

GUIDE : ARROSAGE AUTOMATIQUE D'UNE PLANTE AVEC LE NED 2



NIRYO
HUMAN - MOTION - ROBOT

Sommaire

| | |
|--------------------------|--------------|
| Sommaire | Page 1 |
| Introduction | Page 2 |
| Réalisation | Page 3 à 10 |
| Objectif | Page 3 |
| Matériel | Page 4 |
| Ressources informatiques | Page 5 |
| Circuit électronique | Page 6 |
| Programmation | Page 7-8 |
| Blockly | Page 9-10 |
| 3D | Page 11 à 20 |
| Conception | Page 11 à 15 |
| Impressions | Page 16 à 20 |
| Finitions | Page 21 |
| Pistes d'améliorations | Page 22 |



Introduction



Bienvenue dans ce guide pas à pas passionnant qui vous emmènera à travers l'univers de la **robotique** et de l'**automatisation domestique**.



Dans ce tutoriel, vous découvrirez un projet captivant : l'automatisation de l'arrosage de plantes en utilisant le **Ned 2**, un bras robotique collaboratif polyvalent avec, en parallèle, une carte Arduino et des capteurs d'humidité.



Imaginez la scène : votre plante de bureau baignée de soleil, verdoyante et en pleine santé, grâce à un système d'**arrosage intelligent** qui prend soin d'elle même lorsque vous êtes occupé ou absent. C'est précisément ce que nous allons réaliser ensemble au cours de ce guide.



Réalisation*

Objectif

Notre objectif est simple : créer un système qui mesure l'humidité du sol dans le pot de la plante à l'aide de capteurs, et lorsque la plante a soif, le Ned 2 intervient pour lui apporter l'hydratation nécessaire.

Nous allons donc accomplir cela en suivant ces étapes :

- Création d'un circuit électronique avec Arduino et un capteur d'humidité pour surveiller le niveau d'humidité du sol (simulation + réel).
- Création du code de l'Arduino qui nous permettra de recueillir des données et envoyer des ordres au Ned 2 en conséquence.
- Création d'un programme Blockly dans Niryo Studio (logiciel de la marque) pour permettre au Ned 2 d'arroser la plante en fonction des données reçues de l'Arduino.
- Conception et impression 3D d'un arrosoir adapté au Ned 2

Matériel



Le Ned 2 : Il est le protagoniste de notre projet, ce bras robotique nous permettra de manipuler l'arrosoir et d'arroser la plante en temps voulu.

Arduino Uno : Une carte microcontrôleur

Arduino, telle qu'une Arduino Uno, Nano, ou similaire, sera le cerveau de notre système. Elle sera chargée de collecter les données du capteur et de coordonner l'arrosage avec le Ned 2.



Capteur d'humidité : Nous aurons besoin d'un capteur d'humidité du sol compatible Arduino pour mesurer le niveau d'humidité du substrat de la plante. Il en existe plusieurs modèles. L'un des deux présents ci-contre fonctionnera très bien pour notre projet.

Éléments d'électronique de base :

N'oubliez pas les composants d'électronique de base, tels que des câbles Dupont pour les connexions, une petite LED pour les tests, des résistances si nécessaires, et tout autre composant électronique nécessaire pour le circuit.



Ressources informatiques

Avant de plonger dans la création de notre système d'arrosage automatisé, il est important de s'assurer que vous disposez des logiciels nécessaires et que vous avez accès à des ressources en ligne essentielles pour ce projet.



Niryo Studio :

Niryo Studio est l'environnement de programmation pour le Ned 2 de Niryo. Vous pouvez le télécharger et l'installer en cliquant [ici](#).

Arduino IDE :

L'IDE Arduino est nécessaire pour programmer votre carte Arduino. Vous pouvez le télécharger à partir du site officiel d'Arduino :

www.arduino.cc



Tinkercad :

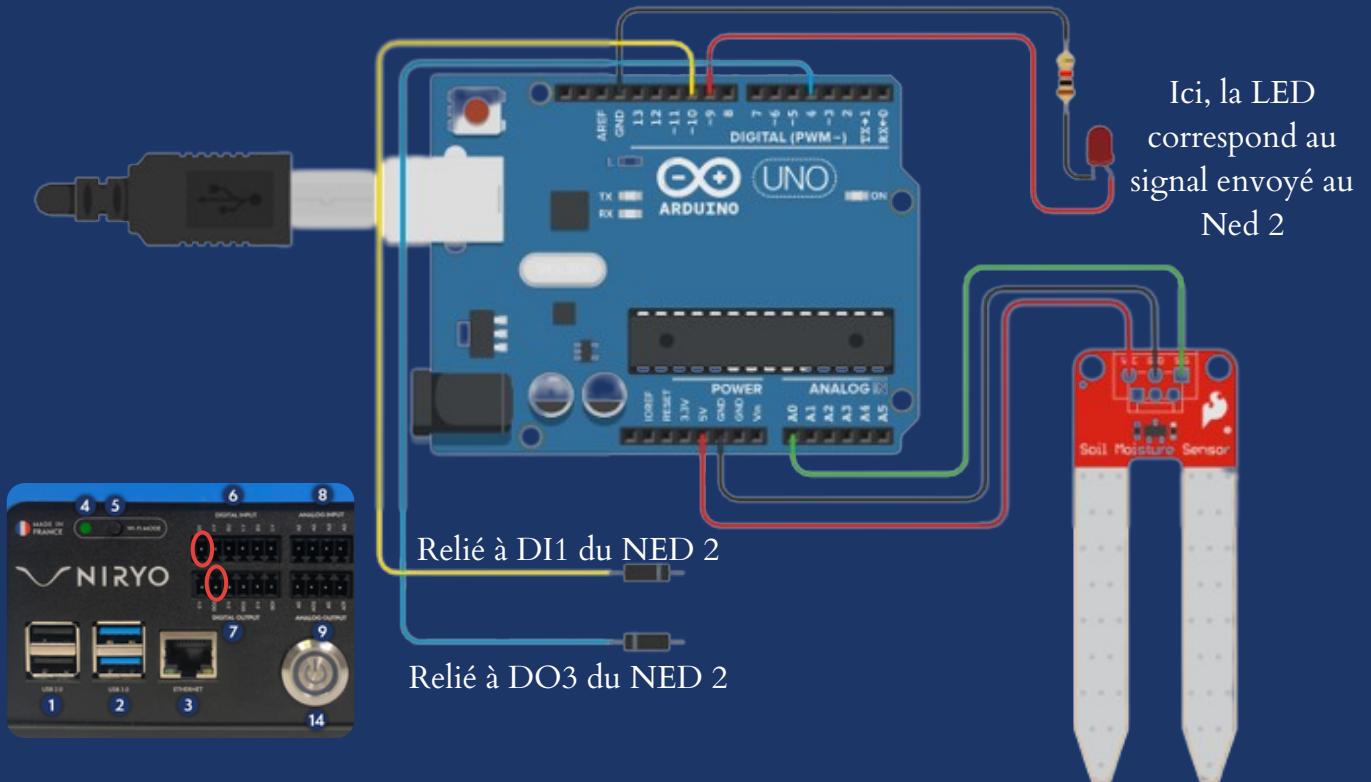
Tinkercad est un outil en ligne puissant qui vous permet de créer et de simuler des circuits électroniques. Vous pouvez y accéder gratuitement en vous rendant sur leur site web :

<https://www.tinkercad.com>

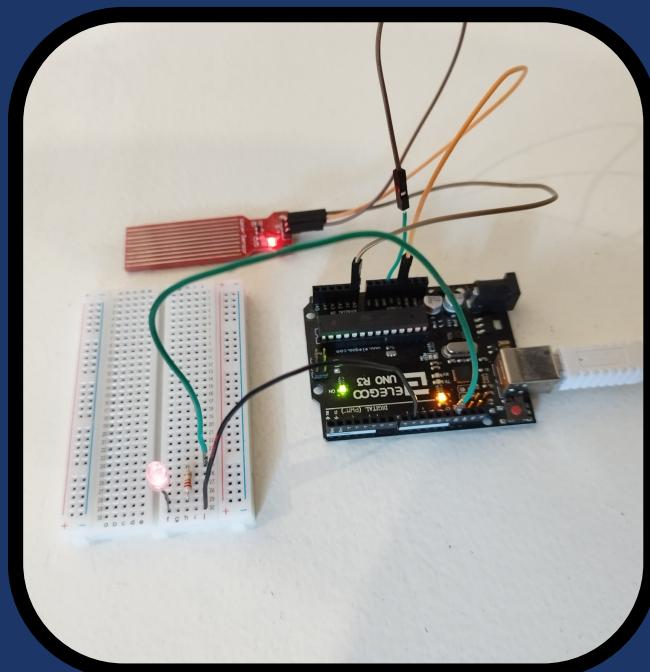
Circuit électronique

Afin de faire fonctionner notre projet, nous devons créer avant tout le circuit qui relie les différents composants entre-eux.

Voici le schéma de simulation : (cliquez sur l'image pour y accéder)



Et voici une photo du circuit réel :



Programmation

Pour faire fonctionner tout le système électronique vu ci-dessus et le faire communiquer avec le Ned 2, il faut donc écrire le code qui sera téléversé dans la carte Arduino Uno grâce au logiciel **Arduino IDE**.

Voici le code :

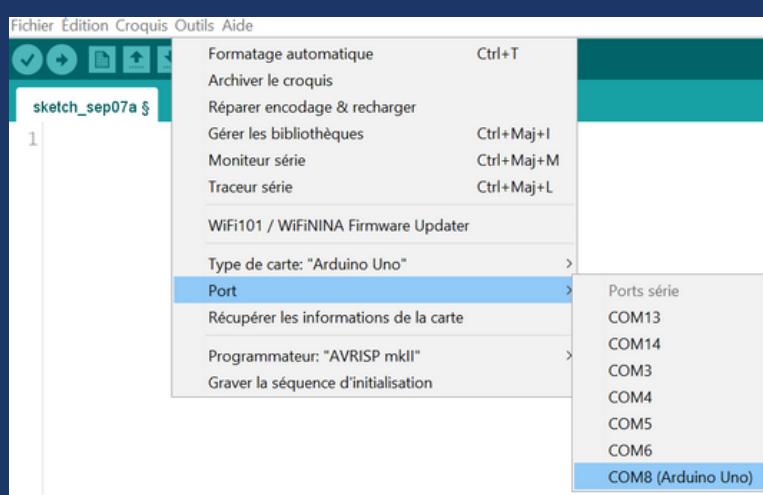
```
 1  /*----- Guide arrosage automatique - Niryo -----*/
 2
 3 #define PIN_CAPTEUR A0      // Broche de l'Arduino à laquelle la broche du capteur de niveau d'eau est reliée
 4 #define LED 9               // Broche de l'Arduino à laquelle la LED est reliée.
 5 #define COMMANDE 10         // Broche envoyant le même signal que la LED sauf que celle-ci est destiné au Ned 2
 6 #define RETOUR 4            // Broche qui va recevoir la confirmation du Ned 2 une fois que la plante aura été arrosée
 7
 8 int value = 0;           // On initialise la variable à 0 avant de commencer
 9 int ret = 0;              // Idem
10
11
12 void setup() {
13   pinMode(LED, OUTPUT);    // On définit la LED comme une sortie car c'est l'Arduino qui
14   // va lui envoyer des informations, pas l'inverse.
15
16   pinMode(COMMANDE, OUTPUT); // Idem (ici la broche COMMANDE enverra les mêmes informations que dans la LED mais pour le Ned 2)
17
18   pinMode(RETOUR, INPUT);   // Ici, c'est le contraire, l'Arduino va recevoir des informations du Ned 2
19
20   Serial.begin(9600);       // Ici on définit simplement le paramétrage du moniteur série
21 }                         // (Attention : bien vérifier dans le moniteur série que le "baudrate" est mis sur 9600)
22
23 void loop() {
24
25   value = analogRead(PIN_CAPTEUR); // Ici on demande à l'Arduino de lire le niveau d'eau et de stocker la valeur dans la variable
26   Serial.print("Valeur du capteur: "); // On fait un retour dans le moniteur série afin de connaître la valeur du niveau
27   Serial.println(value);           // d'eau dans le pot
28
29   // Ici on va s'occuper de l'envoi conditionnel d'informations au Ned 2 en fonction du niveau d'eau mesuré
30
31   ret = digitalRead(RETOUR);
32   Serial.print("Valeur du retour: ");
33   Serial.println(ret);
34
35   if(value <= 310){           // Si la valeur du niveau d'eau est inférieure à 310 (valeur arbitraire) alors on envoie
36     digitalWrite(LED, HIGH);  // l'info à la LED et au Ned 2 pour arroser la plante (et on l'affiche dans le moniteur série)
37     digitalWrite(COMMANDE, HIGH);
38     Serial.println("J'ai soif\n"); // (le "\n" permet de faire un retour à la ligne dans le moniteur série
39
40
41   if(ret == 1){             // Si le robot à fini d'arroser la plante alors on dit à l'Arduino que la plante
42     Serial.println("J'ai été arrosé !"); // a suffisamment d'eau et on retourne dans notre boucle du début.
43     value = 320;              // De plus, on réinitialise les variables à des valeurs correspondant à notre
44     ret = 0;                  // situation initiale mais en prenant en compte que la plante a été arrosée
45     delay(3000);             // Ici on met un petit peu de délai pour que la terre du pot s'imprègne entièrement de l'eau versée
46   }
47
48 }
49
50 else{
51   digitalWrite(LED, LOW);    // Sinon, si le niveau d'eau est correct, on n'envoie pas d'ordre au Ned2 et on éteint la LED
52   digitalWrite(COMMANDE, LOW);
53   Serial.println("J'ai bien bu !\n");
54 }
55
56
57 delay(500);                // On impose un délai d'une demi-seconde avant chaque nouvelle prise de mesure du niveau d'eau
58 }
59
60 /*----- Réalisé par Mathéo Merlin -----*/
61
```

Le code est disponible en cliquant [ici](#).

Afin de téléverser ce code dans l'Arduino Uno, voici les étapes à suivre :



- 1) Lancez le logiciel Arduino IDE
- 2) Branchez votre Arduino Uno à votre ordinateur à l'aide du câble USB adapté.
- 3) Sélectionnez le bon port de l'Arduino



Si vous ne trouvez pas le bon port, allez dans votre PC :
Gestionnaire de périphériques > Ports (COM et LPT)
De là, vous trouverez le numéro du port COM de votre Arduino :



- 4) Dans un nouveau projet, copiez et collez le code fourni plus haut
- 5) Pour finir, téléversez et attendez que le processus soit terminé :



Téléversement terminé

Et voilà, votre code est désormais présent dans votre carte Arduino !

Blockly

La programmation en Blockly est simplifiée comparé à du code brut. Cet outil est souvent destiné aux débutants, mais il peut également servir à des personnes plus confirmées !

Dans notre projet, le Blockly va nous permettre de faire se déplacer le bras robotique étape par étape afin d'arroser la plante. Cependant, il nous faudra également faire très attention à la réception et à l'envoi d'informations avec l'Arduino comme vu précédemment. C'est cette liaison qui va faire que le Ned 2 arrosera ou non la plante en fonction du taux d'humidité mesuré dans la terre du pot de fleur.

Voici le Blockly : (cliquez sur la photo pour accéder au fichier XML)



Pour expliquer en quelques mots comment à été construit ce projet en Blockly, il s'agit simplement d'une boucle qui se répète indéfiniment, voilà à quoi sert le bloc vert “While True, Do ...”

De plus, il faut comprendre que pour que le bras robotique se mette en mouvement et arrose la plante correctement, le Ned 2 doit avant tout attendre de recevoir l'instruction de la part de l'Arduino avant de commencer n'importe quel déplacement ou action quelconque.

Dès le début du “While True” nous demandons au Ned 2 de remettre à l'état bas ("0" en binaire) la sortie DO3 (Digital Output 3), quoi qu'il ait pu arriver auparavant.

Ensuite, on vérifie si une instruction à été recue de la part de l'Arduino sur la broche DI1 (Digital Input 1) du Ned 2, et si c'est le cas, on ordonne au bras d'exécuter une suite de mouvement permettant en finalité d'arroser la plante.



Une fois cela terminé, le Ned 2 va envoyer une information de confirmation à la carte Arduino via la broche DO3 du bras en la mettant à l'état haut ("1" en binaire). Ensuite, nous patientons 3 secondes afin que la terre du pot de fleur se gorge de l'eau fraîchement versée.

Pour finir, nous remettons la sortie DO3 à l'état bas ("0" en binaire) afin de pouvoir recommencer un second cycle d'arrosage sans encombre si cela est nécessaire.

3D

Conception

Notre but principal pour ce projet était également de modéliser et imprimer un arrosoir qui pourrait contenir de l'eau et qui pourrait être attrapé par la pince du Ned 2 mais surtout qui pourrait nous permettre d'arroser la plante grâce au bras robotique. Voici le résultat que nous allons obtenir :



Les modélisations sont disponibles [ici](#).

Nous modéliserons notre arrosoir sur le logiciel **Fusion 360** mais il est également possible de le faire sur d'autres logiciels similaires comme **SolidWorks** par exemple.

Pour vous expliquer brièvement comment marche généralement un logiciel de modélisation 3D comme Fusion 360 par exemple, il suffit simplement de décomposer la modélisation d'une pièce en plusieurs étapes. Dans un premier temps, il nous faudra dessiner un croquis / un dessin en 2D auquel nous donnerons du volume dans un second temps.

Par exemple pour modéliser un cylindre, nous allons tout d'abord dessiner un cercle en 2D, puis nous allons lui donner une hauteur dans un 3ème axe, ce qui va donc donner du volume à notre cercle et nous obtenons en finalité un cylindre. Et il en va de même pour pleins d'autres formes !

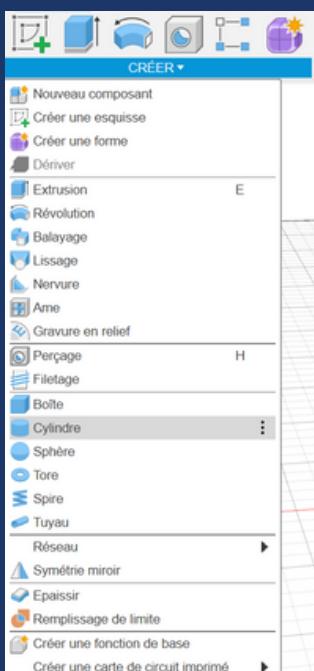


Il faut savoir avant tout, qu'en modélisation 3D, il n'y a ni de bonne ni de mauvaise manière de modéliser, le but est d'arriver au résultat souhaité, peu importe la méthode utilisée. Je vais donc vous présenter aujourd'hui ma méthode qui n'en représente qu'une parmi tant d'autres.

Nous pouvons donc commencer la modélisation de notre arrosoir :

Étape 1 : La base

J'ai choisi de partir sur une base de cylindre à laquelle nous viendrons y ajouter le tube pour verser l'eau ainsi que la anse pour tenir l'arrosoir

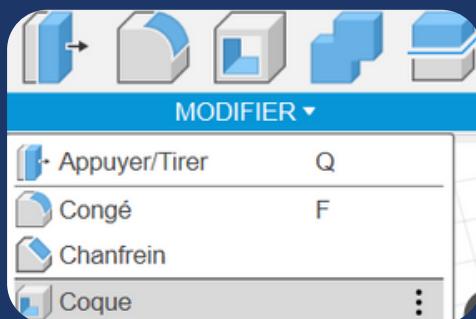


Nous pouvons créer un cylindre très simplement en utilisant un seul outil, l'outil Cylindre. Rien de plus simple.

Veillez à bien renseigner les mesures que vous souhaitez pour votre arrosoir ! Toutes les dimensions que j'ai choisi sont disponible directement sur le fichier F3D ouvrable directement avec Fusion 360 en cliquant [ici](#).

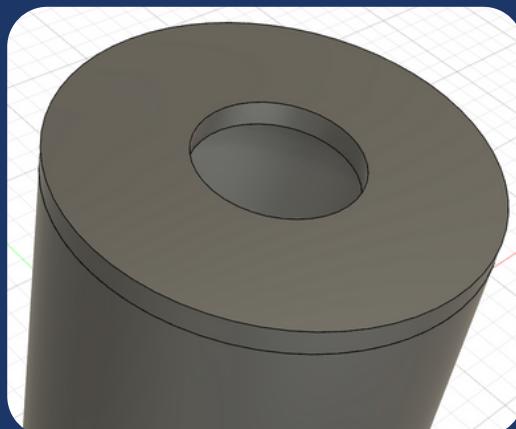
Étape 2 : Création d'une coque

Après avoir créé un cylindre, nous souhaitons pouvoir y mettre de l'eau dedans, il nous faut donc "creuser" dans le cylindre. Nous pouvons faire ceci à l'aide de l'outil "Coque"



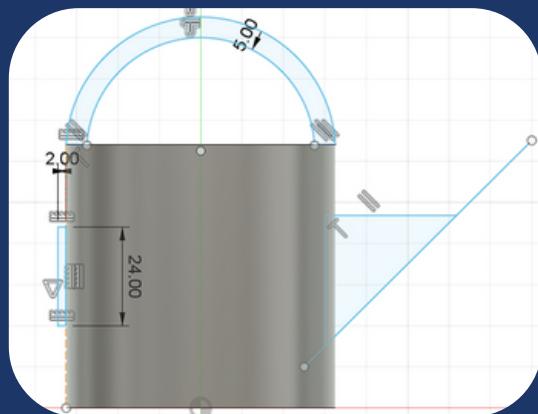
Étape 3 : Création d'un couvercle et d'un orifice

À partir d'ici, nous n'avons ni plus ni moins qu'un très beau verre parfaitement cylindrique, mais ce n'est pas ce que nous souhaitons. Il faut donc maintenant refermer le dessus et y laisser un petit orifice afin d'y insérer de l'eau pour remplir l'arrosoir. Pour ce faire, il nous suffit de créer une nouvelle esquisse (dessins 2D) sur la face du dessus du "verre", y dessiner un cercle du même diamètre ainsi qu'un deuxième plus petit (pour l'orifice) et y donner du volume.



Étape 4 : Crédit de la anse et du tuyau

Pour réaliser la anse, même principe, nous dessinons une esquisse en 2D en forme d'arc de cercle et nous lui donnons du volume ensuite.



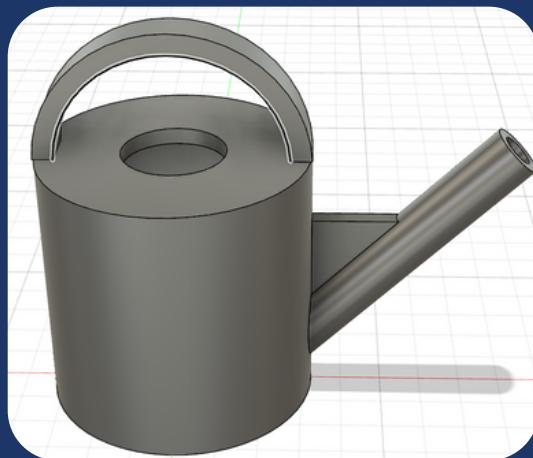
Pour le tube de l'arrosoir, c'est un peu différent. Nous allons utiliser le trait en diagonale qui est visible sur l'image ci-dessus. Nous allons l'utiliser et le “transformer” en tube à l'aide de l'outil “Tuyau”.



Pour finir ce tuyau, il faut maintenant y percer un trou. Il suffit de faire la même choses que les étapes d'avant : dessiner un cercle plus petit sur la face du tuyau et extruder mais dans le sens inverse. Cela retirera de la matière dans le cylindre du tuyau afin que l'eau puisse s'y écouler.

Dernière étape : Finitions

Afin de garantir une certaine rigidité à notre arrosoir, j'ai décidé d'ajouter un renfort triangulaire entre le réservoir et le tuyau afin que ce dernier ne se brise pas au moindre choc.



La dernière étape est purement esthétique mais elle permet de donner un meilleur aspect à notre modèle. Nous allons arrondir certains bords tranchants de notre arrosoir.

Pour faire ceci, nous utilisons l'outil “Congé” :



Il suffit simplement de sélectionner le/les arête(s) que vous souhaitez arrondir et indiquer la valeur du rayon d'arrondissement de l'arête.

Et voilà ! Vous avez maintenant un très bel arrosoir pouvant contenir de l'eau et qui pourra être attrapé par la pince du Ned 2 avec un peu de dextérité !



Impression

Avant de lancer une impression sur une imprimante 3D, il vous faudra dans un premier trancher votre modèle 3D (fichier STL) à l'aide d'un logiciel appelé **Cura** (ou autre logiciel similaire) qui est ce qu'on appelle un "Slicer" (ou un "trancheur" en français).

Voilà pourquoi la majorité des imprimantes 3D à dépôt de matière disponible sur le marché imprime des pièces couche par couche.

Pour pouvoir imprimer une pièce en 3D, il faudra fournir un fichier compréhensible par l'imprimante, on appelle ceci un fichier GCODE.

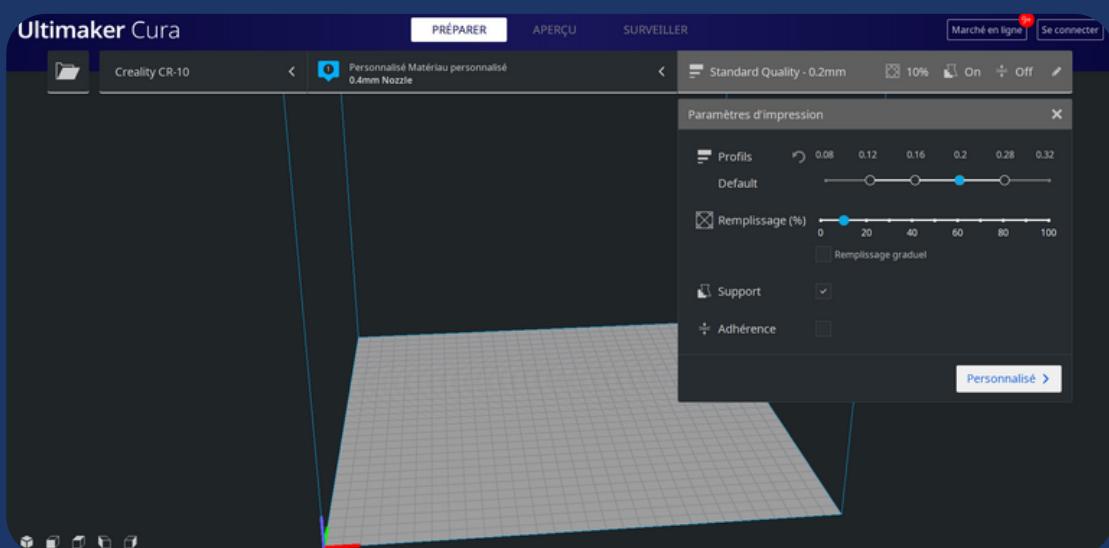
Dans notre cas, c'est **Cura** qui va s'occuper de générer ce fichier.

Voici maintenant comment trancher un modèle et l'imprimer sur une imprimante 3D :

1) Lancez Cura 

2) Réglez les paramètres

Au lancement du logiciel, vous arriverez sur cette fenêtre :



Concernant le premier paramètre "**Profils/Default**", celui-ci vous permet de choisir la hauteur d'une seule couche de plastique.

Etant donné qu'une imprimante 3D dite "classique" à dépôt de filament fonctionne en déposant du filament couche par couche, si vous mettez une hauteur de couche très faible, cela augmentera drastiquement la durée de l'impression mais vous apportera davantage de précision sur la pièce finale.

En général, ce paramètre est laissé sur 0.2

Maintenant à propos du "**Remplissage**", il faut savoir qu'une pièce imprimée en 3D n'est jamais totalement pleine. Par exemple ici :

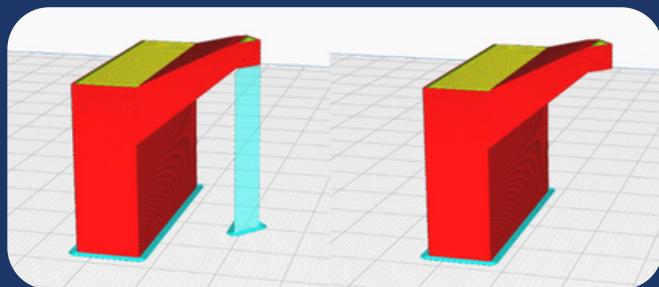
12% 30% 50%



Pour des impressions 3D **peu complexes**, le remplissage à 15% suffit amplement.

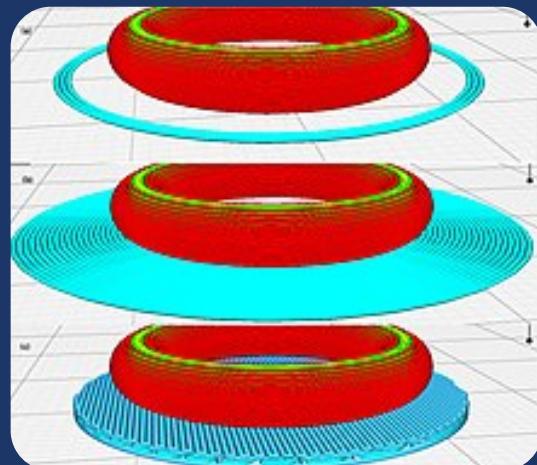
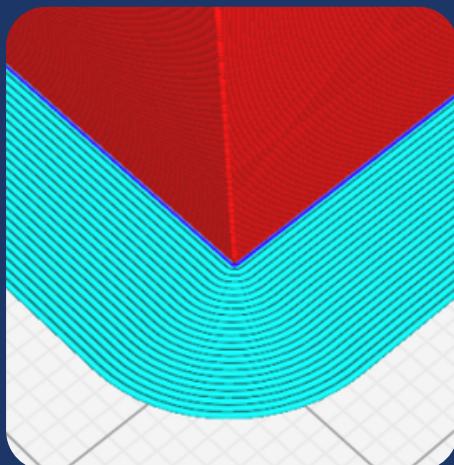
En revanche pour des pièces qui nécessitent bien **plus de rigidité**, il est possible de monter jusqu'à 50% voire même plus pour des pièces extrêmement solides (cas très rare !).

Ensuite pour le paramètre "**Support**", cela dépendra du modèle 3D que vous allez imprimer. Les supports servent lorsque certaines parties de l'impression se font dans le vide, ce qui n'est pas possible de faire sans les supports. Par exemple :



A droite, on peut voir une impression 3D qui à été préparée sur Cura sans support, et à gauche avec support (**pilier bleu**). Le problème de la pièce de droite est que certaines imprimantes ne permettent pas d'imprimer au-dessus du vide sans support comme pour cette pièce. C'est pourquoi, au cas par cas, selon les modèles à imprimer, il faudra ou non activer ou désactiver l'option des supports.

Pour finir, l'utilisation de l'option "**Adhérence**" consiste à ajouter une fine couche de plastique en bordure tout autour de la pièce à imprimer afin d'augmenter la surface d'accroche avec le plateau d'impression et d'éviter quelques soucis par la suite. Voici une image de ce à quoi sert le paramètre "Adhérence" :

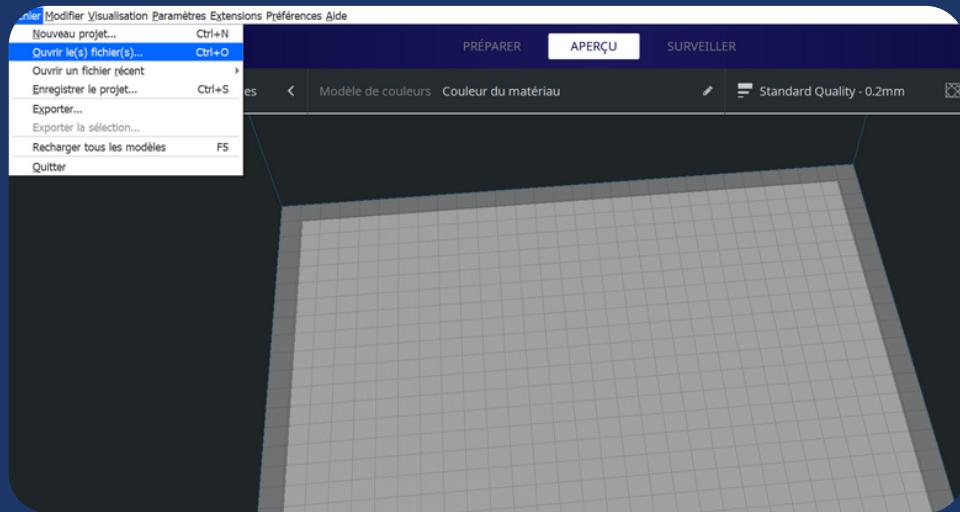


L'activation de ce paramètre permet d'éviter ce genre de problème :

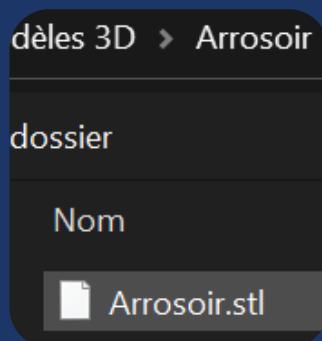


3). Importez votre modèle 3D

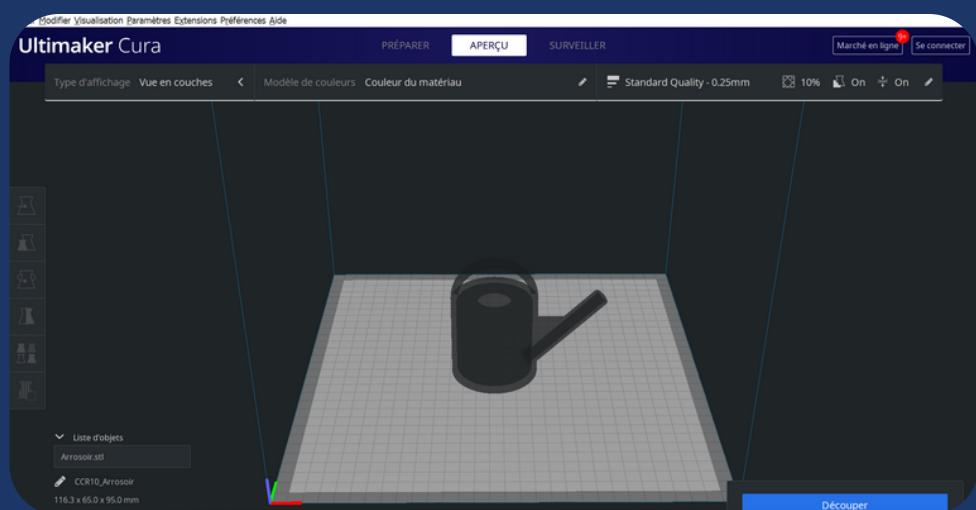
Tout d'abord, allez dans le menu Fichier -> Ouvrir le(s) fichier(s)

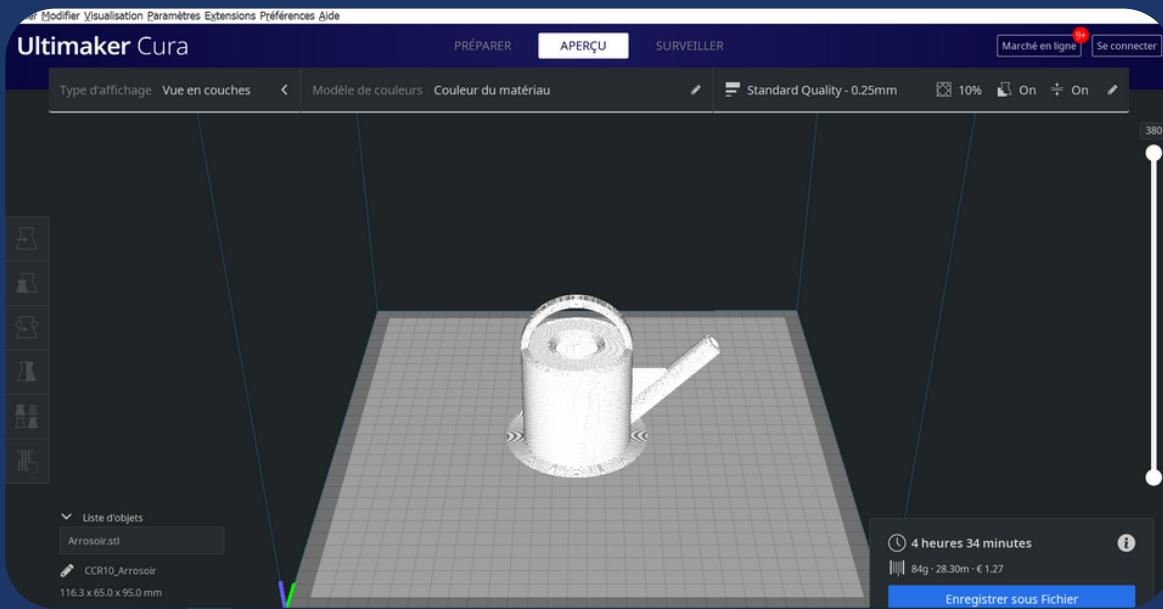


Sélectionnez le fichier à importer :



Votre pièce apparaitra en gris comme ceci. Pour préparer votre fichier après avoir fait tous les paramétrages de l'étape 2, cliquez sur "Découper" en bas à droite :





Voilà ! Votre pièce est prête à être imprimée. Il ne vous reste plus qu'à faire "Enregistrer sous Fichier" et de déposer le fichier dans le périphérique de stockage (USB, microSD, SD, etc) qui sera inséré dans l'imprimante 3D !

En ce qui concerne l'impression 3D en elle-même, si vous avez tout correctement paramétré sur Cura et si votre imprimante 3D est opérationnelle et calibrée, alors tout s'imprimera sans encombre.

Bravo ! Vous avez officiellement terminé ce projet et pouvez en profiter en l'installant dès maintenant chez vous !

Finitions

Ajout d'une prise adaptée pour le préhenseur du Ned 2 (en blanc)



Création d'un boitier facilitant l'utilisation de l'Arduino et ses périphériques



Et voilà pour les finitions esthétiques qui sont bien évidemment optionnelles mais bénéfiques pour le projet.

Sachez que vous pourrez retrouver l'**entièreté** des fichiers partagé sur ce PDF sur ce lien ou en scannant le code QR ci-dessous :

<https://github.com/Mafuo-git/arrosage-automatique-niryo/tree/main>



Pistes d'amélioration



Éclairage
automatisé

Pompe
péristaltique

Pot de fleur
connecté (WiFi ?)



Intégration d'un
écran de feedback
sur le pot

Compacter le
système

Alimentation sur
batterie

Création d'un
arrosoir sur
mesure avec le
préhenseur



À vous de jouer
maintenant !

