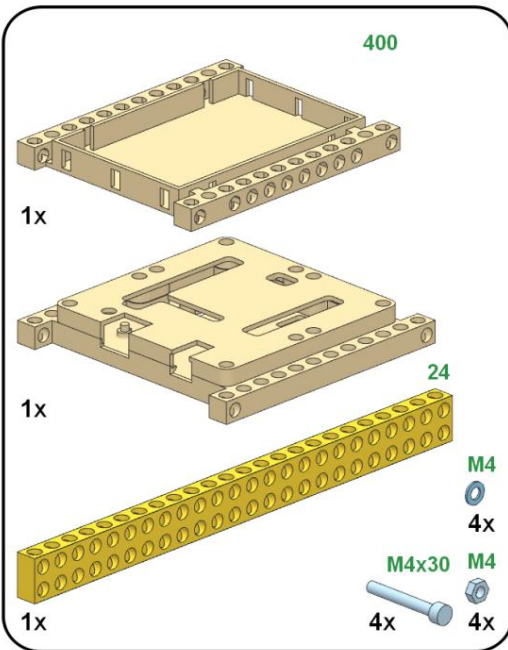


Continuez avec la partie supérieure, assemblée elle ressemblera à ceci :

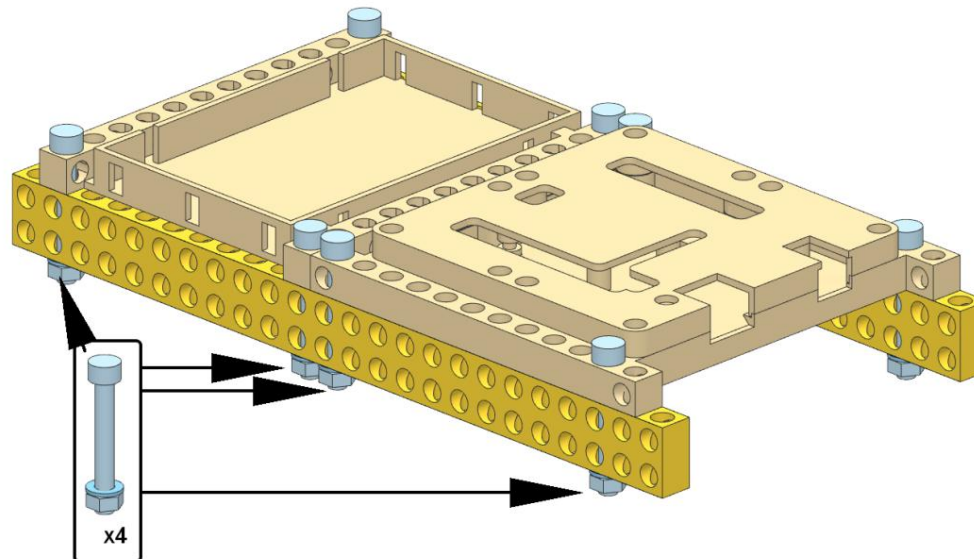
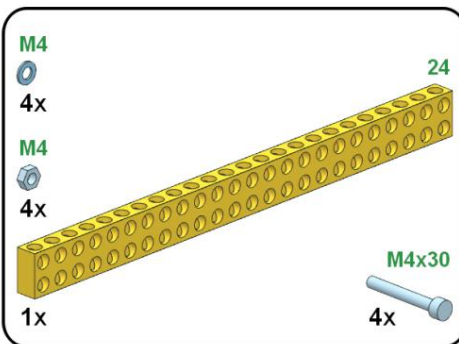




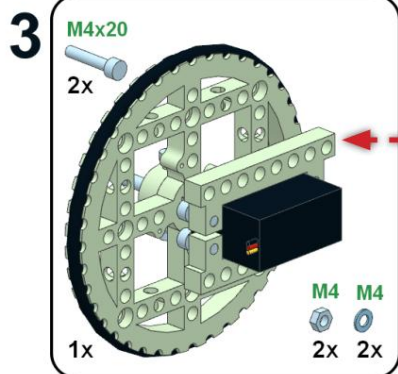
1



2

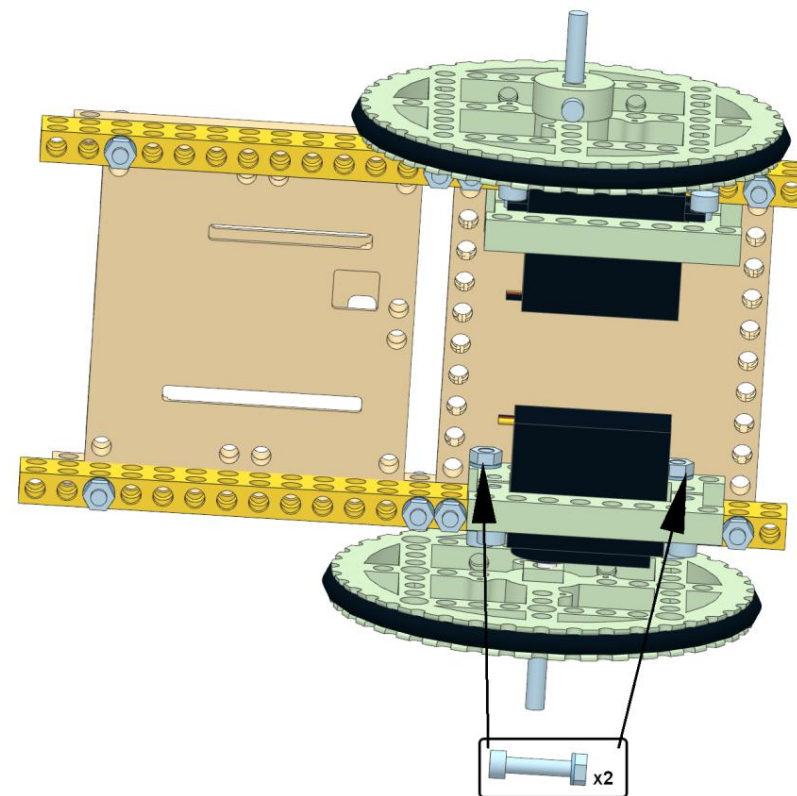
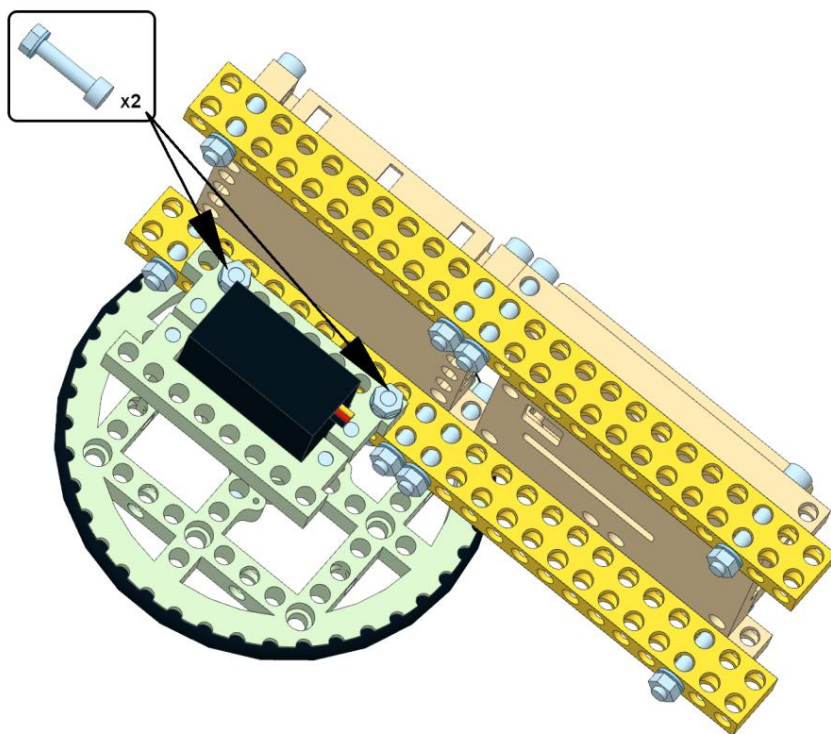
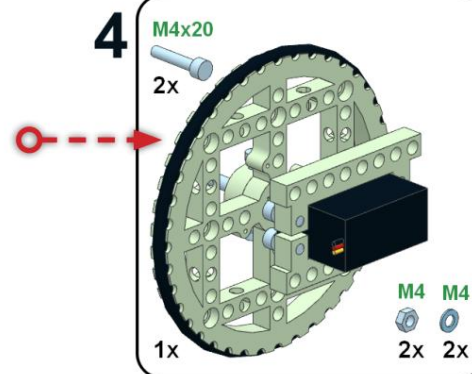




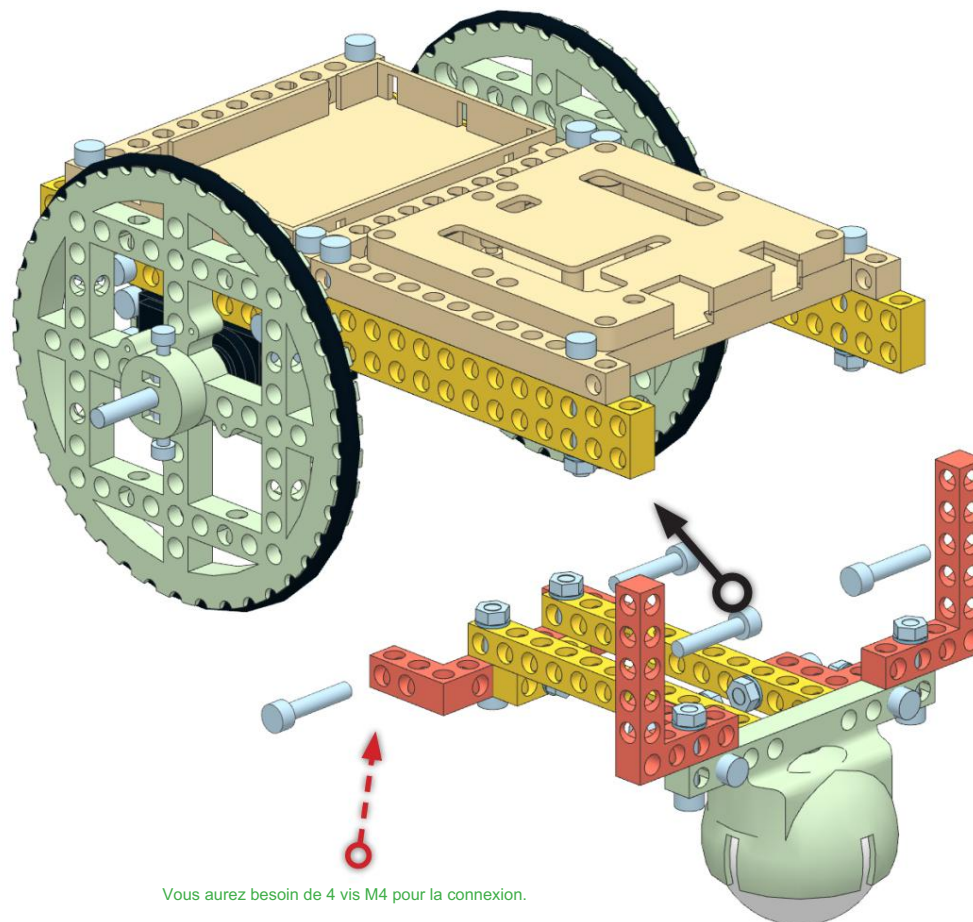


un ensemble de roues gauche et droite à l'avance (roue, joint torique, adaptateur de servo-roue, axe, vis, servomoteur continu, 2 x support de servo - vis à chaque servomoteur).

Des instructions détaillées pour assembler le servo et le jeu de roues se trouvent à la page 17.

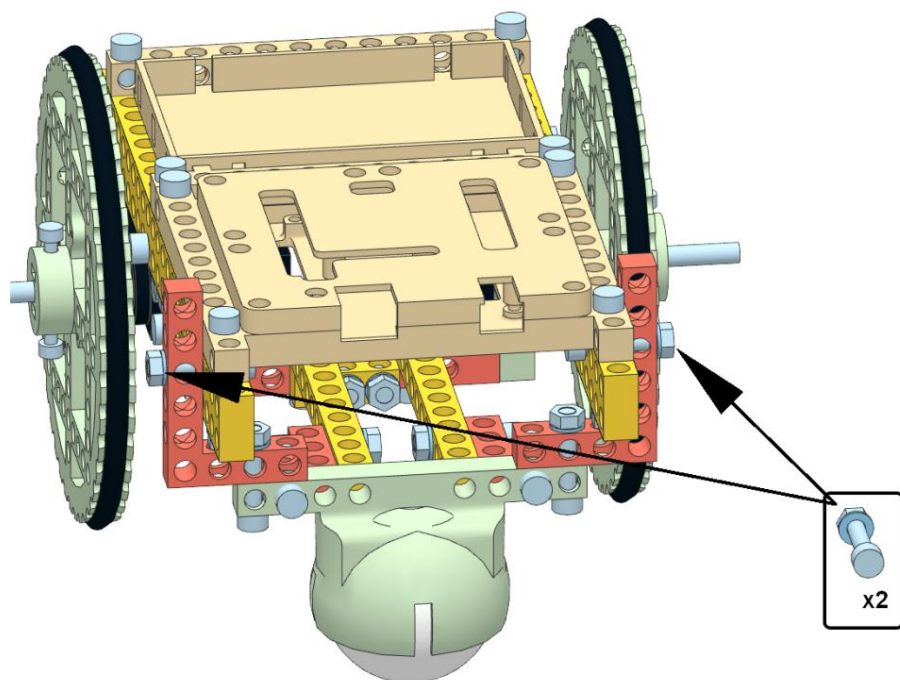
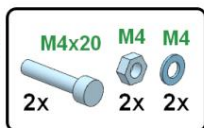


Enfin, nous connectons les parties inférieure et supérieure :

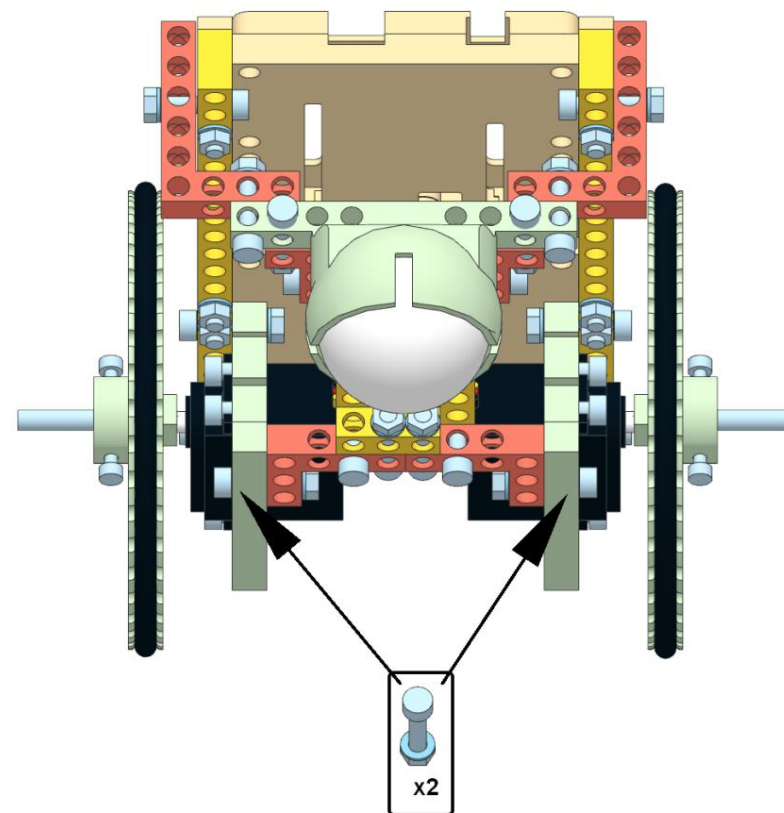
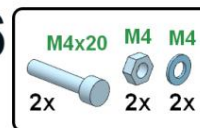


Vous aurez besoin de 4 vis M4 pour la connexion.

5

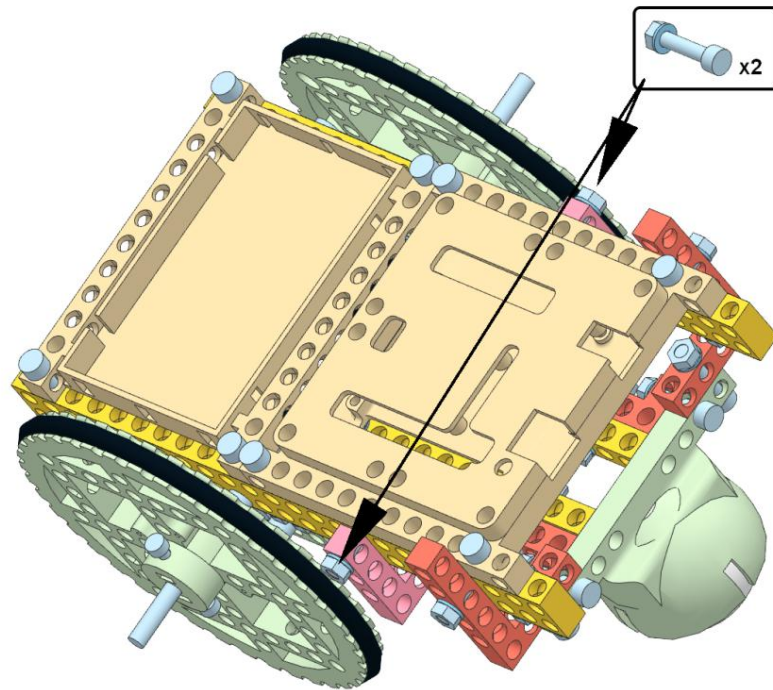
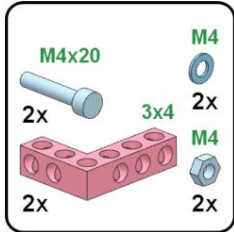


6

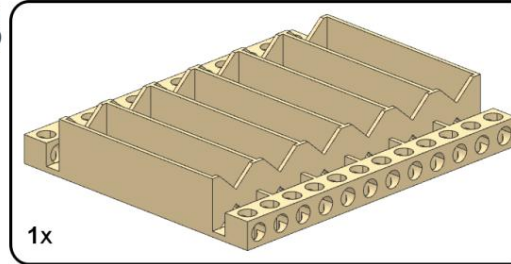




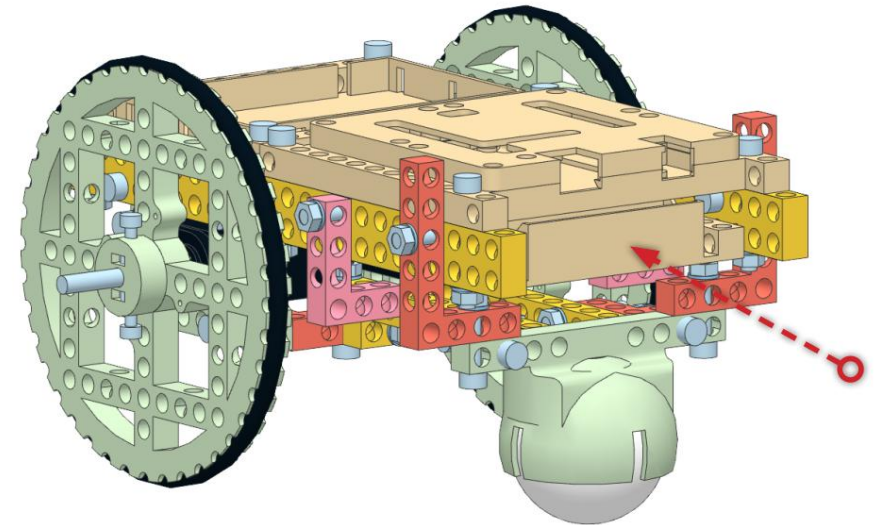
7



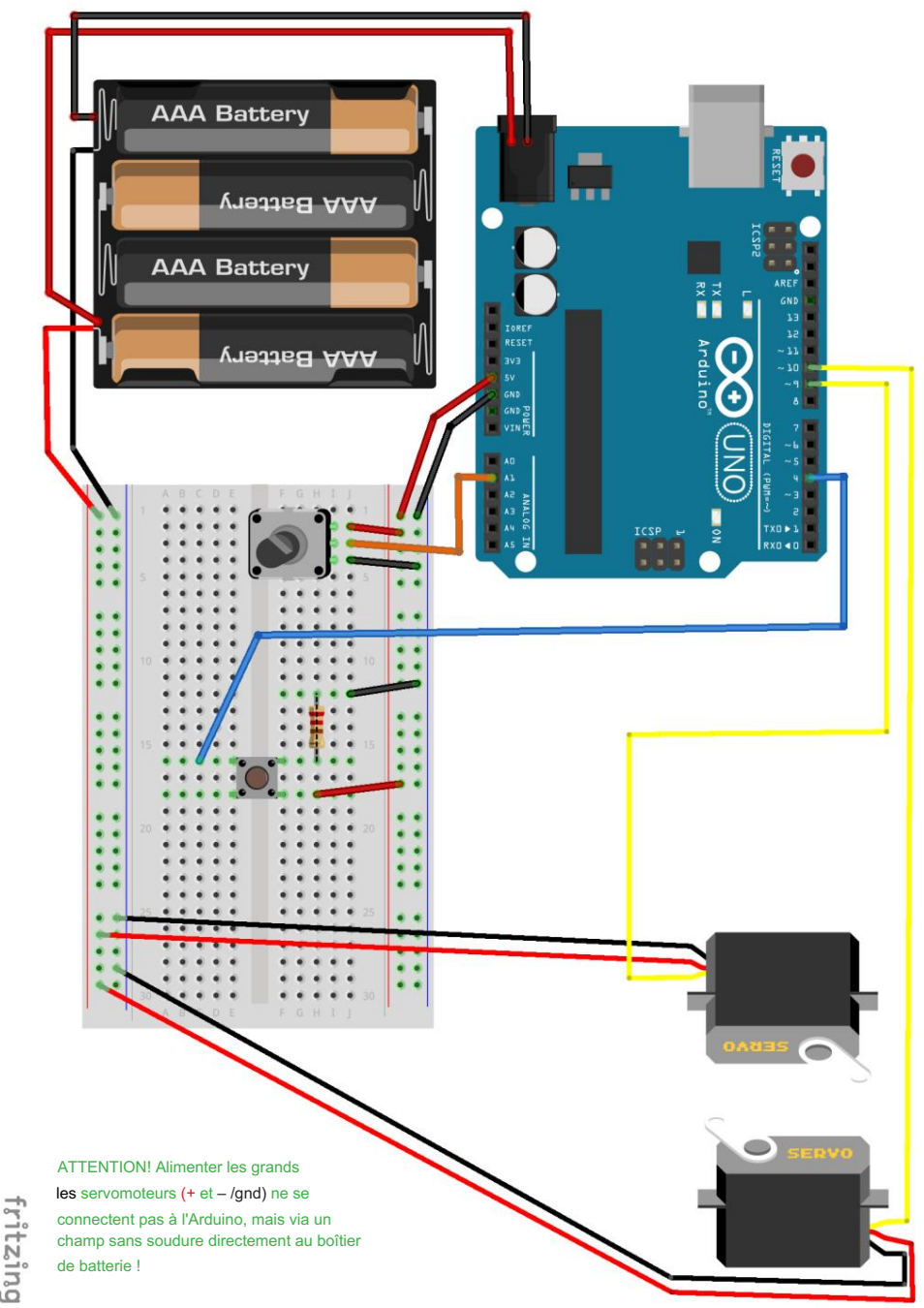
8



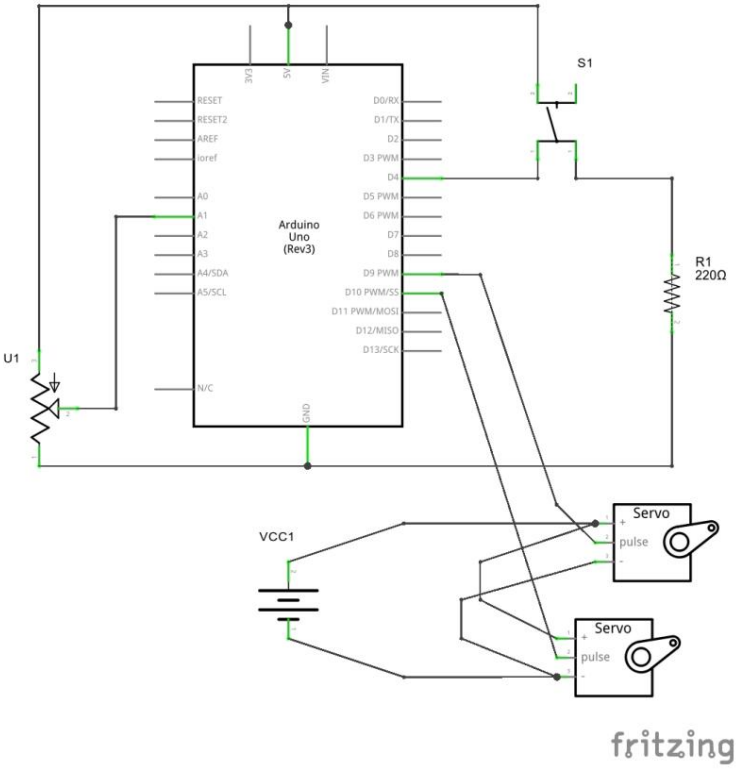
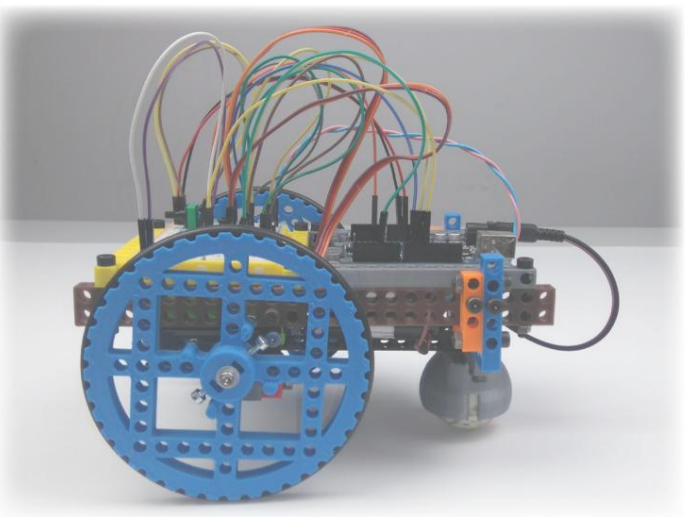
Bien sûr, vous pouvez également utiliser autre chose qu'une boîte imprimée en 3D sur piles.



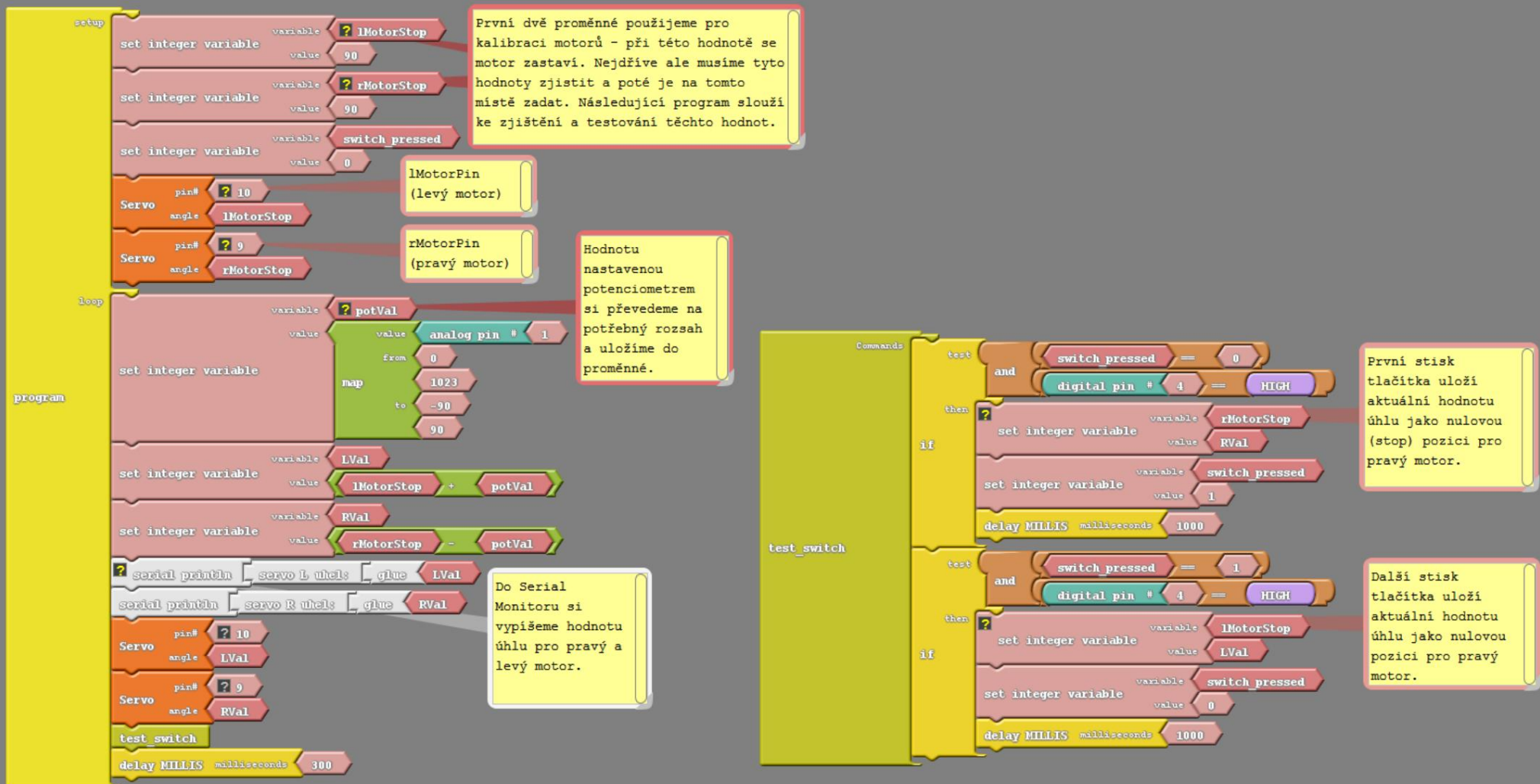
# Câblage de base



ATTENTION! Alimenter les grands  
les servomoteurs (+ et - /gnd) ne se  
connectent pas à l'Arduino, mais via un  
champ sans soude directement au boîtier  
de batterie !



## programme #1 (ArduBlock) - configuration et test





## programme #1 (code personnalisé généré par ArduBlock)

```
#include <Servo.h>
```

```
Servo servo_pin_10 ;
```

```
Servo servo_pin_9 ;
```

```
entier _ABVAR_1_IMotorStop = 0 ; entier
```

```
_ABVAR_2_rMotorStop = 0 ; entier
```

```
_ABVAR_3_switch_pressed = 0 ; entier
```

```
_ABVAR_4_potVal = 0 ; entier _ABVAR_5_LVal = 0 ;
```

```
entier _ABVAR_6_RVal = 0 ;
```

```
annuler test_switch();
```

```
annuler la configuration()
```

```
{
  pinMode( 4 ENTRÉE );
  servo_pin_10.attach(10);
  servo_pin_9.attach(9); Série.begin(9600);
  _ABVAR_1_IArrêtMoteur = 90 ;
  _ABVAR_2_rMotorStop = 90 ;
  _ABVAR_3_switch_pressed = 0 ;
  servo_pin_10.write( _ABVAR_1_IMotorStop );
  servo_pin_9.write( _ABVAR_2_rMotorStop );
}
```

```
boucle vide () {
```

```
  _ABVAR_4_potVal = map ( analogRead(1) , _ABVAR_5_LVal = 0 , 1023 , -90 , 90) ;
  ( _ABVAR_1_IMotorStop + _ABVAR_4_potVal ) ;
  _ABVAR_6_RVal = ( _ABVAR_2_rArrêtMoteur - _ABVAR_4_potVal ) ;
  Serial.print(" angle servo L :");
  Serial.print(_ABVAR_5_LVal); Serial.println();
  Serial.print(" angle servo R :");
  Serial.print(_ABVAR_6_RVal); Serial.println();
  servo_pin_10.write( _ABVAR_5_LVal );
  servo_pin_9.write( _ABVAR_6_RVal ); test_switch();
  retard( 300 );
}
```

```
annuler test_switch() {
```

```
  si ((((_ABVAR_3_switch_pressed) == (0)) && ((digitalRead(4)) == (HIGH)))) {
```

```
    _ABVAR_2_rMotorStop = _ABVAR_6_RVal ;
```

```
    _ABVAR_3_switch_pressed = 1 ; retard( 1000 );
```

```
  } si ((((_ABVAR_3_switch_pressed) == (1)) && ((digitalRead(4)) == (HIGH)))) {
```

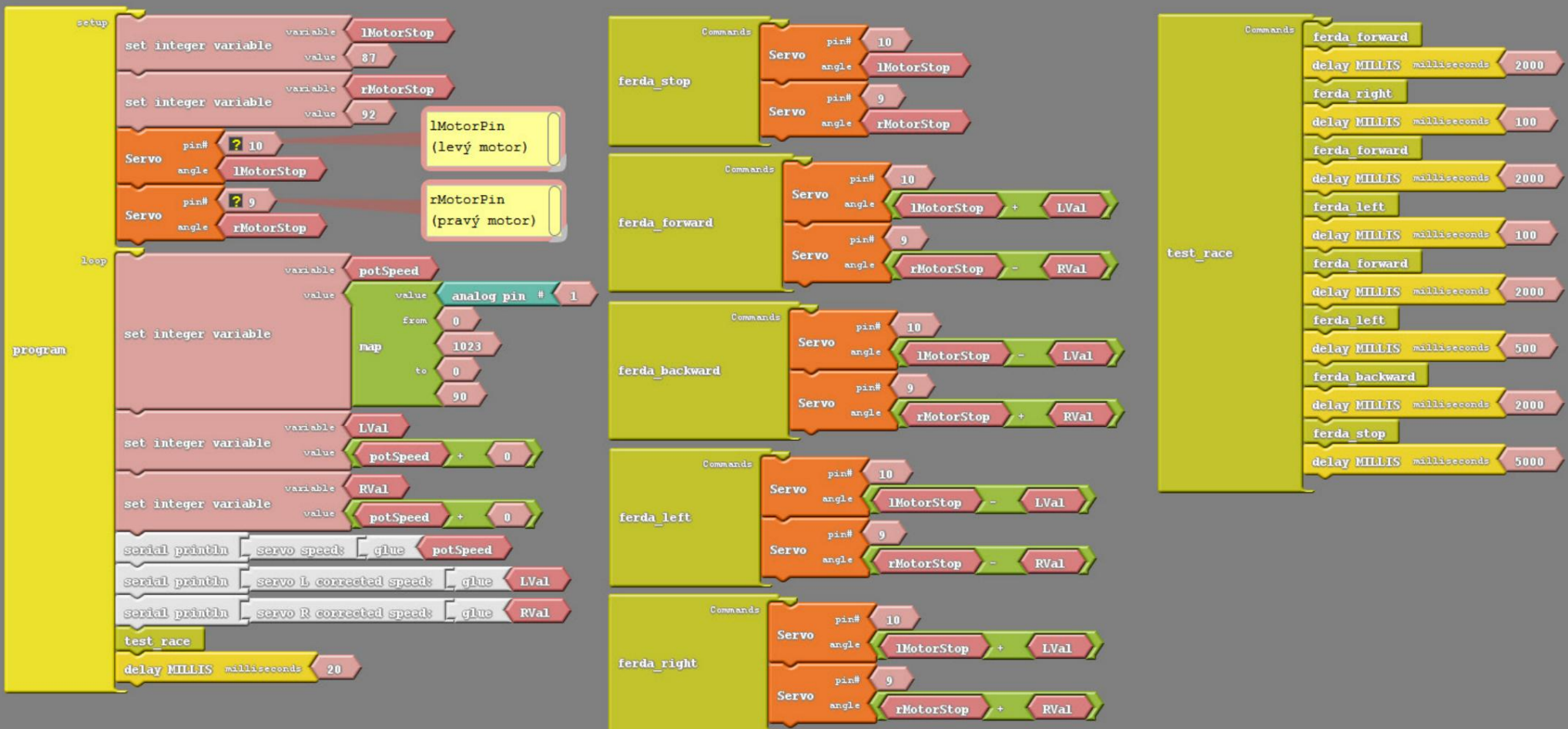
```
    _ABVAR_1_IMotorStop = _ABVAR_5_LVal ;
```

```
    _ABVAR_3_switch_pressed = 0 ; retard( 1000 );
```

```
  }
```

```
}
```

## programme n°2 (ArduBlock) – conduite avec un robot



## programme #2 (code personnalisé généré par ArduBlock)

```
#include <Servo.h>
```

```
Servo servo_pin_10 ;
```

```
Servo servo_pin_9 ;
```

```
entier _ABVAR_1_IMotorStop = 0 ; entier  
_ABVAR_2_rMotorStop = 0 ; entier  
_ABVAR_3_potSpeed = 0 ; entier _ABVAR_4_LVal  
= 0 ; entier _ABVAR_5_RVal = 0 ;
```

```
void ferda_right(); void  
ferda_forward(); void  
ferda_backward(); void test_race();  
void ferda_left(); void ferda_stop();
```

```
annuler la configuration()
```

```
{  
  servo_pin_10.attach(10);  
  servo_pin_9.attach(9); Série.begin(9600);  
  _ABVAR_1_IMotorStop = 87 ;  
  _ABVAR_2_rMotorStop = 92 ;  
  servo_pin_10.write( _ABVAR_1_IMotorStop );  
  servo_pin_9.write( _ABVAR_2_rMotorStop );  
}
```

```
boucle vide () {
```

```
  _ABVAR_3_potSpeed = carte ( analogRead(1) , _ABVAR_4_LVal = 0 , 1023 , 0 , 90 ) ;  
  ( _ABVAR_3_potSpeed + 0 ) ;  
  _ABVAR_5_RVal = ( _ABVAR_3_potSpeed + 0 ) ;  
  Serial.print(" vitesse du servo :");  
  Serial.print(_ABVAR_3_potSpeed); Serial.println();  
  Serial.print(" vitesse corrigée servo L :");  
  Serial.print(_ABVAR_4_LVal); Serial.println(); Serial.print(" vitesse  
corrigée servo R :"); Serial.print(_ABVAR_5_RVal); Serial.println();  
  test_race(); retard( 20 );
```

```
}
```

```
annuler test_race() {
```

```
  ferda_forward();  
  retard( 2000 ); ferda_right();  
  retard( 100 );  
  ferda_forward();  
  retard( 2000 ); ferda_left();  
  retard( 100 );  
  ferda_forward();  
  retard( 2000 ); ferda_left();  
  retard( 500 );  
  ferda_backward();  
  retard( 2000 ); ferda_stop();  
  retard( 5000 );
```

```
}
```

```
annuler ferda_forward() {
```

```
  servo_pin_10.write( ( _ABVAR_1_IMotorStop + _ABVAR_4_LVal ) ); servo_pin_9.write( ( _ABVAR_2_rMotorStop  
- _ABVAR_5_RVal ) );  
}
```

```
annuler ferda_right() {
```

```
  servo_pin_10.write( ( _ABVAR_1_IMotorStop + _ABVAR_4_LVal ) ); servo_pin_9.write( ( _ABVAR_2_rMotorStop  
+ _ABVAR_5_RVal ) );  
}
```

```
void ferda_left() {
```

```
  servo_pin_10.write( ( _ABVAR_1_IMotorStop - _ABVAR_4_LVal ) ); servo_pin_9.write( ( _ABVAR_2_rMotorStop  
- _ABVAR_5_RVal ) );  
}
```

```
annuler ferda_backward() {
```

```
  servo_pin_10.write( ( _ABVAR_1_IMotorStop - _ABVAR_4_LVal ) ); servo_pin_9.write( ( _ABVAR_2_rMotorStop  
+ _ABVAR_5_RVal ) );  
}
```

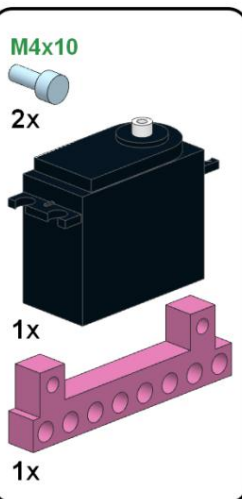
```
annuler ferda_stop() {
```

```
  servo_pin_10.write( _ABVAR_1_IMotorStop );  
  servo_pin_9.write( _ABVAR_2_rMotorStop );  
}
```

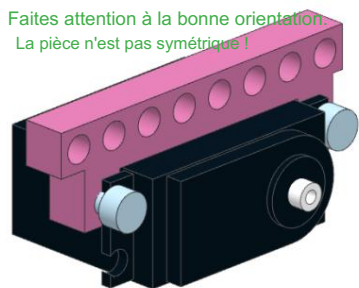


## ANNEXE (montage du jeu de servo-roues)

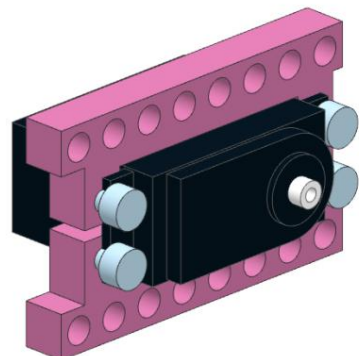
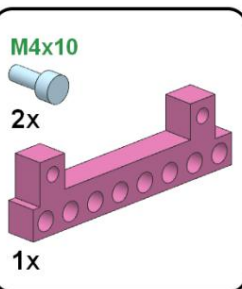
1



Faites attention à la bonne orientation.  
La pièce n'est pas symétrique !



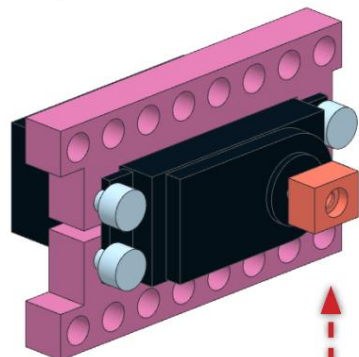
2



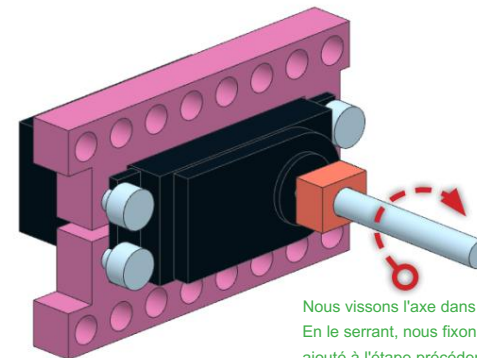
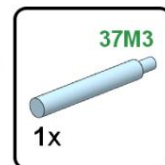
3



La pièce doit être légèrement chauffée (par exemple avec un briquet) et placée sur un axe métallique dentelé au servo afin que cette pièce ne tourne pas. Le collage est également possible, car la pièce est utilisée universellement au sein de m-Bitbeam et peut rester en permanence sur le servo.

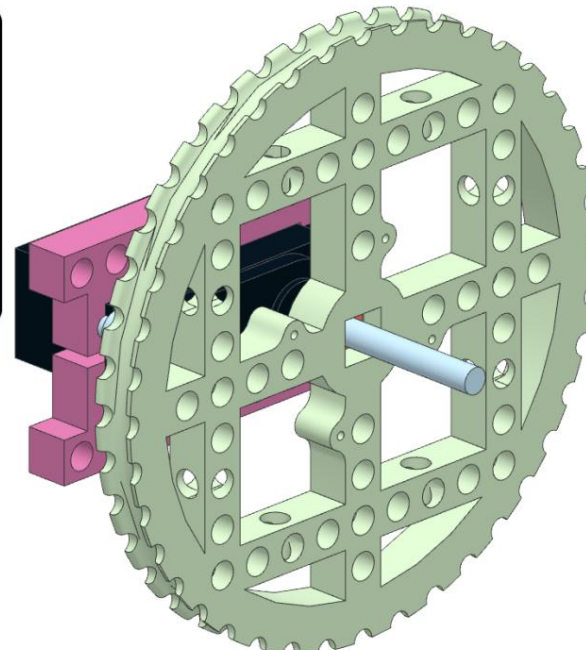
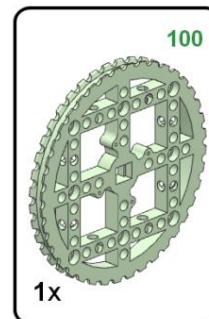


4

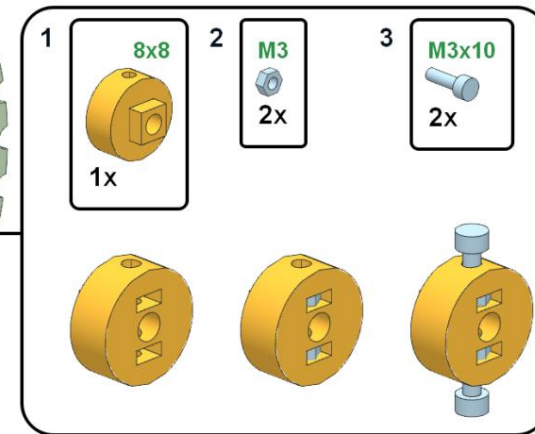
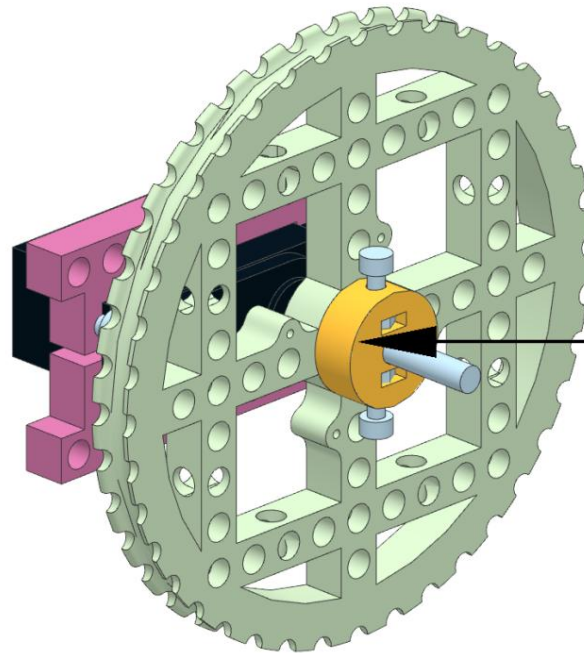


Nous vissons l'axe dans le filetage.  
En le serrant, nous fixons en plus l'embrayage ajouté à l'étape précédente.

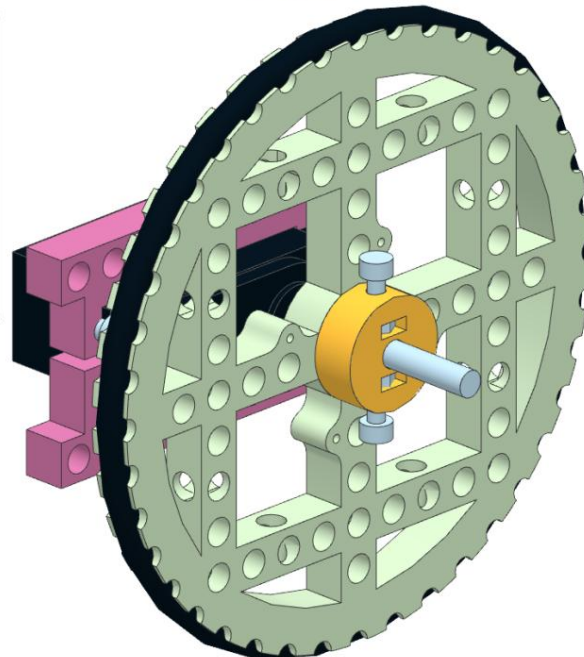
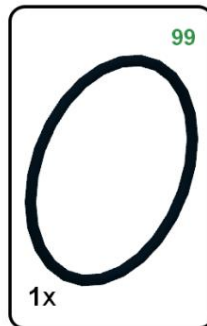
5



6

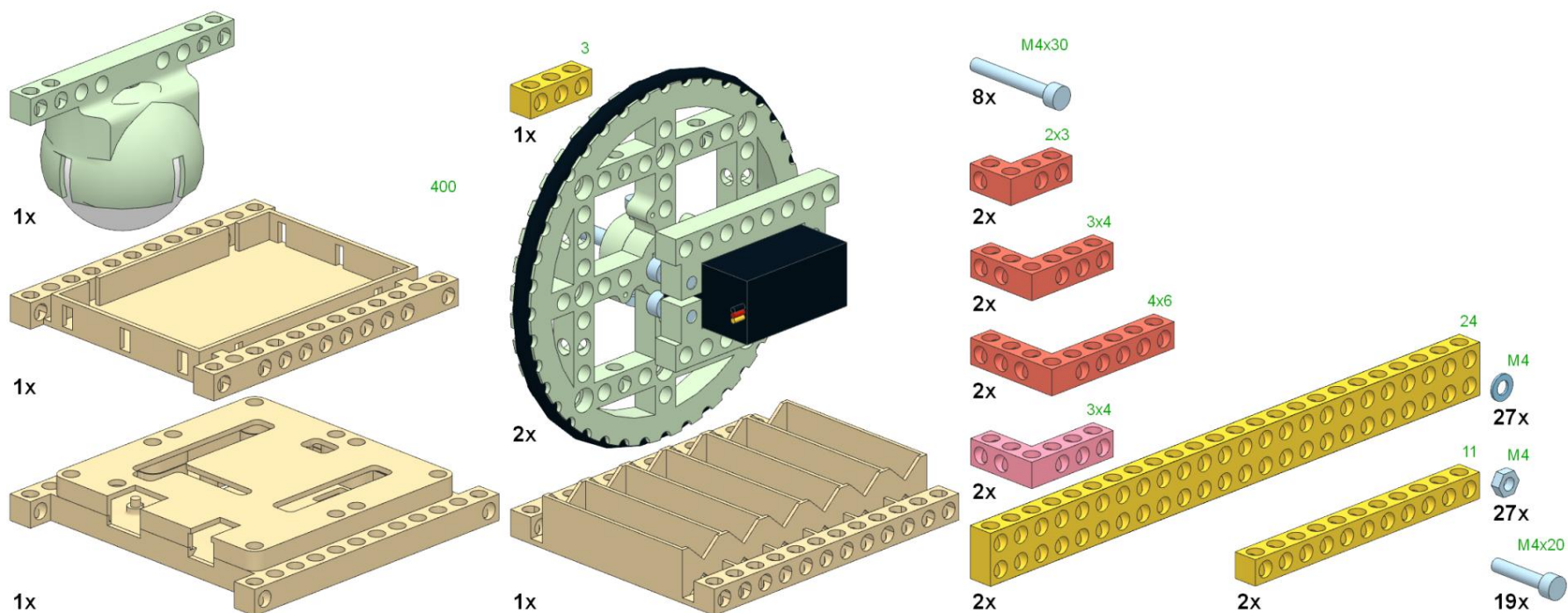


7



# Notice de construction L07 – FERDA

La liste des pièces nécessaires correspond au kit imprimé en 3D m-Bitbeam Education Base Set (set éducatif de base).



Ce guide de construction a été créé à l'aide d'outils SW disponibles gratuitement : MLCad (<http://mlcad.lm-software.com>), LDView (<http://ldview.sourceforge.net>) et LPub (<http://lpub.sourceforge.net>).

Si vous êtes intéressé par ce guide de construction sous forme imprimée, contactez-moi à [skolniprojekty@gmail.com](mailto:skolniprojekty@gmail.com).