

# **TEKBOT ROBOTICS**

# **CHALLENGE 2025**

Documentations des Tests Techniques  
de présélection

## Sommaire

1. Contexte général
2. Tableau récapitulatif des tests de présélection
3. Roadmap des tests de présélection
4. Tests de présélection
  - 4.1. Tests Electronique
  - 4.2. Tests IT
  - 4.3. Tests Mécanique
  - 4.4. Test Final

## 1. Contexte général

Dans le cadre de la **Tekbot Robotics Challenge 2K25**, un processus de présélection rigoureux, s'étalant sur **cinq semaines**, est mis en place afin d'évaluer les compétences techniques des équipes participantes. **Chaque semaine, une nouvelle compétence sera mise à l'épreuve.** Ainsi chaque **sous-équipe** de vos **équipes respectives** devront démontrer leur expertise en **électronique, mécanique et informatique** à travers des **épreuves techniques**.

L'objectif de cette présélection est d'identifier les équipes les plus **qualifiées, polyvalentes** et capables de **travailler en synergie**. À la fin du parcours, un **test final** rassemblera **toutes les sous-équipes de chaque équipe** afin de relever un **défi multidisciplinaire**.

## 2. Tableau récapitulatif des tests de présélection

Domaine	Tests	Durée
<b>Électronique</b>	Test 1: Test input : Gyroscope et accéléromètre	1 semaine
	Test 2: Test communication : La boîte noire	1 semaine
	Test 3: Test output: Réalisation d'un afficheur 7 segments	1 semaine
<b>IT</b>	Test 1: Réalisation d'un système de classe de gestion d'un Robot	1 semaine
	Test 2: Introduction à ROS2	1 semaine
	Test 3: Création d'un Algorithme de Pathfinding	1 semaine
<b>Mécanique</b>	Test 1: Niveau Débutant	1 semaine
	Test 2: Niveau Intermédiaire	1 semaine
	Test 3: Niveau Avancé	1 semaine
<b>Électronique, IT, Mécanique</b>	Test final : Réalisation d'un système de convoyeur	2 semaines

## 3. Roadmap des tests de présélection

Semaine : Date	Tests
<b>Semaine 1 : 05/06/2025 au 12/06/2025</b> Réalisation et documentation des tests 1 <b>Review - 12/06/2025</b> 16h-18h : présentation et notation, Consultation des ressources github déposées par chaque équipe et finalisation de la notation.	Test 1 (électronique, IT, mécanique)

<p><b>Semaine 2 : 12/06/2025 au 19/06/2025</b>  Réalisation et documentation des tests 2  <b>Review - 19/06/2025 :</b>  16h-18h : présentation et notation,  Consultation des ressources github déposées par chaque équipe et finalisation de la notation.</p>	Test 2 (électronique, IT, mécanique)
<p><b>Semaine 3 : 19/06/2025 au 26/06/2025</b>  Réalisation et documentation des tests 3  <b>Review - 26/06/2025 :</b>  16h-18h : présentation et notation,  Consultation des ressources github déposées par chaque équipe et finalisation de la notation.</p>	Test 3 (électronique, IT, mécanique)
<p><b>Semaine 4 et 5 : 26/06/2025 au 10/07/2025</b>  Réalisation et documentation du test final  <b>Final Review - 10/07/2025 :</b>  16h - 19h : présentation et notation (meet général avec toutes les équipes),  - Consultation des ressources github déposées par chaque équipe et finalisation de la notation.</p>	<b>Test final</b>

## 4. Tests de présélection

### 4.1. Tests Electronique

**Test input :** Gyroscope et accéléromètre

**Durée :** 1 semaine

#### Présentation du test

L'**orientation dans l'espace** ou plus simplement la capacité à reconnaître sa gauche, sa droite, le haut et le bas est une compétence que l'on acquiert au cours de notre enfance. Nous arrivons à **nous repérer dans l'espace grâce à nos sens** et puisant dans la connaissance commune des Hommes, nous attribuons un nom à chaque espace autour de nous. Ainsi nous avons tous la même droite et la même gauche lorsque nous sommes dans une file indienne par exemple. **Mais comment font les robots pour se repérer dans l'espace ?**

L'évolution de la technologie et particulièrement celle de l'électronique prend en compte notre capacité à exploiter les informations de notre environnement. Ainsi apparaît le terme de **capteur**. Un capteur peut être définie comme étant **un dispositif électronique capable de convertir des données environnementales ou physiques en signaux électriques**. Il existe aujourd'hui une panoplie de capteurs mais ceux qui nous intéressent sont ceux **capables de donner la vitesse d'un objet ainsi que sa rotation ou son orientation** dans l'espace. C'est en effet avec ces types de capteurs que **les robots peuvent se repérer dans l'espace**.

Pour la première épreuve de sélection en électronique, vous aurez à **identifier un capteur** exécutant cette fonction et **l'utiliser pour donner l'orientation** dans l'espace ainsi que **la vitesse de votre main**.

### Description technique

Ce projet vise à évaluer votre capacité à identifier puis exploiter les valeurs d'un capteur gyroscope et accéléromètre. Afin de valider cette première étape de sélection nous attendons de vous futures champions de la TRC2K25 que vous :

- Identifiez un capteur combinant les fonctions gyroscope et accéléromètre puis expliquez son fonctionnement dans votre documentation. Le capteur doit communiquer par I2C.
- Placez le capteur dans **la paume de main d'un membre de l'équipe** puis la bouger dans toutes les directions : vers le haut, vers le bas, vers la droite, vers la gauche, vers l'avant et vers l'arrière.
- Réalisez un code Arduino bien détaillé qui exploite les données fournies par le capteur pour indiquer sur un écran LCD le sens de déplacement de votre main ainsi que son accélération.
- Faites un schéma électronique sous KICAD. Vous n'avez pas à designer le PCB.
- Fabriquez votre propre alimentation électrique pour alimenter le circuit.
- Documentez votre travail sur le dépôt GitHub qui vous sera attribué

**NB:**

Vous n'êtes pas obligés d'imprimer en 3D le projet à réaliser. Vous pouvez utiliser les composants disponibles autour de vous. Cependant, veillez à soigner la présentation du projet et à faire preuve d'innovation.

Pour ce projet l'utilisation de la carte Arduino ou de breadboard dans la présentation des résultats n'est pas interdite. Néanmoins la bonne gestion des câbles et l'esthétique du rendu final sera notée. Ensuite, la vidéo de présentation des résultats doit être prise de sorte que nous puissions voir le capteur dans la main, l'affichage sur l'écran et tous les membres de la team.

### Grille de notation

Nous partageons avec vous la notation du test.

Le test sera noté sur 100 points répartis comme suit :

1. La circuiterie (**25 points**) :

- la qualité du schéma électronique : le choix des composants et la pertinence de leurs rôles
- la gestion et la sécurité de l'alimentation. La sécurité de l'alimentation prend en compte les protections contre les courts circuits et le respect des niveaux de tension et de courant pour chaque composant.
- la gestion des câbles.

2. Le code (**20 points**):

- la facilité de compréhension du code au travers des commentaires

- l'optimisation du code, c'est-à-dire la méthodologie exploitée pour écrire le code
- la lisibilité du code (respecter les indentations)

NB: Le code sera évalué durant la rencontre de présentation en ligne du projet; il ne devra donc pas être présenté en détail dans votre documentation.

3. Le test du fonctionnement (**25 points**)
4. La structure de la documentation et sa facilité de compréhension (**25 points**).  
L'accent devra être mis sur le fonctionnement du capteur.
5. La présentation (**5 points**)
  - la qualité des slides
  - la maîtrise du sujet
  - l'aptitude à répondre aux questions

**Test communication :** La boîte noire

**Durée :** 1 semaine

### Présentation du test

Dans l'industrie de l'automobile, de l'aviation ou même des trains, **les boîtes noires sont des éléments indispensables** pour le suivi de l'activité des appareils. Les boîtes noires sont en effet des **systèmes d'enregistrement de divers paramètres** relatifs à l'activité des engins. Dans le cas d'un avion, la boîte noire permet d'enregistrer les communications des pilotes ainsi que toutes les données de vol de l'avion comme la vitesse, l'altitude et même l'orientation dans l'espace de l'appareil grâce à des capteurs de pointe. Ce sont ces données recueillies qui servent en cas de crash à déterminer les causes de l'incident.

Dans le cadre de la Tekbot Robotics Challenge, vous aurez à **concevoir une boîte noire** capable d'**enregistrer les données de vitesse et de position dans l'espace** grâce au **capteur gyroscope et accéléromètre** que vous avez utilisé la semaine précédente. La boîte noire devra **transmettre** ensuite en temps réel **les données collectées à une station de contrôle via I2C** où les informations pourront être visualisées sur un écran LCD. Le projet se décline donc en deux éléments : **une boîte qui collecte et transmet** les données et **une station de contrôle** qui **reçoit et affiche** ces données. La présentation du projet consistera donc à montrer une vidéo où vous faites bouger la boîte dans l'espace et visualisez la vitesse et les variations de la position de la boîte sur les axes au niveau du poste de contrôle.

### Description technique

Ce projet est un prolongement du précédent car il emploie le même capteur. Son succès dépend donc de votre exploit lors du premier test. Cette fois ci le niveau de complexité augmentera d'un cran car vous devrez :

- utiliser directement les microcontrôleurs Atmega328P dans vos circuits finaux et non les cartes Arduino. Vous devrez réaliser des schémas électronique avec KICAD et designer les PCB pour ensuite les produire.
- Fabriquer votre propre alimentation électrique pour alimenter le circuit. Elle ne devra pas être incluse dans la boîte.
- fabriquer un cube de 7 cm d'arrêt pour représenter la boîte noire. La face supérieure du cube devra être ouverte si elle est faite dans un matériau opaque afin de permettre de voir le circuit à l'intérieur.
- créer un bus I2C dont le seul maître sera le microcontrôleur à l'intérieur du cube. Le capteur (dans le cube) et le microcontrôleur (au niveau de la station de contrôle) devront être les esclaves sur le bus. L'écran LCD doit être branché en mode 4 bits.
- Comme dans le challenge de la semaine précédente, vous devrez faire bouger le cube dans toutes les directions et visualiser les informations sur l'écran LCD.
- Documenter votre travail sur le dépôt GitHub qui vous sera attribué

**NB:**

Vous n'êtes pas obligés d'imprimer en 3D le projet à réaliser. Vous pouvez utiliser les composants disponibles autour de vous. Cependant, veillez à soigner la présentation du projet et à faire preuve d'innovation.

Pour ce projet l'utilisation de la carte Arduino ou de breadboard dans la présentation des résultats est interdite. Vous devrez soit réaliser un PCB soit vous servir de veroboard pour assembler les composants. Ensuite, la vidéo de présentation des résultats doit être prise de sorte que nous puissions voir le capteur à l'intérieur du cube.

### Grille de notation

Nous partageons avec vous la notation du test.

Le test sera noté sur 100 points répartis comme suit :

1. La circuiterie (**25 points**) :

- la qualité du schéma électronique : le choix des composants et la pertinence de leurs rôles,
- l'optimisation et l'esthétique du design du PCB,
- La conformité entre le rendu 3D et la réalisation physique du PCB
- la finesse des soudures et l'esthétique de la carte électronique,
- la gestion et la sécurité de l'alimentation. La sécurité de l'alimentation prend en compte les protections contre les courts circuits et le respect des niveaux de tension et de courant pour chaque composant.
- la gestion des câbles.

NB: La présence de la carte Arduino ou de breadboard dans votre circuit final vous donne un malus de 13 points. Votre circuit sera donc noté sur 12 points au lieu de 25.

2. Le code (**25 points**):

- la facilité de compréhension du code au travers des commentaires
- l'optimisation du code, c'est-à-dire la méthodologie exploitée pour écrire le code
- la lisibilité du code (respecter les indentations)

NB: Le code sera évalué durant la rencontre de présentation en ligne du projet; il ne devra donc pas être présenté en détail dans votre documentation.

3. La réalisation physique de la boîte (**10 points**):
  - l'esthétique de la boîte
  - l'esthétique du poste de contrôle
4. Le test du fonctionnement (**25 points**).
5. La structure de la documentation et sa facilité de compréhension (**10 points**).
6. La présentation (**5 points**)
  - la qualité des slides
  - la maîtrise du sujet
  - l'aptitude à répondre aux questions.

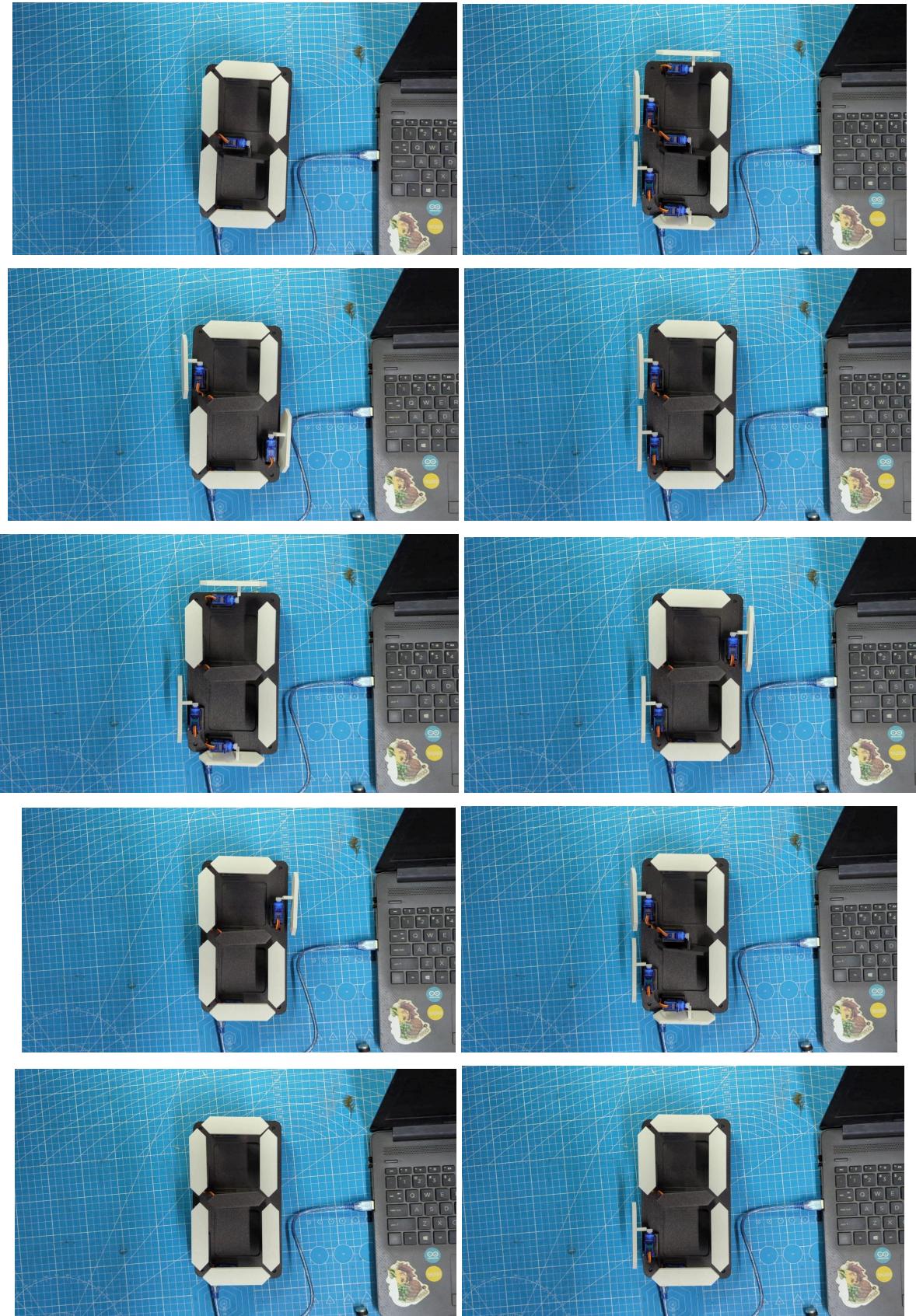
**Test output:** Réalisation d'un afficheur 7 segments

**Durée :** 1 semaine

### Présentation du test

Qui a dit que les afficheurs 7 segments devraient toujours être lumineux ? Teckbot City étant la cité de l'innovation, nous ferons autrement. Mais quand nous disons nous, il s'agit de vous, les futurs champions de la Teckbot Robotics Challenge. Vous aurez en effet à nous prouver une fois de plus votre aptitude à innover. Plus précisément, vous devrez **réaliser un digit d'afficheur 7 Segments avec des servomoteurs** (appelons ça un afficheur 7 servos) et y **afficher les chiffres 0 à 9**.

À la fin de la phase de développement, vous devrez présenter votre travail avec une **documentation bien détaillée** qui contiendra tous les **schémas électriques réalisés sous KICAD** (le zip du projet KICAD comprenant les **fichiers KICAD et les images des rendus 3D des circuits**) **les codes Arduino**, les **Datasheets des composants** électroniques utilisés ainsi qu'une vidéo d'utilisation. La vidéo montrera **l'afficheur compter de 0 à 9 puis de 9 à 0 avec une vitesse d'une seconde par incrémentation**.



### Description technique

Le défi que vous devrez relever ici est de vous servir de 7 servomoteurs (un pour chaque segment) et les piloter dans la bonne configuration pour représenter les chiffres. Il s'agira ensuite d'afficher les chiffres dans un ordre croissant puis décroissant.

Il vous faudra :

- Utiliser directement un microcontrôleur Atmega328P dans le circuit final et non une carte Arduino. Vous devrez réaliser des schémas électroniques avec KICAD et concevoir les PCB pour ensuite les produire.
- Alimenter le circuit avec des batteries au lithium.
- Ne pas utiliser de fonction bloquante telle que "**delay()**" dans le code Arduino.
- Documenter votre travail sur le dépôt GitHub qui vous sera attribué

**NB:**

Vous n'êtes pas obligés d'imprimer en 3D le projet à réaliser. Vous pouvez utiliser les composants disponibles autour de vous. Cependant, veillez à soigner la présentation du projet et à faire preuve d'innovation.

Pour ce projet l'utilisation de la carte Arduino ou de breadboard dans la présentation des résultats est interdite. Vous devrez soit réaliser un PCB soit vous servir de veroboard pour assembler les composants. Vous avez la possibilité d'utiliser un module PCA9685.

### Grille de notation

Nous partageons avec vous la notation du test.

Le test sera noté sur 100 points répartis comme suit :

1. La circuiterie (**25 points**) :

- la qualité du schéma électronique : le choix des composants et la pertinence de leurs rôles,
- l'optimisation et l'esthétique du PCB,
- La conformité entre le rendu 3D et la réalisation physique du PCB
- la finesse des soudures et l'esthétique de la carte électronique,
- la gestion et la sécurité de l'alimentation. La sécurité de l'alimentation prend en compte les protections contre les courts circuits et le respect des niveaux de tension et de courant pour chaque composant,
- la gestion des câbles.

NB: La présence de la carte Arduino ou de breadboard dans votre circuit final vous donne un malus de 13 points. Votre circuit sera donc noté sur 12 points au lieu de 25.

2. Le code (**20 points**):

- la facilité de compréhension du code au travers des commentaires,
- l'optimisation du code, c'est-à-dire la méthodologie exploitée pour écrire le code,
- la lisibilité du code (respecter les indentations).

NB: Le code sera évalué durant la rencontre de présentation en ligne du projet; il ne devra donc pas être présenté en détail dans votre documentation.

3. La réalisation physique de l'afficheur (**20 points**):
  - l'aspect esthétique de la maquette,
  - l'ingéniosité pour matérialiser les segments de l'afficheur.
4. Le test du fonctionnement (**20 points**).
5. La structure de la documentation et sa facilité de compréhension (**10 points**).
6. La présentation (**5 points**)
  - la qualité des slides
  - la maîtrise du sujet
  - l'aptitude à répondre aux questions.

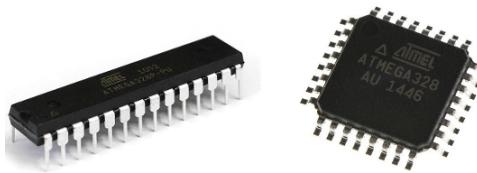
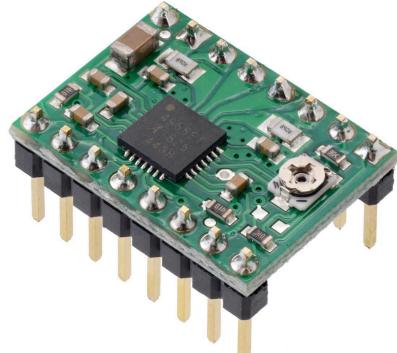
### **Recommandation technique**

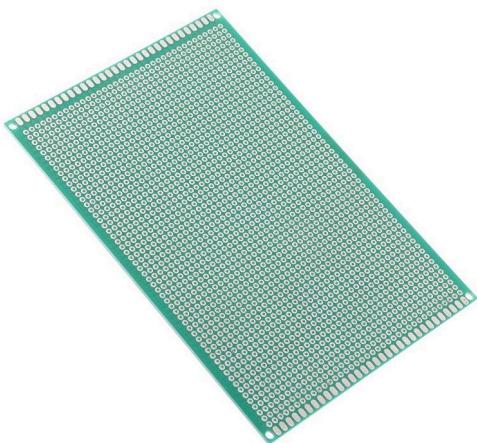
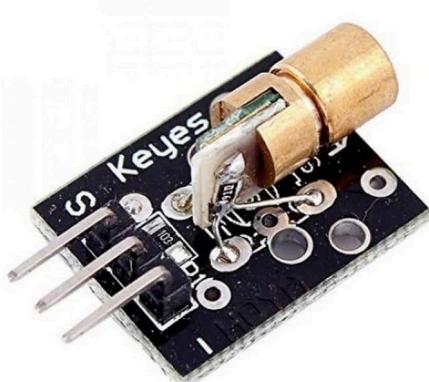
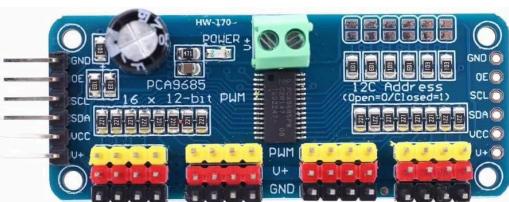
Nous partageons avec vous ci-dessous une liste de logiciels et matériels que nous vous recommandons d'utiliser.

#### **Logiciels :**

Nom	Rôle et lien de téléchargement	Logo
KICAD	Ce logiciel est idéal pour le dessin des circuits électroniques et le design des circuits imprimés. Vous pouvez le télécharger via le lien suivant : <a href="https://www.kicad.org">https://www.kicad.org</a>	
Arduino IDE	Le logiciel de programmation Arduino est un outil pratique et simple d'utilisation. Vous pourrez l'obtenir sous le lien suivant dans l'onglet software : <a href="https://www.arduino.cc">https://www.arduino.cc</a>	

#### **Matériels :**

Nom	Rôle et lien de documentation	Image
<b>Microcontrôleur ATmega328P</b>	<p>Il s'agit du microcontrôleur au cœur de la carte Arduino Uno. C'est un microcontrôleur d'architecture AVR du constructeur Atmel disposant des caractéristiques idéales pour la plupart de vos projets grâce à ces périphériques internes assez variés.</p> <p>Vous pourrez en savoir plus via le lien 1; le lien 2 vous permettra d'étudier la correspondance entre les noms des bits de port du microcontrôleur et les broches de la carte Arduino.</p> <ol style="list-style-type: none"> <li>1. <a href="#">ATmega48A/PA/88A/PA/168A/PA/328/P</a></li> <li>2. <a href="https://docs.arduino.cc/resources/datasheets/A000066-data-sheet.pdf">https://docs.arduino.cc/resources/datasheets/A000066-data-sheet.pdf</a></li> </ol>	
<b>Moteur pas à pas Nema 17</b>	<p>Ce moteur pas à pas est idéal pour le projet de l'ascenseur en raison de son grand couple et de la précision de ses pas. Vous pourrez en savoir plus via le lien suivant :</p> <p><a href="https://www.handsontec.com/datasheets/17HS4401S.pdf">https://www.handsontec.com/datasheets/17HS4401S.pdf</a></p>	
<b>Driver de moteur pas à pas Pololu A4988</b>	<p>Ce driver vous permettra de contrôler de façon précise le moteur pas à pas. Il offre de nombreuses possibilités pour exploiter la puissance et la précision du moteur pas à pas. Vous pourrez le découvrir en utilisant le lien ci après:</p> <p><a href="https://www.pololu.com/product/1182">https://www.pololu.com/product/1182</a></p>	

<b>Veroboard</b>	<p>Le veroboard ou platine d'essaie est une plaque isolante possédant des plaques ou des pastilles de cuivre sur une face. Il permet d'assembler les composants électroniques pour constituer des circuits de tous les niveaux de complexité possibles. L'usage des verobards demande un grand soin et une bonne méthodologie mais permet de se passer des circuits imprimés classiques.</p>	
<b>Emetteur laser KY-008</b>	<p>Il s'agit d'un module qui intègre une diode laser et sa résistance de protection. La lumière laser émise par la diode est suffisamment puissante pour parcourir une grande distance et résiste bien à l'influence des lumières environnantes. Combiné avec une photorésistance ou son récepteur dédié, il sera un allier idéal pour détecter la présence des déchets sur le tapis du convoyeur dans le projet de sélection final.</p>	
<b>PCA9685</b>	<p>Le module <b>PCA9685</b> est un contrôleur PWM (modulation de largeur d'impulsion) conçu pour contrôler plusieurs périphériques, comme des servomoteurs ou des LED, à partir d'un microcontrôleur. Il est très pratique pour gérer jusqu'à <b>16 sorties PWM</b> à partir d'une seule interface I2C. Il pourrait être très utile pour la réalisation de l'afficheur 7 segments.</p>	

### Tips

Afin de faciliter la réalisation de vos projets nous vous suggérons de suivre les conseils ci-après.

- Complétez la documentation de vos travaux de façon journalière afin de retracer toute l'évolution de vos travaux.
- Élaborez une documentation très explicite sur un site web relié à un repo GitHub afin d'exposer votre travail. Souvenez-vous que la documentation nous permettra de juger vos compétences. Plus détaillée sera la documentation, plus nous aurons d'éléments pour analyser votre travail.
- La documentation devra contenir des images et des vidéos claires nous permettant de voir vos réalisations. Elle devra aussi contenir les **schémas électroniques réalisés sous KICAD, les codes Arduino, les datasheets des composants** électroniques utilisés.
- Écrivez un code Arduino impeccable. Le code devra en effet être lisible (bien indenté), succinct, le plus optimisé possible et surtout bien commenté. Veillez à respecter la structure qui vous est présentée dans la suite du document.
- Pour la documentation des circuits électroniques, vous devrez inclure les schémas électroniques, le tracé des pistes du PCB et le rendu 3D des circuits réalisés dans KICAD. Vous devrez inclure aussi les fichiers zippés des différents projets réalisés.

### Fabrication de PCB

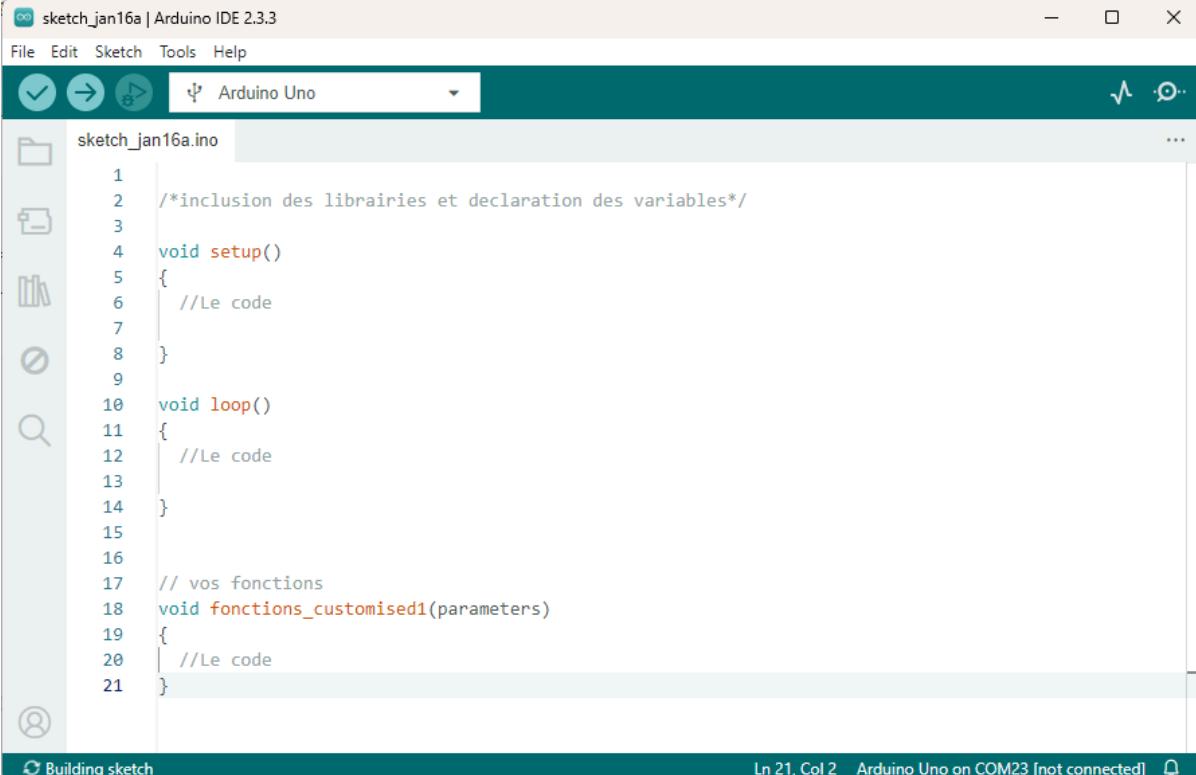
Dans la fabrication de PCB, il est parfois essentiel d'adopter certaines bonnes pratiques, notamment

- Maintenir une distance raisonnable entre les pistes.
- Respecter une largeur de piste minimale afin d'éviter des tracés trop fins, susceptibles de surchauffer ou de se détériorer.
- Veiller à un placement optimal des composants.
- Optimiser les connexions des capteurs et actionneurs avec le PCB en utilisant des Header Pins ou des TBlock.
- Veillez à sélectionner les footprints correspondant précisément aux composants réels.

### Structuration du code dans ARDUINO IDE

La structure de votre code devra être conforme à celle des codes présentés dans les images ci-dessous.

#### Structure générale



```

1 //inclusion des librairies et declaration des variables/
2
3
4 void setup()
5 {
6     //Le code
7
8 }
9
10 void loop()
11 {
12     //Le code
13
14 }
15
16
17 // vos fonctions
18 void fonctions_customised1(parameters)
19 {
20     //Le code
21 }

```

Building sketch      Ln 21, Col 2 Arduino Uno on COM23 [not connected]

NB : La déclaration des variables se fera suivant la conventions de nommage **lower CamelCase**

Exemple : maVariable; nombreDePersonnes; temperatureActuelle

### Structure itérative

```

// L'utilisation des structures itératives for et while devront se faire comme suit :
for (initialisation; condition; incrément)
{
    // Bloc de code à exécuter
}

while (condition)
{
    // Bloc de code à exécuter tant que la condition est vraie
}

```

### Structure conditionnelle

```
// L'utilisation des structures conditionnelles if et switch devront suivre les structures suivantes :  
if (condition)  
{  
    // Bloc de code si la condition est vraie  
}  
else if (autre_condition)  
{  
    // Bloc de code si la deuxième condition est vraie  
}  
else  
{  
    // Bloc de code si aucune condition n'est remplie  
}  
  
switch (maVariable)  
{  
    case 1:  
        //instruction;  
}
```

## 4.2. Tests IT

### Test 1 : Création d'une classe pour un Robot

#### Présentation du test

**Langages de programmation :** C++, Python.

Pour ce premier test du **Tekbot Robotics Challenges 2024**, vous allez démontrer vos compétences en programmation orientée objet en réalisant une classe de gestion d'un Robot. Vous aurez à concevoir une structure flexible permettant la création de sous-classes et de méthodes dédiées aux actions du robot. Ce test est l'occasion de montrer votre capacité à utiliser des principes comme l'encapsulation, l'héritage et le polymorphisme.

À la fin de la phase de développement, vous devrez présenter votre travail avec une documentation détaillée, y compris des schémas UML pour expliquer l'architecture de votre code. Vous devrez notamment implémenter une méthode `move()`, qu'il faudra redéfinir dans les sous-classes dérivées.

#### Ce qu'il faut faire

- **Langages de programmation :** C++, Python.
  - Vous devez créer une classe `Robot` avec des fonctionnalités de gestion et d'héritage. Il faudra :
    - Créer la classe principale `Robot`.
    - Ajouter au moins deux sous-classes.
    - Implémenter une méthode `move()` dans la classe principale et la redéfinir dans les sous-classes.
    - La classe devra être bien documentée avec des schémas UML.
- 

#### Ce qu'on attend techniquement

- Des classes correctement conçues avec des attributs pertinents.
- Une utilisation de l'encapsulation avec des getters et setters.
- Une bonne démonstration de l'héritage et du polymorphisme.

- Un code propre et bien expliqué, avec des schémas UML pour bien illustrer les concepts.

### Ressources :

- <https://docs.python.org/3/>
- <https://cplusplus.com/doc/tutorial/>

### Critères de notation

Critère	Ce qu'on regarde	Points
<b>Structure</b>	Les classes et sous-classes sont bien structurées.	<b>25</b>
<b>Héritage et polymorphisme</b>	Vous avez correctement utilisé l'héritage et le polymorphisme.	<b>20</b>
<b>Méthodes</b>	La méthode move() est correctement implémentée et redéfinie.	<b>15</b>
<b>Encapsulation</b>	L'encapsulation est bien mise en œuvre avec des getters et setters.	<b>15</b>
<b>Documentation</b>	Des schémas UML et des explications claires.	<b>15</b>
<b>Qualité du code</b>	Le code est propre, lisible et respecte les bonnes pratiques.	<b>10</b>

## Test 2 : Introduction à ROS2

### Présentation du test

**Langages de programmation :** C++, Python.

Ce test introduit le **Framework ROS2**, que vous utiliserez pour créer des systèmes de contrôle robotique. Vous devrez implémenter un **package ROS2** contenant une **node publisher** qui génère des données aléatoires de capteurs (température, humidité, pression) et une **node subscriber** qui vérifie si ces données sont dans les bonnes plages prédéfinies.

Ce test est une excellente opportunité pour vous de vous familiariser avec les mécanismes de publication et de souscription dans ROS2, ainsi qu'avec l'utilisation de topics pour la communication entre nodes.

### Ce qu'il faut faire

- **Langages de programmation :** C++, Python.
  - Vous devez créer un package ROS2 appelé `sensor_data_evaluation` qui contient :
    - Un **node publisher** qui envoie des données aléatoires (température, humidité, pression).
    - Un **node subscriber** qui vérifie que ces données respectent des intervalles donnés.
  - L'objectif est de faire fonctionner l'ensemble sans erreurs.
- 

### Ce qu'on attend techniquement

- Créez un package ROS2 nommé `sensor_data_evaluation`.
- Implémentez une **node publisher** :
  - Publiez des données aléatoires toutes les 0,5 secondes sur le topic `/sensor_data`.
  - Les données doivent respecter les plages suivantes :
    - **Température** : Entre **15°C et 35°C**.
    - **Humidité** : Entre **30% et 70%**.
    - **Pression** : entre **950 hPa et 1050 hPa**.
- Implémentez une **node subscriber** :
  - Vérifiez si les données sont dans les bonnes plages et affichez un message dans le log.

- Créez un fichier de lancement pour tout exécuter.

### Ressources :

- <https://docs.ros.org/en/humble/index.html>
- <https://husarion.com/tutorials/ros2-tutorials>

### Critères de notation

Critère	Ce qu'on regarde	Points
<b>Structure du package</b>	Le package ROS2 est bien structuré et compilable sans erreurs.	<b>25</b>
<b>Node Publisher</b>	La publication des données aléatoires fonctionne bien et respecte la fréquence.	<b>25</b>
<b>Node Subscriber</b>	La vérification des données et les logs sont correctement implémentés.	<b>20</b>
<b>Fichier de lancement</b>	Le fichier de lancement permet d'exécuter le tout sans problème.	<b>15</b>
<b>Documentation</b>	Vous expliquez clairement comment tout fonctionne, avec des commentaires et des explications.	<b>10</b>
<b>Bonus</b>	Utilisation de types personnalisés ou ajout de fonctionnalités intéressantes.	<b>5</b>

## Test 3 : Création d'un Algorithme de Pathfinding

### Présentation du test

Ce test vous permettra de mettre en œuvre un système de **navigation autonome** pour un robot mobile en utilisant **ROS2** et **Gazebo**. Vous devrez implémenter un **algorithme de pathfinding** (A\*, Dijkstra, RRT, etc.) qui permettra à votre robot de trouver un chemin entre deux points tout en évitant les obstacles dans l'environnement simulé. Ce test testera votre capacité à utiliser des algorithmes classiques pour la navigation et l'intégration de ces algorithmes dans un système ROS2.

### Ce qu'il faut faire

- L'objectif est de créer un système de navigation autonome pour un robot mobile en utilisant ROS2 et Gazebo.
  - Vous devez :
    - Implémenter un algorithme de pathfinding (A\*, Dijkstra, RRT, etc.).
    - Gérer l'évitement d'obstacles.
    - Tester le tout dans Gazebo et visualiser les résultats dans RViz 2.
- 

### Ce qu'on attend techniquement

1. Un algorithme de pathfinding qui fonctionne et qui est performant.
2. Une gestion des obstacles pour que le robot évite les collisions.
3. Simuler et visualiser le tout dans Gazebo et RViz2.

### Ressources :

<https://docs.ros.org/en/humble/index.html>  
<https://husarion.com/tutorials/ros2-tutorials/ros2/>

### Critères de notation

Critère	Ce qu'on regarde	Points
Configuration	La configuration ROS2 et Gazebo est correcte.	<b>25</b>

<b>Algorithme de pathfinding</b>	L'algorithme choisi est efficace et correctement implémenté.	<b>30</b>
<b>Gestion des obstacles</b>	Le robot évite bien les obstacles sur son chemin.	<b>20</b>
<b>Simulation et visualisation</b>	La simulation dans Gazebo et la visualisation dans RViz 2 sont claires et fonctionnent bien.	<b>20</b>
<b>Documentation</b>	Vous avez bien expliqué l'algorithme, les choix techniques et les résultats obtenus.	<b>5</b>

## Information préalable sur le robot et l'environnement

### Robot utilisé : TekBot

Le robot TekBot sera utilisé pour les tests 2 et 3. Il est préconstruit avec des capteurs intégrés permettant de naviguer dans des environnements simulés. Sa configuration est prédéfinie dans le package **tekbot\_description**.

### Environnement simulé

L'environnement dans lequel le robot évoluera est contenu dans le package **maze\_solving**. Cet environnement comprend un labyrinthe adapté pour les tests de navigation et d'évitement d'obstacles. Les simulations s'exécutent principalement dans Gazebo, un simulateur robotique 3D.

### Instructions pour la configuration initiale

1. **Clonage du metapackage tekbot\_sim depuis GitHub:**

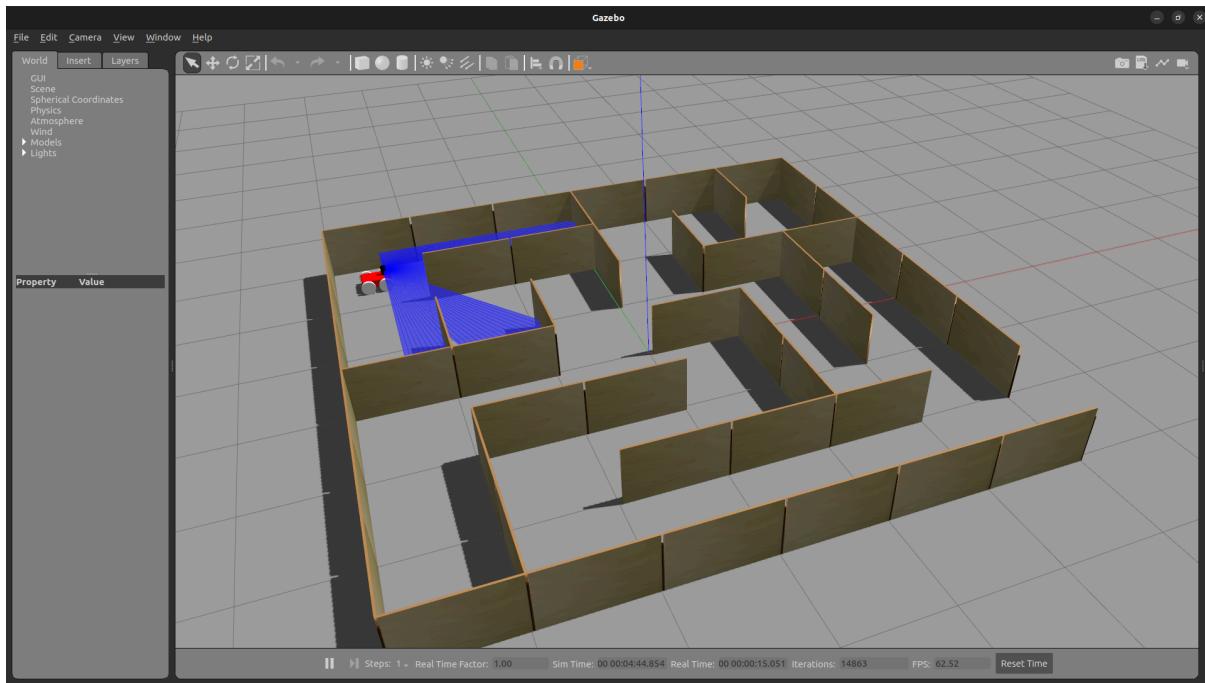
```
git clone https://github.com/charif-tekbot/tekbot_sim.git
```

2. **Compilation du package en suivant le README.md**

3. Pour lancer le robot **TekBot** dans l'environnement simulé, exédez:

```
ros2 launch maze_solving tekbot_maze.launch.py
```

4. **Visualisation dans Gazebo** Une fois l'environnement lancé, Gazebo affichera le robot et le labyrinthe. Voici un exemple de l'apparence du robot TekBot dans son environnement :



## Outils à utiliser

Les outils suivants seront nécessaires pour les tests 2 et 3 :

- **ROS 2 Humble** : Version de ROS à utiliser sous **Ubuntu 22.04**.
- **Gazebo Classic**: Simulateur 3D permettant de visualiser l'environnement et le robot.
- **RViz** : Outil de visualisation pour les données et les capteurs ROS2.
- **SLAM Toolbox** : Utilisé pour le mapping et la localisation simultanée.
- **Nav2 (optionnel)** : Framework pour la navigation autonome, intégrant le pathfinding et l'évitement d'obstacles.

## Notes et recommandations

- Avant de commencer les tests, assurez-vous que votre environnement ROS2 est correctement configuré.
- Familiarisez-vous avec les outils mentionnés ci-dessus, en particulier Gazebo et RViz.
- Consultez la documentation officielle pour ROS2 et les outils supplémentaires afin de maximiser vos chances de réussite.

Bon courage à tous les participants du TRC2K25 !



### 4.3. Tests Mécanique

#### Test 1 : Niveau Débutant

Ce test a pour but d'évaluer les compétences des participants en CAO en utilisant SolidWorks, notamment dans la maîtrise des fonctions de base telles que la création de croquis 2D de formes basiques, la modélisation 3D de pièces simples et l'assemblage de structures élémentaires.

#### Conception de pièces

Le test de niveau débutant consistera dans un premier temps en un lot de pièces à réaliser en un temps record. Chaque pièce devra être réalisée en respectant les contraintes imposées (matériaux, dimensions, ...) afin de pouvoir évaluer les participants à travers la détermination de la masse des pièces devant être dans une zone de tolérance de  $\pm 5\%$ .

Les participants devront réaliser des croquis incluant des formes géométriques de base telles que des rectangles, des cercles et des polygones. Ces exercices permettent de juger sur leur familiarité avec l'interface de conception de pièces et l'utilisation des fonctions d'extrusion et de révolution dans le but de modéliser des pièces tridimensionnelles à partir des croquis créés.

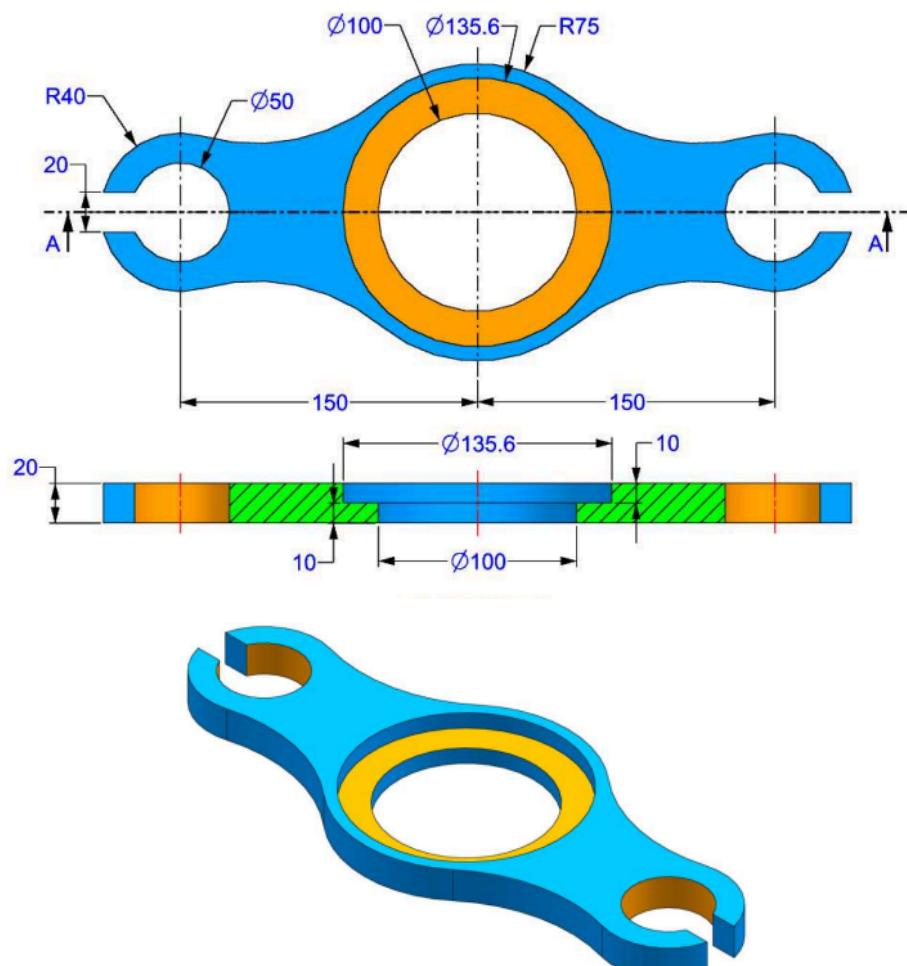
**Pièce I**

**Systèmes d'unités :** MMGS (millimètre, gramme, seconde) ;

**Décimales :** 2

**Tous les trous sont débouchants sauf indication contraire**

**Matériaux :** acier AISI 1020 ; Densité : 0,0079 g/mm<sup>3</sup>



Quelle est la masse de cette pièce(en grammes) ?

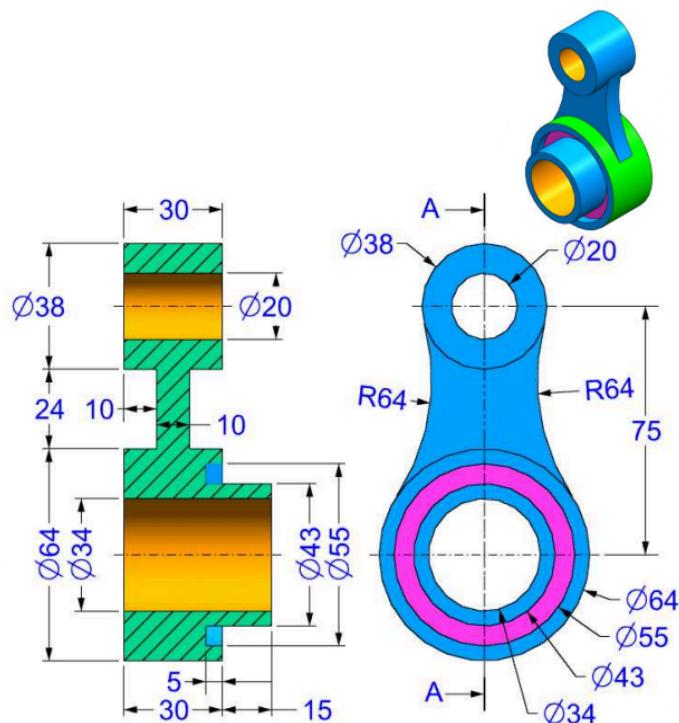
## Pièce II

**Systèmes d'unités :** MMGS (millimètre, gramme, seconde) ;

**Décimales :** 2

**Tous les trous sont débouchants sauf indication contraire**

**Matériaux :** Aluminium Alliage 1060 ; Densité : 0.0027 g/mm<sup>3</sup>



Quelle est la masse de cette pièce(en grammes) ?

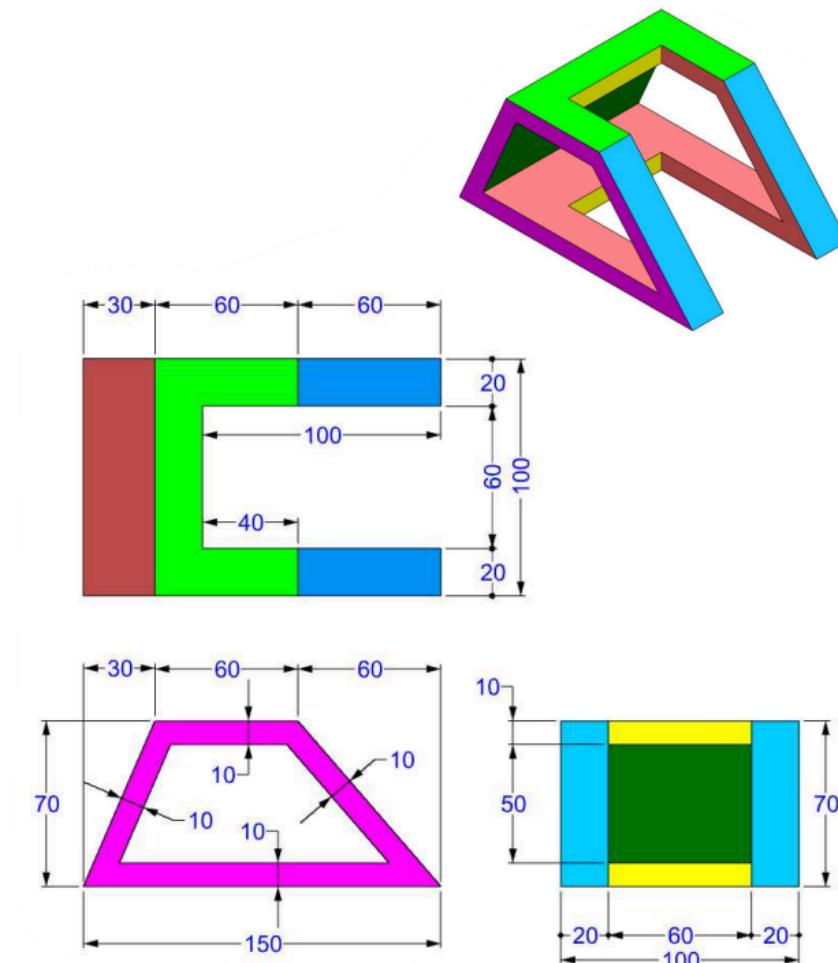
### Pièce III

**Systèmes d'unités :** MMGS (millimètre, gramme, seconde) ;

**Décimales :** 2

**Tous les trous sont débouchants sauf indication contraire**

**Matériaux :** acier AISI 1020 ; Densité : 0,0079 g/mm<sup>3</sup>



Quelle est la masse de cette pièce(en grammes) ?

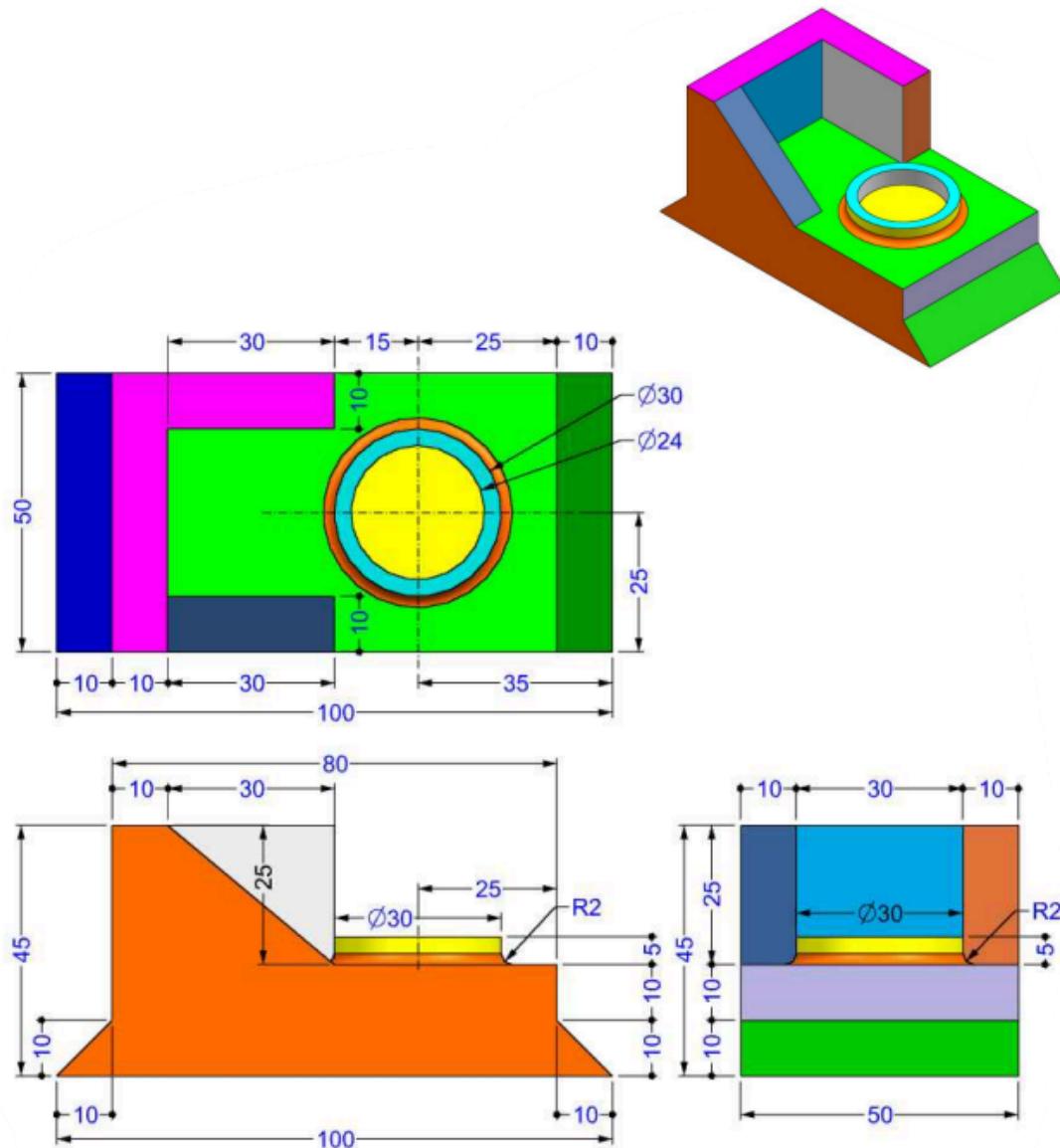
## Pièce IV

**Systèmes d'unités :** MMGS (millimètre, gramme, seconde) ;

**Décimales :** 2

**Tous les trous sont débouchants sauf indication contraire**

**Matériau :** Aluminium Alliage 1060 ; Densité : 0.0027 g/mm<sup>3</sup>



Quelle est la masse de cette pièce(en grammes) ?

## Assemblage

La deuxième session de ce test est dédiée à l'assemblage de pièces pour former un système fonctionnel. Il s'agit ici d'une pince mécanique. Les pièces à utiliser seront mises à la disposition des participants, ainsi que des plans d'assemblage, afin que tous les participants démarrent tous sur la même base pour une meilleure appréciation. L'assemblage devra être réalisé en respectant les contraintes imposées par les plans d'assemblage. Les participants seront ensuite évalués à travers la détermination des coordonnées du centre de gravité de l'assemblage final.

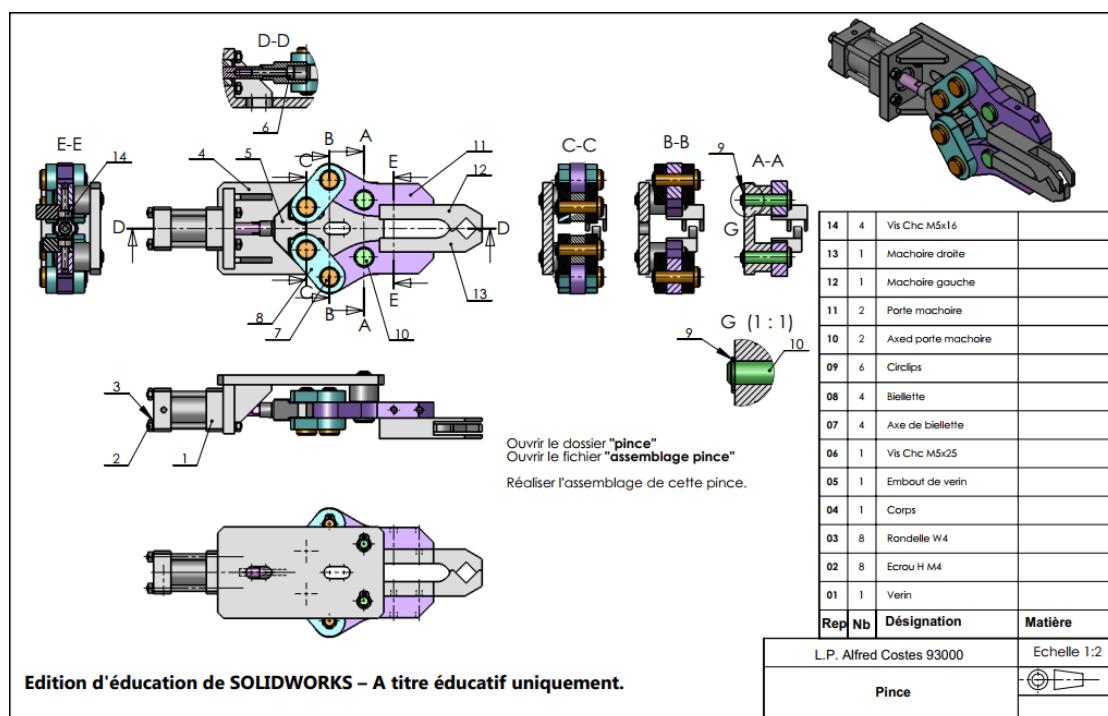
Les participants devront réaliser l'assemblage en suivant des plans d'assemblages. Cet exercice permet de juger les participants sur leur familiarité avec l'interface assemblage de Solidworks, l'utilisation des contraintes standard et l'utilisation du référentiel absolu de l'assemblage qui sont des bases pour tout assemblage.

**Systèmes d'unités :** MMGS (millimètre, gramme, seconde) ;

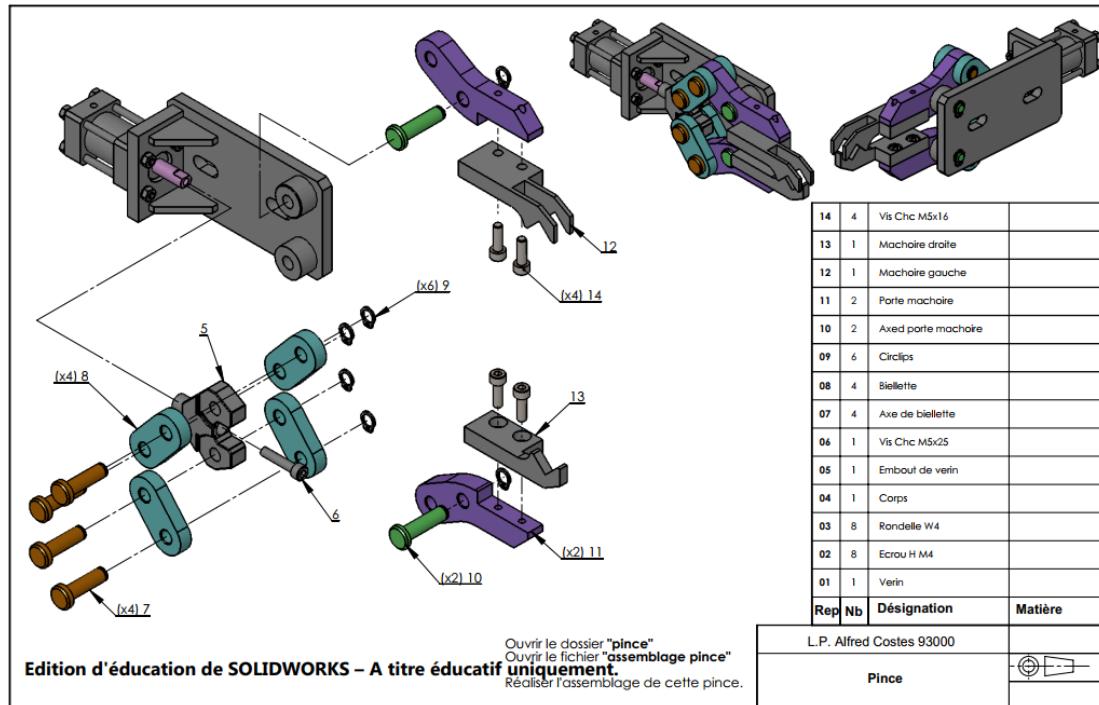
**Décimales :** 2

**Origine de l'assemblage :** Celui du fichier assemblage Pince

## Mise en plan



## Perspective



Construire cet assemblage dans SOLIDWORKS (Assemblage d'une pince mécanique)

- Télécharger le fichier zip ci-joint et ouvrez-le.
- Enregistrez les pièces contenues et ouvrez-les dans Solidworks. (Remarque : si Solidworks demande « Voulez-vous procéder à la reconnaissance des caractéristiques ? », cliquez sur « Non »)
- Ouvrir le fichier “Assemblage Pince” et réaliser l’assemblage final dans ce fichier
- Appliquer une contrainte de symétrie entre les faces des biellettes opposées l’une à l’autre
  - Après avoir fixé l’embout de vérin dans sa position minimale**, donner les coordonnées du centre de masse de cet assemblage (en millimètres) ?
  - Après avoir fixé l’embout de vérin dans sa position maximale**, donner les coordonnées du centre de masse de cet assemblage (en millimètres) ?

## Test 2 : Niveau Intermédiaire

### Conception et modification de pièces

Similaire au test de niveau débutant, le test intermédiaire est tiré d'un examen de certification Associate de Solidworks. Il permettra d'évaluer les participants sur des fonctions plus poussées que celles utilisées lors du test précédent. L'examen se déroule en trois phases où il faut en premier concevoir une pièce puis la modifier lors de deux phases suivantes. La validation du test se fera à travers l'évaluation de la masse des pièces devant être dans une tolérance de  $\pm 1\%$  de la bonne valeur.

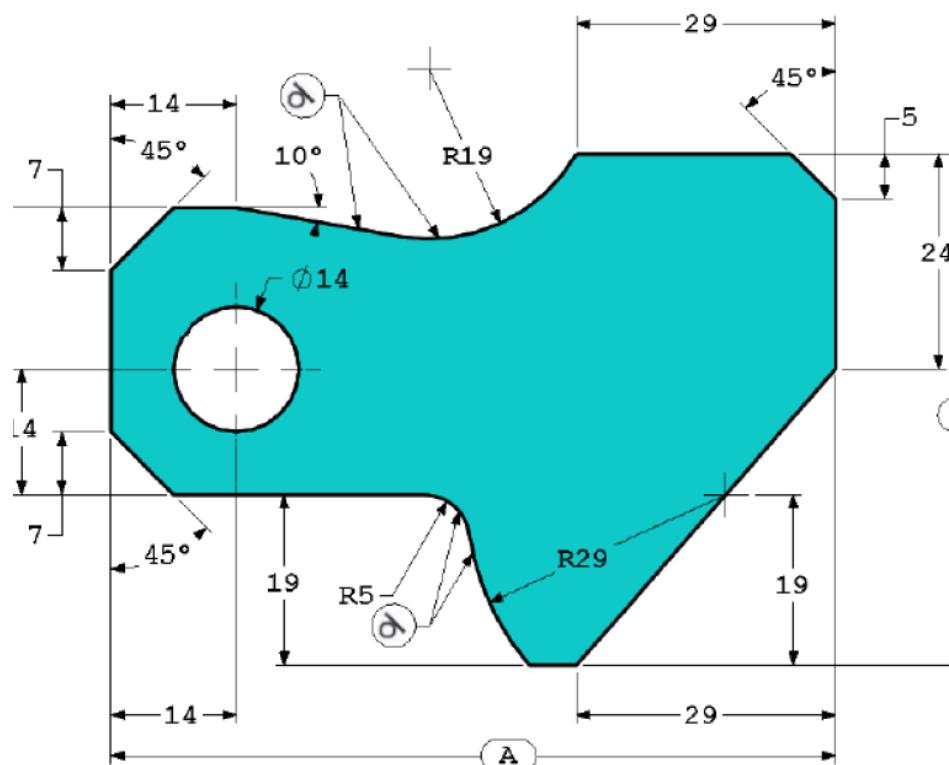
#### Partie I

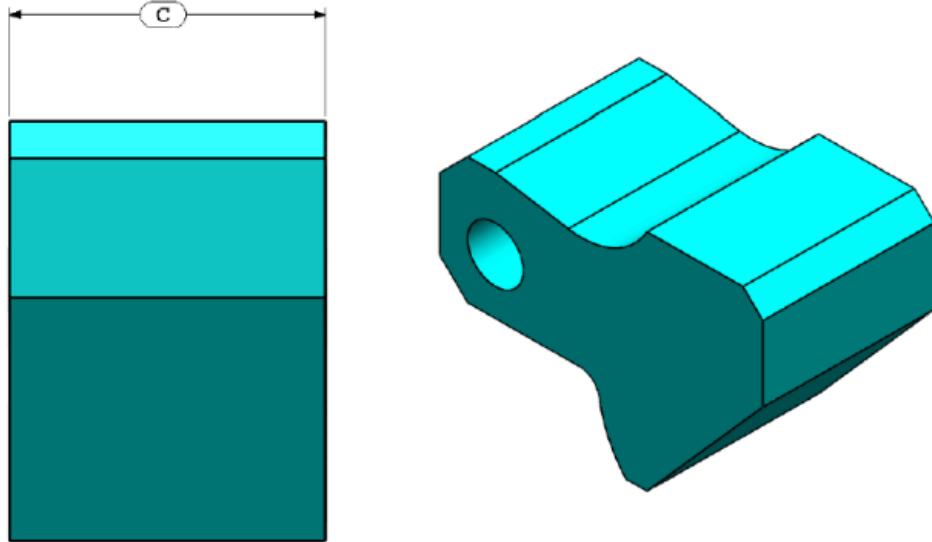
**Systèmes d'unités :** MMGS (millimètre, gramme, seconde) ;

**Décimales :** 2

**Tous les trous sont débouchants sauf indication contraire**

**Matériau :** acier AISI 1020 ; Densité : 0,0079 g/mm<sup>3</sup>



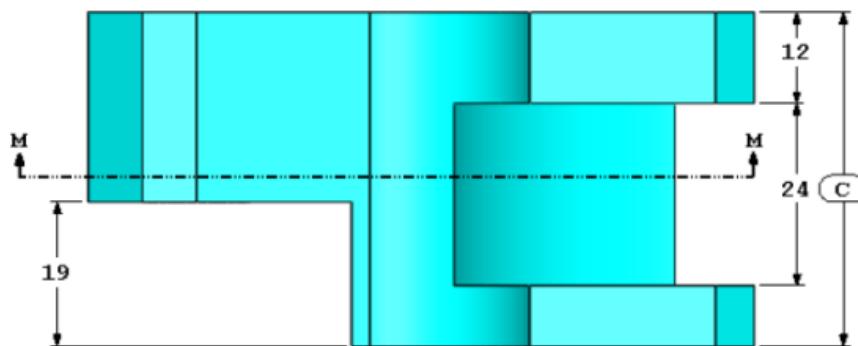
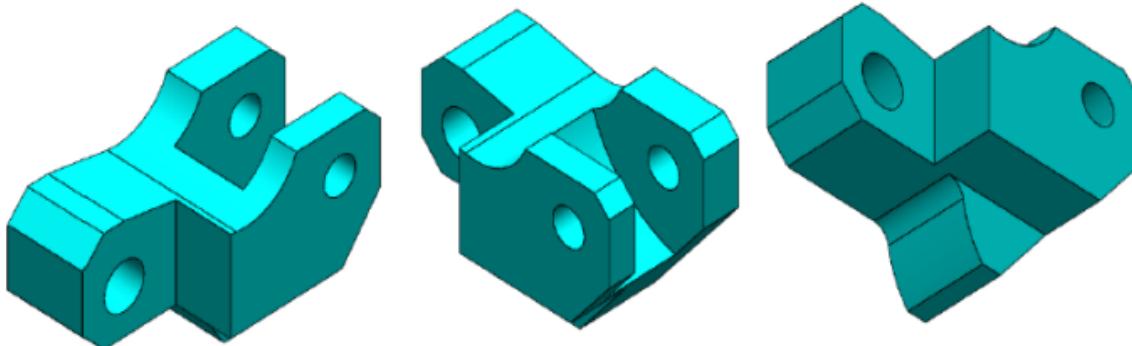


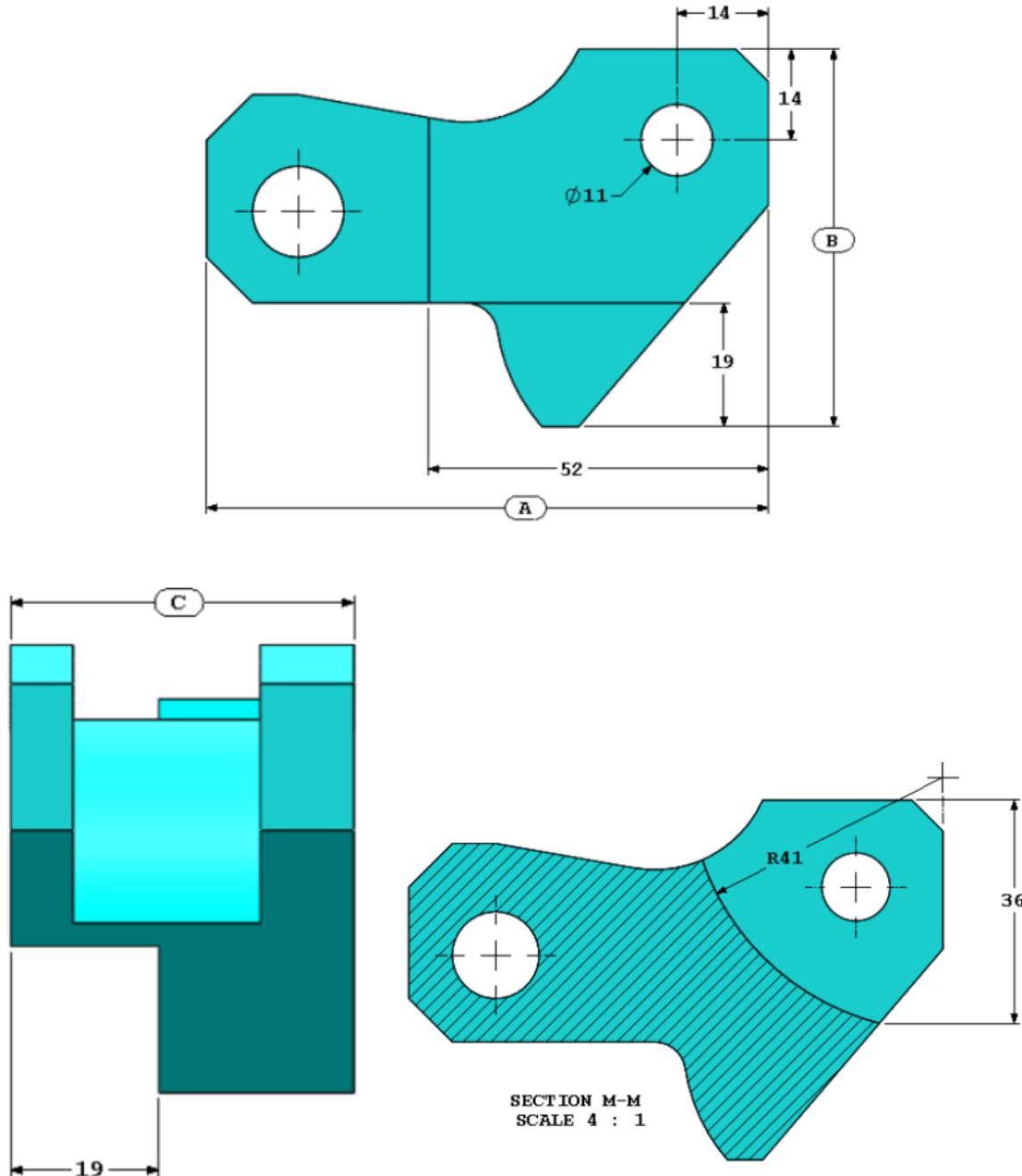
Pour chaque valeur des paramètres géométriques A, B, et C, créer la pièce puis faire ressortir la masse de la pièce.

- a) A= 81.00 ; B= 57.00 ; C= 43.00.**
- b) A= 84.00 ; B= 59.00 ; C= 45.00.**

Rappel : Enregistrez votre pièce dans un fichier différent après chaque question.

## Partie II



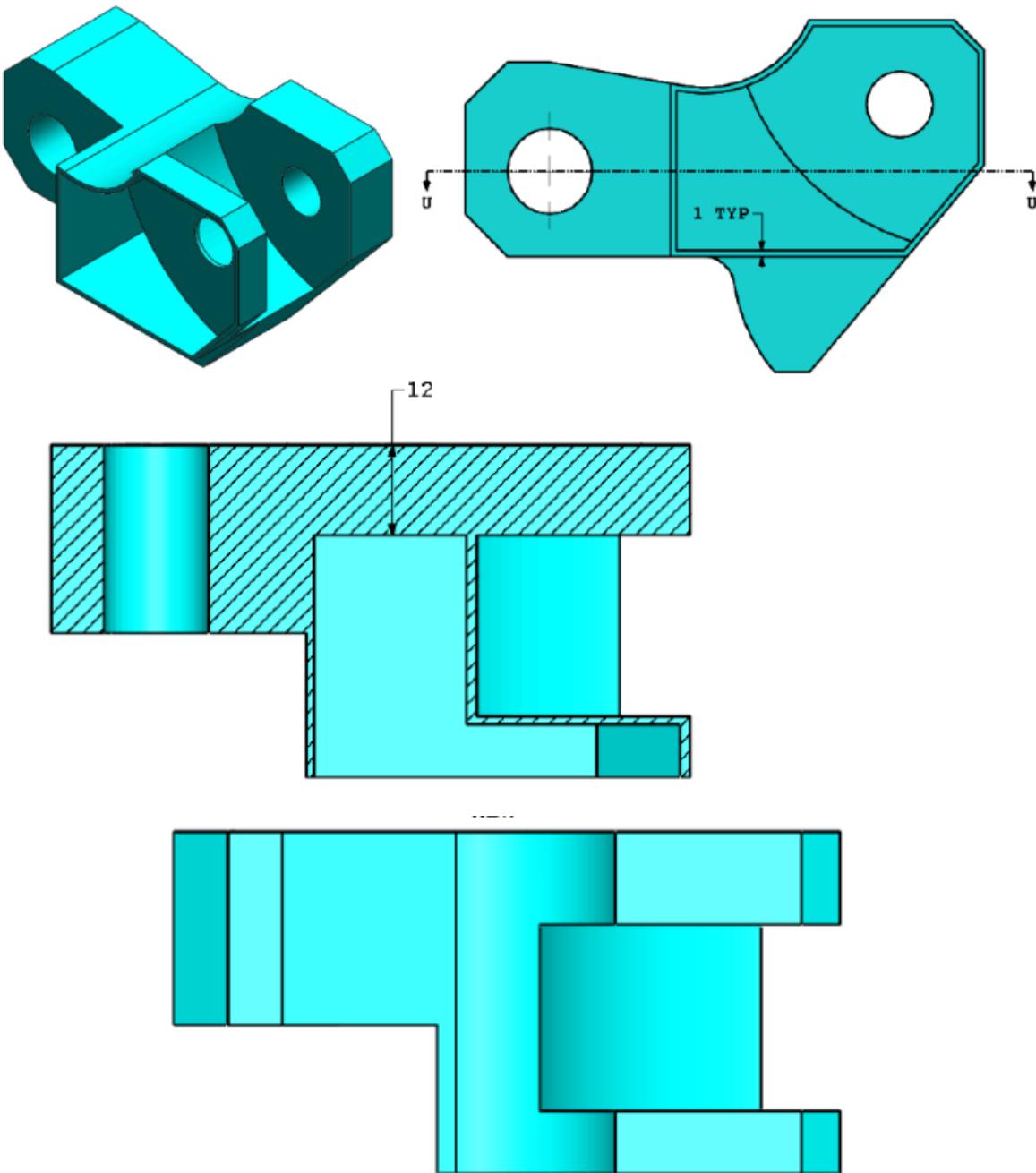


Utilisez la pièce créée dans la question précédente et modifiez-la en enlevant de la matière et en changeant les paramètres suivants :

$$A = 86.00 ; B = 58.00 ; C = 44.00.$$

Quelle est l'actuelle masse de cette pièce(en grammes) ?

### Partie III



Utilisez la pièce créée à la question précédente et modifiez-la en y ajoutant une poche.

Note 1 : Une seule poche doit être ajoutée sur un côté. Cette pièce modifiée n'est pas symétrique

Note 2 : Supposez que toutes les dimensions non spécifiées restent les mêmes que dans la partie précédente.

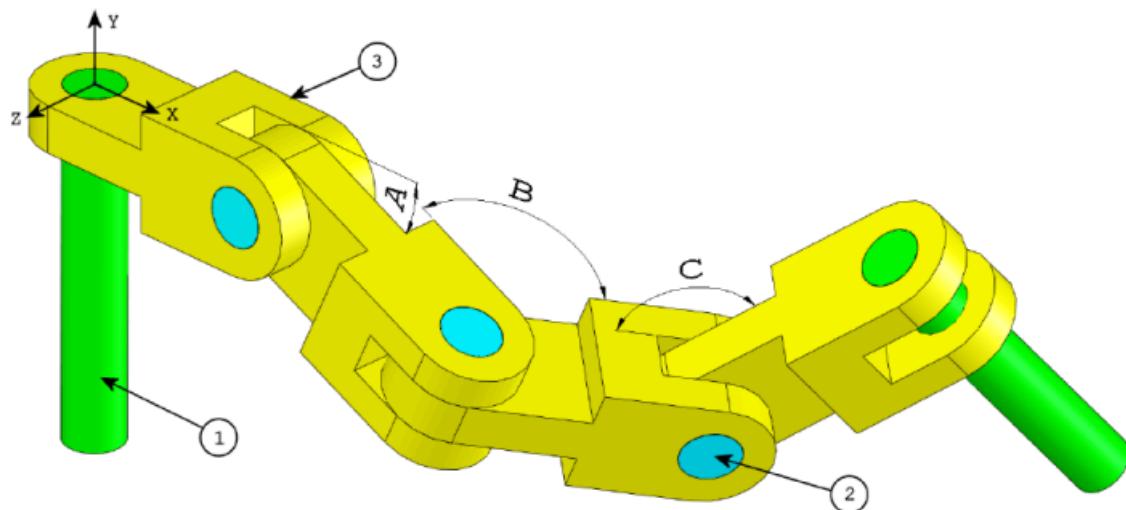
Quelle est la masse totale de la pièce (en grammes) ?

### Assemblage

**Systèmes d'unités :** MMGS (millimètre, gramme, seconde) ;

**Décimales :** 2

**Origine de l'assemblage :** Comme indiqué sur l'image



Construire cet assemblage dans SOLIDWORKS (Assemblage de maillons de chaîne)

- Télécharger le fichier zip ci-joint et ouvrez-le.
- Enregistrez les pièces contenues et ouvrez-les dans Solidworks. (Remarque : si Solidworks demande « Voulez-vous procéder à la reconnaissance des caractéristiques ? », cliquez sur « Non »)

**IMPORTANT :** Créer l'assemblage par rapport à l'origine, comme indiqué dans la vue isométrique. (Ceci est important pour calculer le centre de masse approprié).

Créer l'assemblage en respectant les conditions suivantes :

- Les goupilles sont accouplées de manière concentrique aux trous des maillons de la chaîne (pas de jeu).
- Les faces d'extrémité des axes coïncident avec les faces latérales des maillons de la chaîne (pas de jeu).

Quels sont les coordonnées du centre de masse de cet assemblage (en millimètres) pour les paramètres A, B et C suivants ?

- a) A = 25 degrees ; B = 125 degrees ; C = 130 degrees**
- b) A = 30 degrees ; B = 115 degrees ; C = 135 degrees**

### Test 3 : Niveau Avancé

Le test de niveau avancé aura pour objectif l'évaluation des participants à la conception d'une pièce complexe et d'une solution mécanique complète afin de résoudre un problème concret. Il met en avant l'utilisation des compétences acquises lors des tests précédents dans le design d'une solution mécanique à assembler. L'objectif de ce test est d'évaluer le rapport des participants à la complexité et leur créativité mais aussi leurs capacités à concevoir une solution fonctionnelle en gérant correctement les erreurs.

#### Conception de pièces

Dans cette session, il s'agit de modéliser une pièce complexe qui sera valide à l'obtention de la masse appropriée.

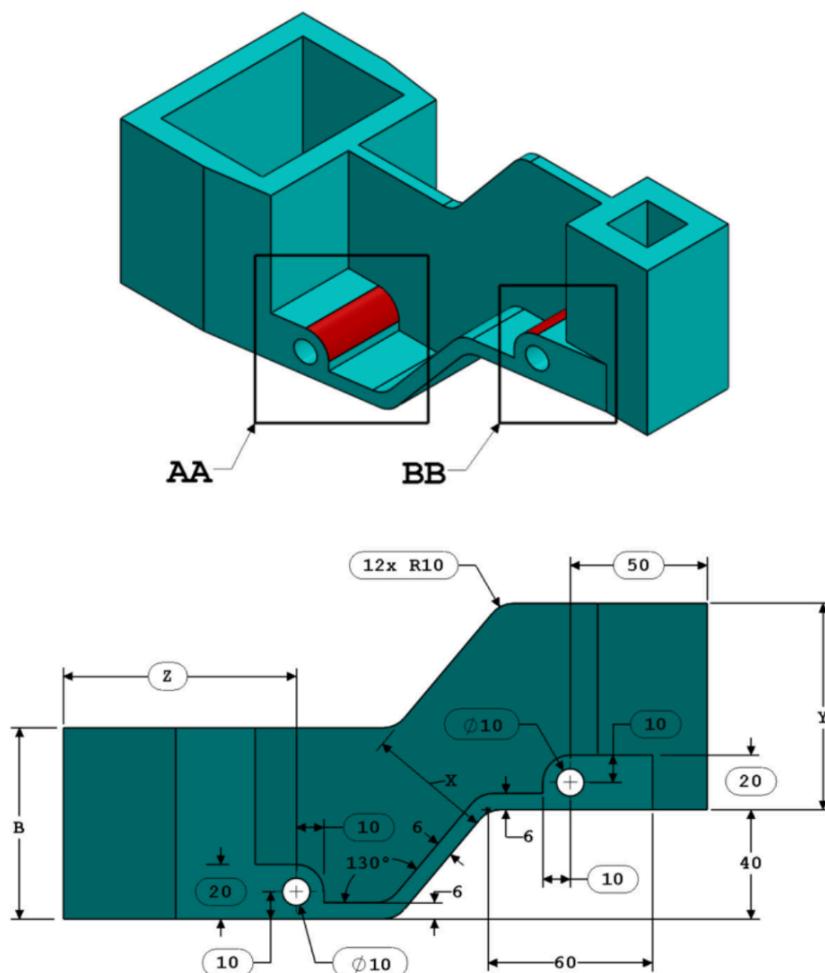
**Systèmes d'unités :** MMGS (millimètre, gramme, seconde) ;

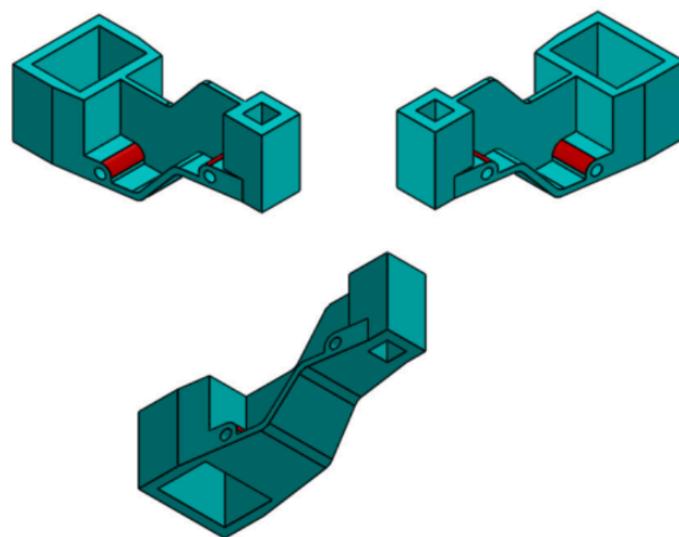
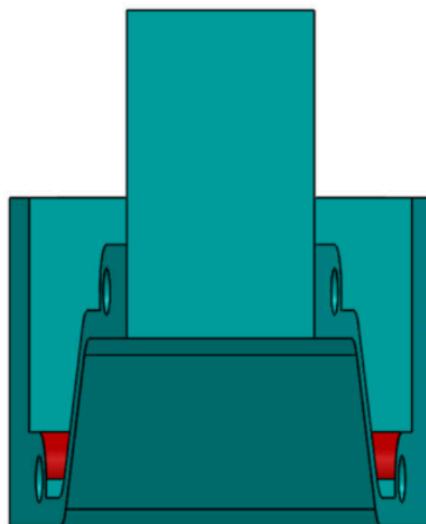
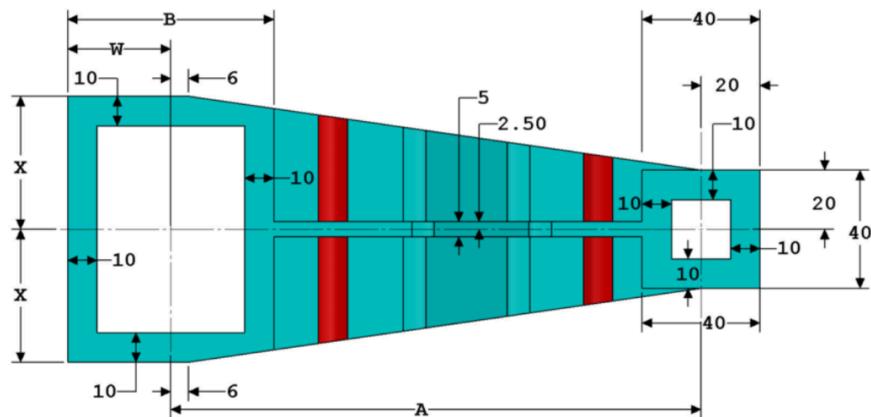
**Décimales :** 2

**Tous les trous sont débouchants sauf indication contraire.**

**Matériau :** 1060 Alloy Aluminium ; Densité : 2700 Kg/m<sup>3</sup>.

**Rayon des 12 filets = 10 mm sauf indication contraire.**





Quelle est la masse de cette pièce pour les dimensions suivantes (en grammes) ?

- a)  $A = 193 \text{ mm}$  ;  $B = 88 \text{ mm}$  ;  $W = B/2 \text{ mm}$  ;  $X = A/4 \text{ mm}$  ;  $Y = B+5.5 \text{ mm}$  ;  $Z = B+15 \text{ mm}$  ;

- b)  $A = 205 \text{ mm}$  ;  $B = 100\text{mm}$  ;  $W = B/2 \text{ mm}$  ;  $X = A/4 \text{ mm}$  ;  $Y = B+5.5 \text{ mm}$  ;  $Z = B+15 \text{ mm}$  ;
- c)  $A = 210 \text{ mm}$  ;  $B = 105 \text{ mm}$  ;  $W = B/2 \text{ mm}$  ;  $X = A/4 \text{ mm}$  ;  $Y = B+5.5 \text{ mm}$  ;  $Z = B+15 \text{ mm}$  ;

## 4.4. Test Final

### Réalisation d'un système de convoyeur

Team : Electronique, IT et Mécanique

Durée : 2 semaines

#### Présentation du test

Une entreprise de recyclage de déchets voudrait s'installer dans la zone industrielle de TEKBOT CITY avec un tout nouveau système de convoyeur pour faciliter le tri des déchets. Vous êtes appelé à user de beaucoup de **créativité** en concevant et en réalisant ce nouveau système de convoyeur. Le système de convoyeur auras à **trier 04 types** de déchets qui seront représentés par des objets (des cubes par exemple) **de couleurs différentes (le vert, le jaune, le rouge, et le bleu)**. L'administrateur du convoyeur doit pouvoir suivre en **temps réel sur une interface web les quantités de déchets triés par type de déchets**.

#### Description technique

Pour ce projet final, vous devrez réaliser un convoyeur mécanique associé à un dispositif intelligent de tri des déchets. Son fonctionnement est le suivant : la bande du convoyeur reste immobile et ne se meut que lorsque le système détecte des déchets. Ensuite, les déchets passent par la zone de détection avant d'arriver à la fin du convoyeur ou ils seront collectés manuellement par les membres de l'équipe. Vous devez placer les déchets dans la berne qui leur sera indiquée par le système de tri.

Pour permettre un suivi en temps réel du tri des déchets, une **interface web** sera mise en place. Celle-ci affichera les **quantités de déchets triés** par type (vert, jaune, rouge, bleu). L'interface devra être intuitive et accessible aux administrateurs, leur permettant de **visualiser l'état du tri en temps réel**.

L'**interface web** devra inclure les logos de TEKBOT et de la TRC 2025.

Vous devrez respecter certaines contraintes du point de vue électronique. En effet :

- Les objets représentant les déchets doivent être manuellement collectés dans des récipients spécifiques à chaque type de déchet à la sortie du convoyeur après le tri. Vous devrez placer les déchets dans les bernes que votre algorithme vous aura indiqué.
- Dans votre rendu final, vous devrez utiliser un microcontrôleur ATmega328P comme dans les tests précédents ou cette fois-ci une carte Arduino nano.
- La source d'alimentation devra être un bloc de batteries Lithium.
- Un capteur de couleur sera utilisé pour le tri des déchets.
- L'utilisation du module laser KY-008 avec son récepteur ou une photorésistance est fortement recommandée pour détecter la présence d'un déchet sur le convoyeur. Vous êtes tout à fait libres de choisir le système de détection qui vous convient.

- Documenter votre travail sur le dépôt GitHub qui vous sera attribué.

Le design du convoyeur devra respecter des spécifications en termes de dimensionnement.

- Longueur totale du convoyeur : 650 mm (Les autres dimensions sont libres de choix)
- Hauteur du tapis par rapport au sol: 100 mm
- Caractéristique d'un déchet : Cube de 30 mm avec XX g
- Logiciel utilisé : SolidWorks

### **Exigence de documentation**

Afin d'assurer la traçabilité et la compréhension complète du projet, tout le processus de conception mécanique doit être rigoureusement documenté. Cela inclut, mais ne se limite pas à ce qui suit :

- Études préliminaires : Analyse des besoins fonctionnels et des contraintes mécaniques
- Choix et justification : Sélection des matériaux et des composants en fonction des spécifications techniques
- Modélisation CAO : Présentation des modèles 3D de chaque composant et assemblage pour valider l'intégrité mécanique et l'agencement

**NB:**

Vous êtes libres d'utiliser la technologie de votre choix pour la réception des données sur l'interface web.

Dans le cadre du challenge final de la compétition, le système de convoyeur que vous aurez à concevoir et réaliser dans ce test vous sera d'une grande utilité. Vous devrez en effet le modifier pour l'utiliser lors du challenge final. Tâchez donc de concevoir le mieux possible pour pouvoir en bénéficier plus tard.

Dans ce challenge final, chaque sous-domaine (**Électronique, IT, Mécanique**) sera évalué séparément. La note finale sera calculée en faisant la moyenne des notes obtenues dans ces trois domaines. **note finale = (Note Élec + Note IT + Note Méca) / 3**

### **Grille de notation électronique**

Nous partageons avec vous la notation du test.

Le test sera noté sur 100 points répartis comme suit :

1. La circuiterie (**40 points**) :

- la qualité du schéma électronique : le choix des composants et la pertinence de leurs rôles,
- l'optimisation et l'esthétique du PCB,

- La conformité entre le rendu 3D et la réalisation physique du PCB
- la finesse des soudures et l'esthétique de la carte électronique,
- la gestion et la sécurité de l'alimentation. La sécurité de l'alimentation prend en compte les protections contre les courts circuits et le respect des niveaux de tension et de courant pour chaque composant,
- la gestion des câbles.

NB: La présence d'une carte Arduino autre que la nano ou de breadboard dans votre circuit final vous donne un malus de 18 points. Votre circuit sera donc noté sur 12 points au lieu de 30.

2. Le code (**15 points**):

- la facilité de compréhension du code au travers des commentaires,
- l'optimisation du code, c'est-à-dire la méthodologie exploitée pour écrire le code,
- la lisibilité du code (respecter les indentations).

3. Le test du fonctionnement (**30 points**).

4. La structure de la documentation et sa facilité de compréhension (**10 points**).

5. La présentation (**5 points**)

- la qualité des slides
- la maîtrise du sujet
- l'aptitude à répondre aux questions.

### Grille de notation IT

Dans le cadre du développement du système de convoyeur intelligent à TEKBOT CITY, cette grille permet d'évaluer toutes les composantes liées à l'informatique et aux systèmes embarqués, de la détection à l'affichage en temps réel des données. Les 100 points à attribuer seront répartis comme suit:

1. Détection intelligente (**20 points**)

- Capteur de couleur intégré et capable d'identifier les 4 types de déchets (Rouge, Bleu, Vert, Jaune)
- Détection de présence fiable (laser KY-008, photorésistance ou équivalent)

2. Automatisation du convoyeur (**15 points**)

- Activation du tapis uniquement lorsqu'un déchet est détecté
- Algorithme de gestion du tri (déttection, identification, consigne de tri)

### 3. Interface Web – Suivi en temps réel (**15 points**)

- Interface affichant les quantités de déchets triés, différenciés par couleur
- Interface ergonomique, claire et intuitive
- Présence des logos TEKBOT et TRC 2025

### 4. Traitement des données (**10 points**)

- Comptage précis et différencié par type de déchet
- Actualisation des données en temps réel ou quasi-instantané

### 5. Intégration système (**10 points**)

- Communication efficace entre le microcontrôleur (ATmega328P / Arduino Nano) et l'interface web

### 6. Robustesse logicielle (**15 points**)

- Code bien structuré, lisible, modulaire, avec gestion d'erreurs
- Utilisation appropriée de bibliothèques pour l'embarqué et le web

### 7. Documentation & GitHub (**5 points**)

- Documentation claire sur GitHub (README, schémas, explications techniques)
- Suivi de développement avec des commits réguliers et bien commentés

### 8. Tests et validation (**5 points**)

- Tests réalisés sur tous les types de déchets et scénarios possibles

### 9. Créativité / Bonus IT (**5 points**)

- Fonctionnalités avancées ou originales (IA, commande vocale, version mobile...)
- Architecture élégante, sécurité, maintenabilité, commentaires clairs

## Grille de notation mécanique

La grille de notation établit des critères clairs et objectifs pour évaluer la qualité des modélisations réalisées par les compétiteurs lors du **TEKBOT Robotics Challenge (TRC)**. Ces critères permettent de juger à la fois la précision technique et la pertinence des solutions proposées face aux problématiques imposées. Le test en entier sera noté dans un premier temps sur 300 mais ramené à 100 points dans un souci d'uniformité des notes avec les challenges des autres disciplines.

Ainsi, les **challenges débutant et intermédiaire** seront notés sur **70 points chacun** en prenant en compte dans les grandes lignes:

- l'application des contraintes géométriques adéquates
- le respect de toutes les dimensions
- la capacité à trouver les bonnes valeurs de masses
- l'utilisation de toutes les pièces fournies pour les assemblages
- l'utilisation de toutes les contraintes d'assemblages nécessaires
- l'obtention des bonnes valeurs de coordonnées du centre de masse

Le **challenge avancé** quant à lui, sera noté sur **110 points**. La conception de pièces et l'assemblage seront notés sur 100 points et il y aura un bonus de 10 points pour les équipes qui en plus du challenge seront capables de réaliser une simulation pertinente, recueillir des informations utiles, les analyser et s'en servir pour l'amélioration de leur convoyeur. Les grands points du challenge sont les suivants:

5. l'application des contraintes géométriques adéquates
6. le respect de toutes les dimensions
7. la capacité à trouver les bonnes valeurs de masses
8. la capacité à concevoir un convoyeur totalement fonctionnel et capable de déplacer de petits objets
9. Respecter les spécifications techniques de conception
10. Faisabilité des modèles pour l'impression 3D et l'assemblage

Enfin, une **évaluation globale** du travail de chaque équipe sera attributrice de **50 points** décernés à la présentation de chaque équipe ainsi que la collaboration et la répartition des tâches afin de s'assurer de l'implication de tous les membres de l'équipe dans la réalisation du challenge. La documentation du travail étant très importante, toute équipe n'ayant pas réalisé une documentation complète et détaillée de son travail se verra retirer 30 points sur sa note finale.