

# Groupe A2 : Débitmètre

## **Groupe A2**

- Alexandre MACAULLY - alexandre.macaully@etu.sorbonne-universite.fr
  - Rita SILVA - rita.silva@etu.sorbonne-universite.fr
  - Noah SEGONDS-RICHARD - noah.seconds-richard@etu.sorbonne-university.fr
  - Guillem JOSEPH - guillem.joseph\_planas@etu.sorbonne-universite.fr
- 

## Projet Débitmètre

Dans cette page figurent des explications plus détaillées sur le débitmètre.

## Semaine du 25 janvier 2022

### 1. *Idées de projet*

Chambre à étincelle, Mesure de vitesse (radar) Vélocimètre/Débitmètre à turbine/Débitmètre électromagnétique (vitesse/débit fluide).

### 2. *Essais documentation et dessin 3D :*

Difficultés à prendre en main les logiciels Inkscape et FreeCAD, ainsi que la documentation du wiki.

## Semaine du 31 janvier 2022

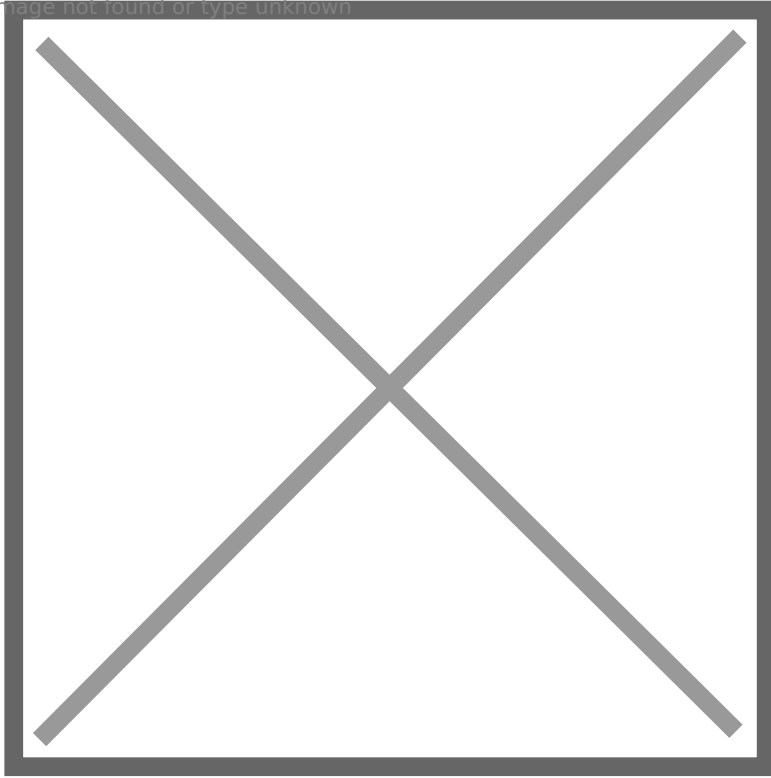
Lors de cette séance, nous devons nous familiariser avec les logiciels de modélisation 3D (Freecad) et 2D (Inkscape).

### **Freecad**

Nous avons pour objectif de modéliser un cube en 3D de 5mm de coté avec 3 trous centrés de 1mm de rayon. Pour ce faire, nous avons commencé par regarder un tuto, ([http://wiki.labaixbidouille.com/index.php/Creation\\_d%27un\\_cube\\_troue\\_avec\\_FreeCAD\\_Francois](http://wiki.labaixbidouille.com/index.php/Creation_d%27un_cube_troue_avec_FreeCAD_Francois)) afin d'avoir une idée de la marche à suivre pour réaliser notre objectif.

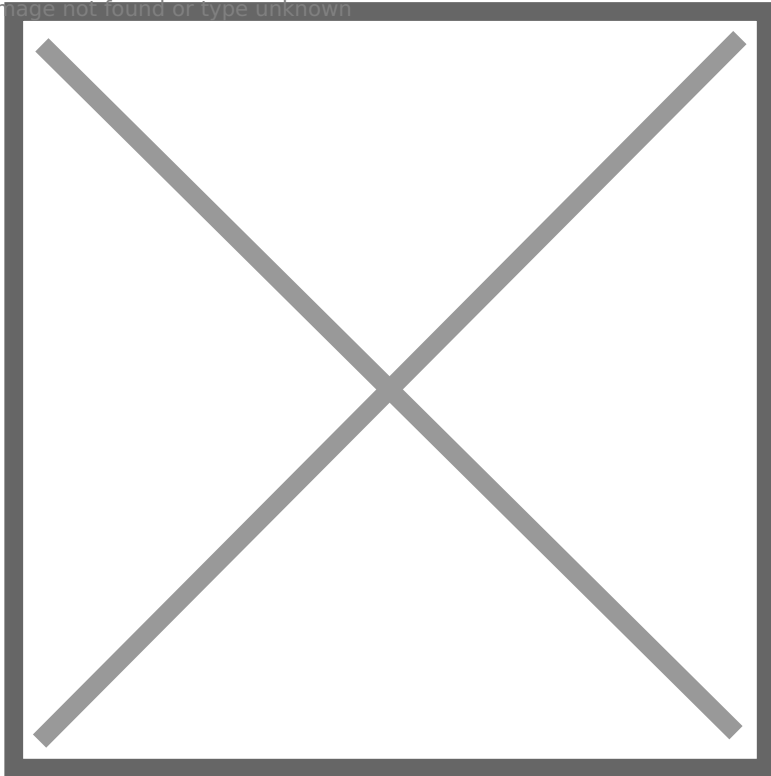
Grâce à ce tuto nous avons pu créer une esquisse d'un carré de 5mm de coté.

Image not found or type unknown



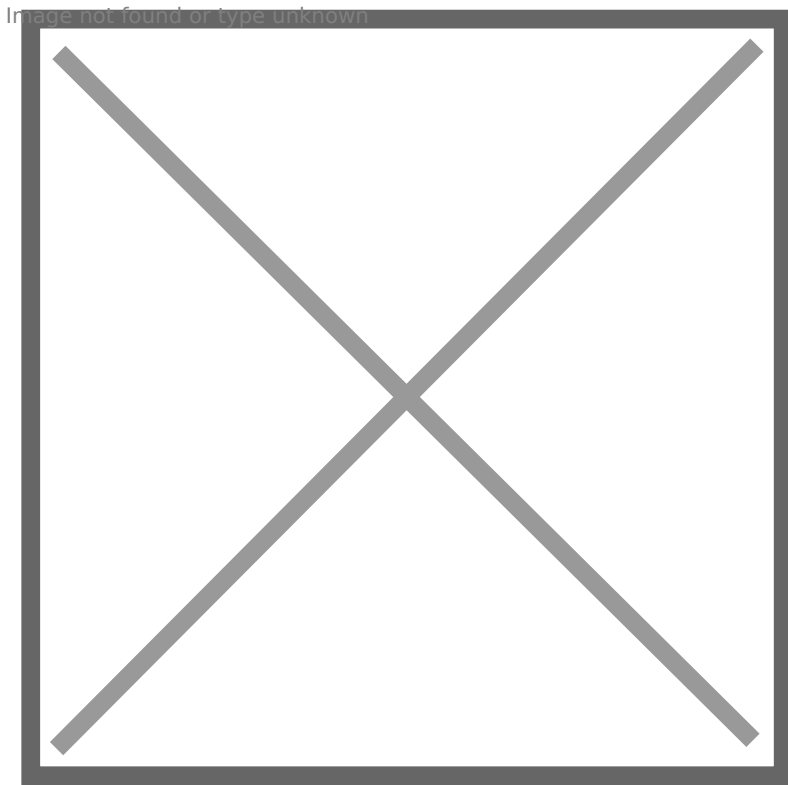
Ensuite, grâce à la fonctionnalité « faire une protrusion de l'esquisse sélectionnée » nous avons pu former un cube à partir de l'esquisse précédemment créé.

Image not found or type unknown



A partir de cette étape nous avons suivi un second tuto (<https://youtu.be/NcvW9L5p6FU>) afin de pouvoir tracer notre cube.

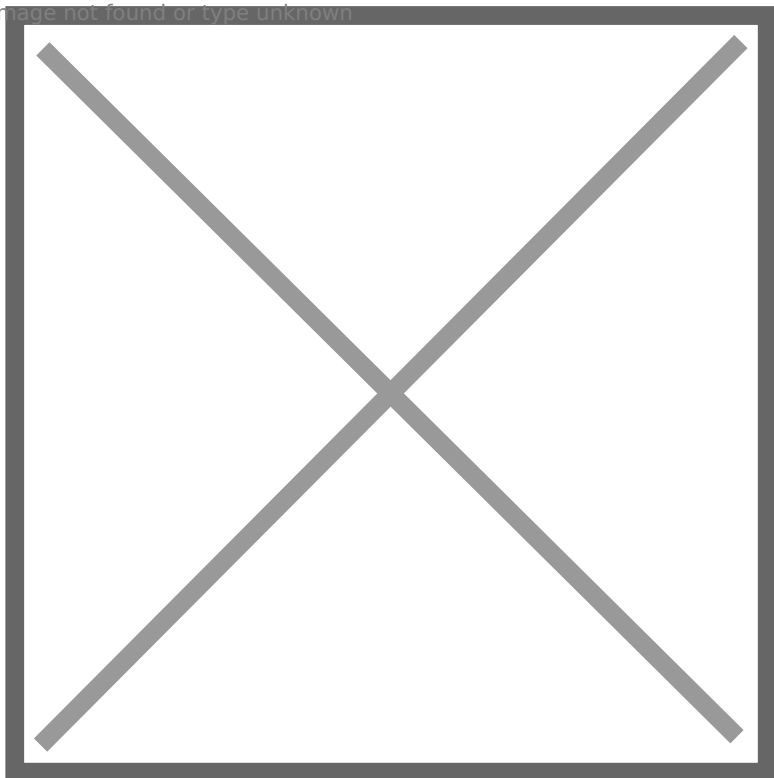
Nous avons donc commencé par créer un cylindre grâce à la fonction « créer un cylindre » de la fenêtre « part ».



Afin de créer le trou, nous avons modifié le diamètre du cylindre ainsi que sa position initiale afin de le centrer sur notre cube.

Nous avons ensuite extruder ce cylindre à notre cube afin de créer un trou dans ce dernier en sélectionnant notre cube ainsi notre cylindre en utilisant la fonction « faire un découpage entre deux formes ».

Image not found or type unknown



Enfin, nous avons répété cette dernière étape 2 fois de plus en modifiant les coordonnées initiales du cylindre afin d'obtenir le modèle attendu.

Image not found or type unknown

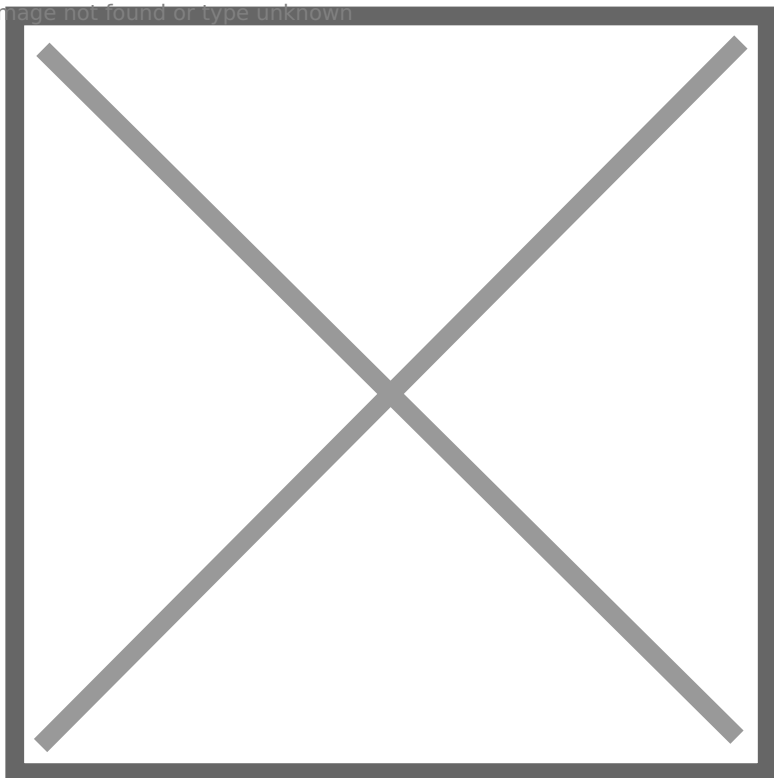
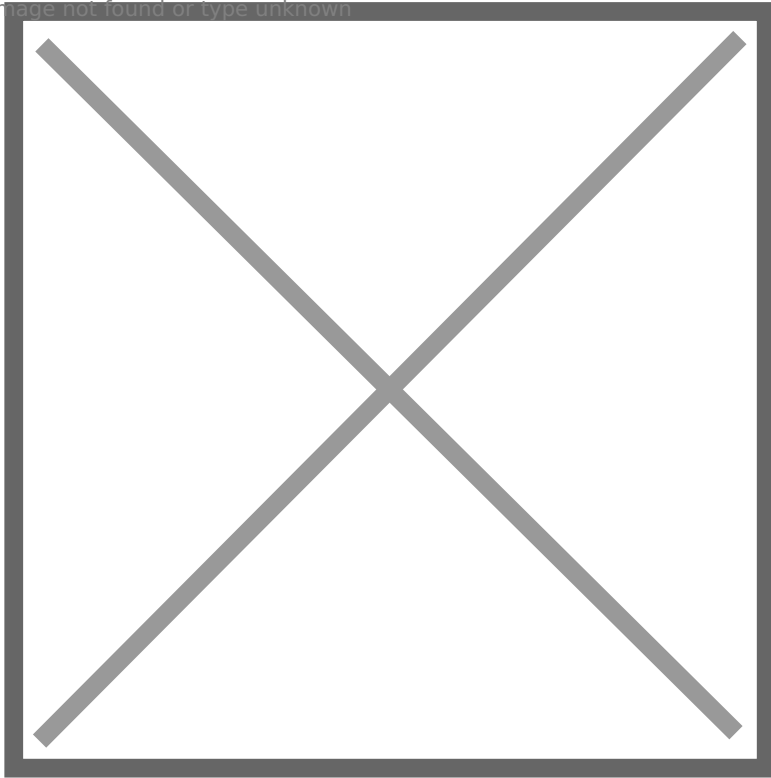


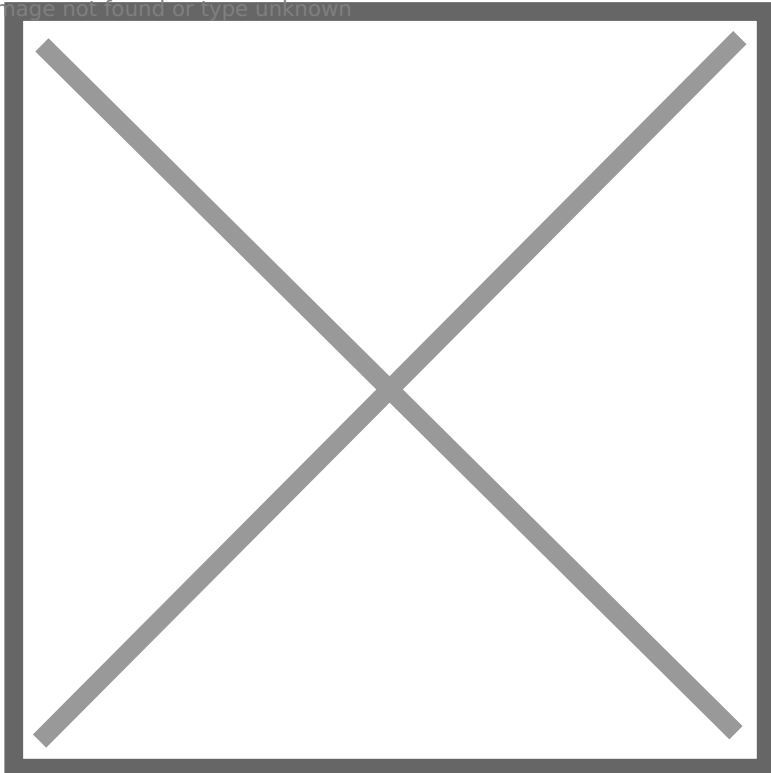
Image not found or type unknown



## Inkscape

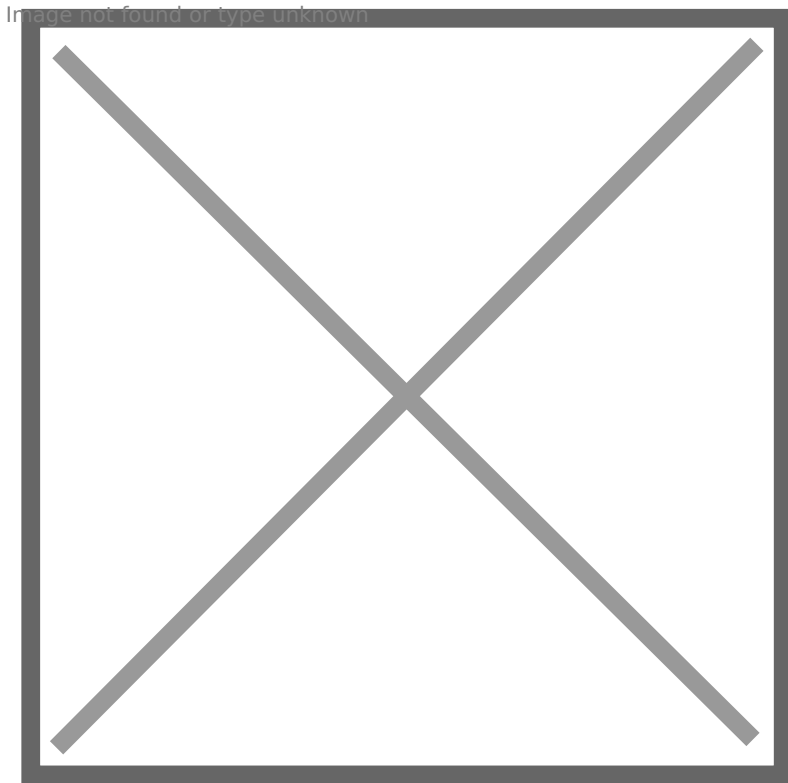
Pour cette deuxième partie du travail, nous avons voulu dessiner un carré de 50mm perforé en chacun de ses coins par des cercles de 2mm de rayon chacun.

Image not found or type unknown



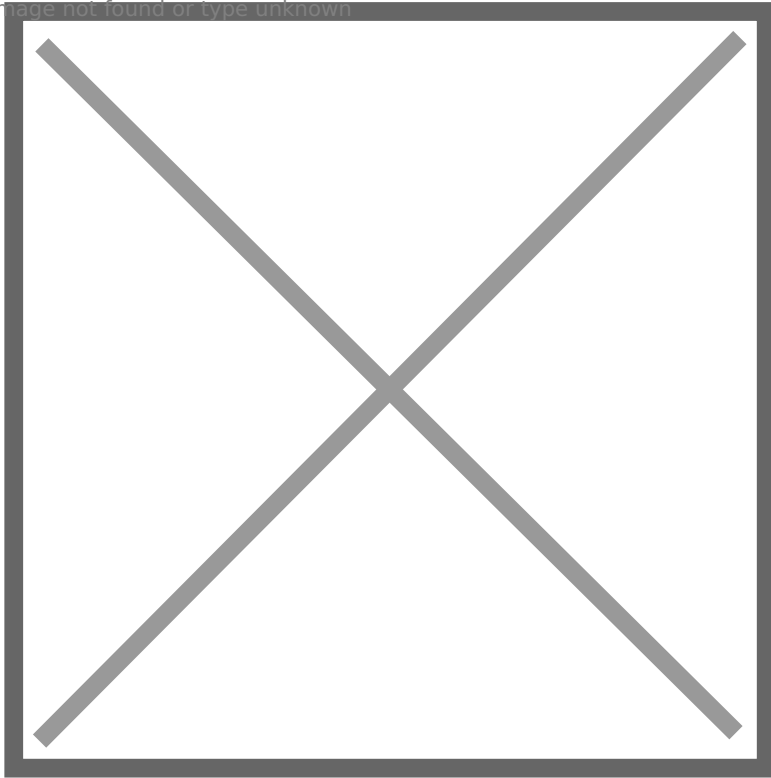
*Première étape :* Dessiner un remplissage carré + dimensionnement + positionnement Sur la barre latérale gauche figure une icône en forme de carré. En cliquant dessus, un carré rempli apparaît. Il

nous faut conserver uniquement les contours de ce carré. Pour ce faire, nous cliquons sur le polygone, et une interface avec la rubrique « Fond et contour » apparaît. Enfin, il suffit de cliquer sur l'icône en forme de croix pour ne laisser que le contour. Nous colorions ce dernier en rouge. A l'aide de la barre horizontale supérieure, nous fixons Hauteur H et longueur L de notre figure à 50mm. Enfin, nous choisissons de positionner le sommet inférieur droit de notre carré, simplement en fixant les coordonnées x et y à la valeur zéro.



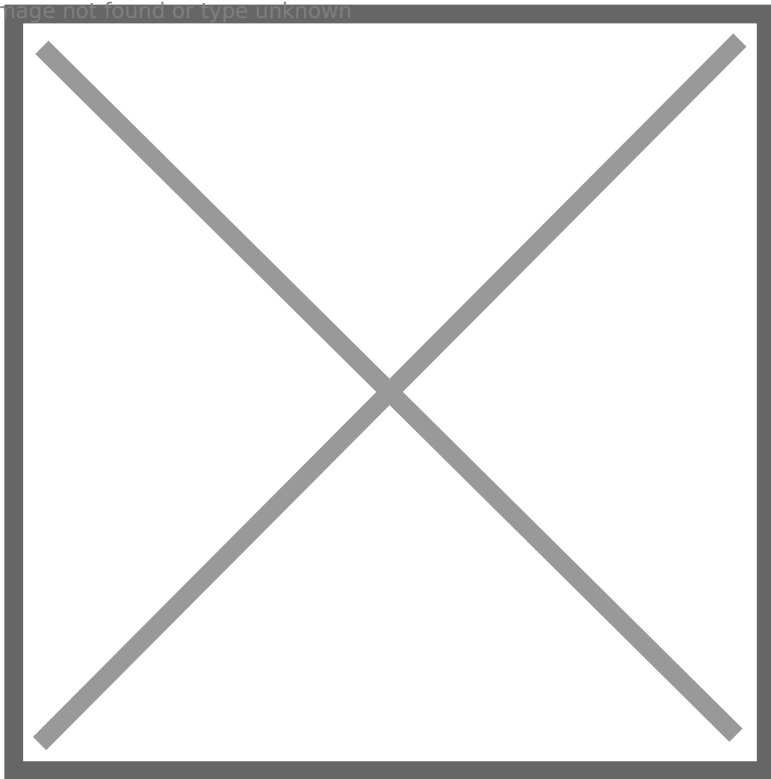
*Deuxième étape :* Dessiner quatre cercles + dimensionnement + positionnement Sur la barre latérale gauche figure une icône « Dessiner des cercles et des ellipses » avec laquelle nous formons un premier cercle. Nous fixons ses dimensions (hauteur H et longueur L) à 4mm. En effet, H correspond au diamètre vertical, tandis que L correspond au diamètre horizontal. On obtient donc un cercle de 2mm de rayon.

Image not found or type unknown



Nous positionnons ce cercle à 1mm du coin inférieur gauche (origine du repère). Nous copions trois fois ce cercle, et nous positionnons les trois copies dans les autres coins du carré, toujours à 1mm de chaque sommet.

Image not found or type unknown

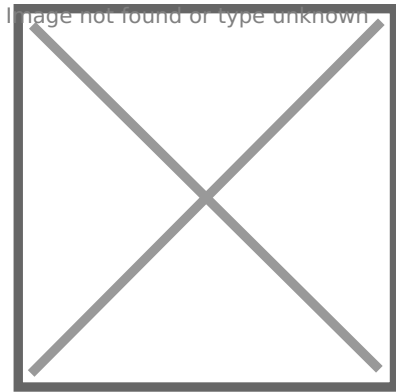


# Semaine du 7 février 2022

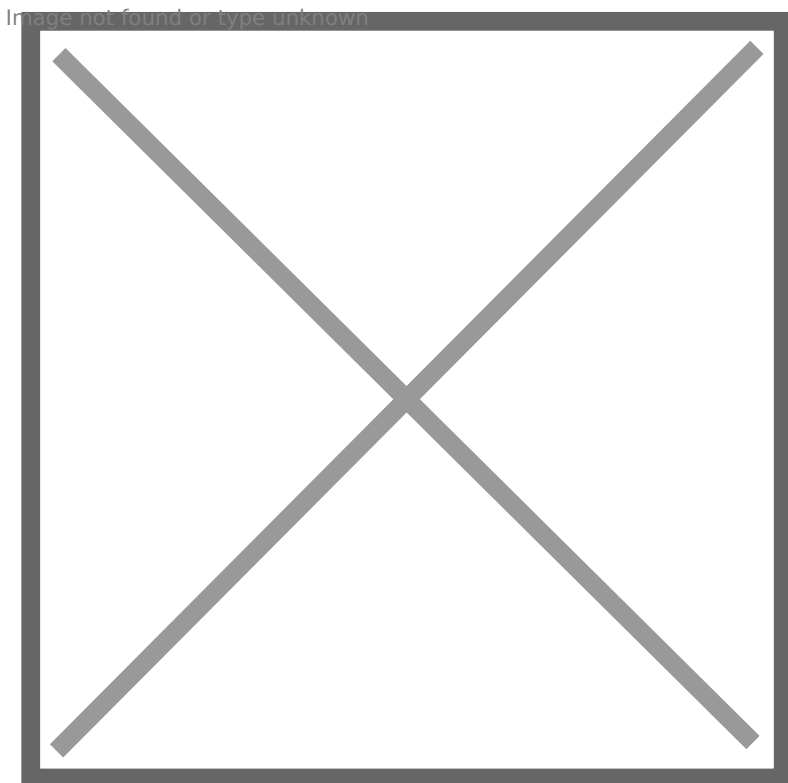
Cette séance nous a permis de prendre en main la carte programmable Arduino.

## Faire clignoter une LED

Pour commencer, nous avons essayé de faire clignoter une LED. Pour cela, nous avons branché en série une résistance  $220\Omega$  et une LED bleue.



Programme:



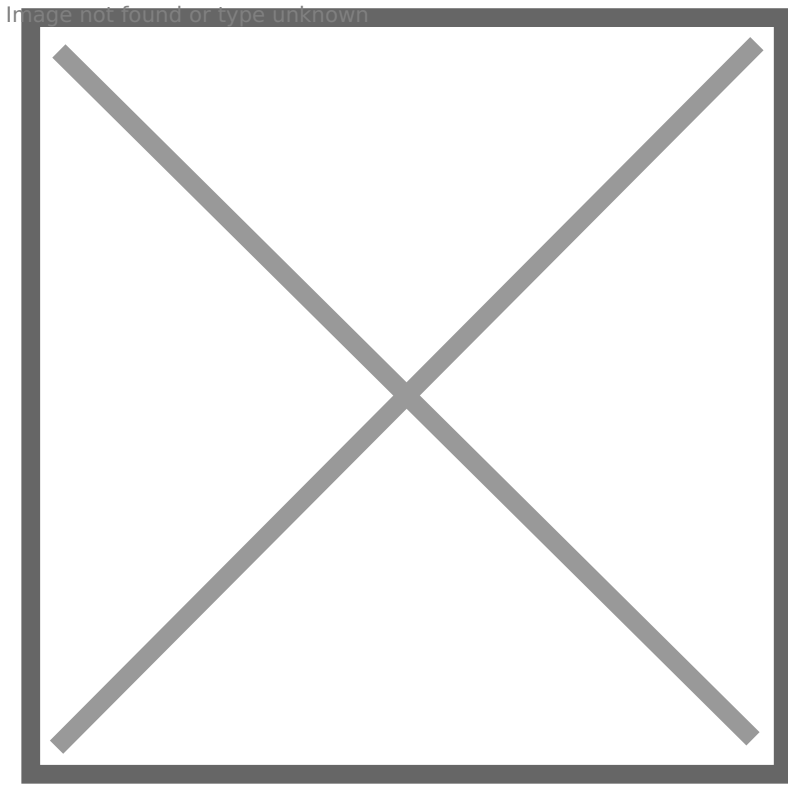
## Chenillard de LED

Nous avons ensuite réalisé un chenillard de LED, c'est-à-dire plusieurs LED qui clignotent avec un décalage temporel entre 2 LED successives. Le montage est le même que celui de la LED qui



clignote, mais répété 3 fois - notre montage étant constitué de 3 LED -

Programme:



## Température

Pour ce projet nous avons suivi deux tutoriels: [https://www.youtube.com/watch?v=v8Pway-5sFY&list=PL0YfVdOGWSEQCIEZj\\_-dFsnClqyxqDmA\\_&index=6](https://www.youtube.com/watch?v=v8Pway-5sFY&list=PL0YfVdOGWSEQCIEZj_-dFsnClqyxqDmA_&index=6) et [https://www.youtube.com/watch?v=8s5wY9kljyE&list=PL0YfVdOGWSEQCIEZj\\_-dFsnClqyxqDmA\\_&index=9](https://www.youtube.com/watch?v=8s5wY9kljyE&list=PL0YfVdOGWSEQCIEZj_-dFsnClqyxqDmA_&index=9).

Nous avons utilisé le capteur «thermat» et le «1602 display» (l'écran) afin de détecter la température de la salle et ensuite l'afficher sur un écran numérique.

Nous avons le montage final suivant (capteur + écran):

Image not found or type unknown

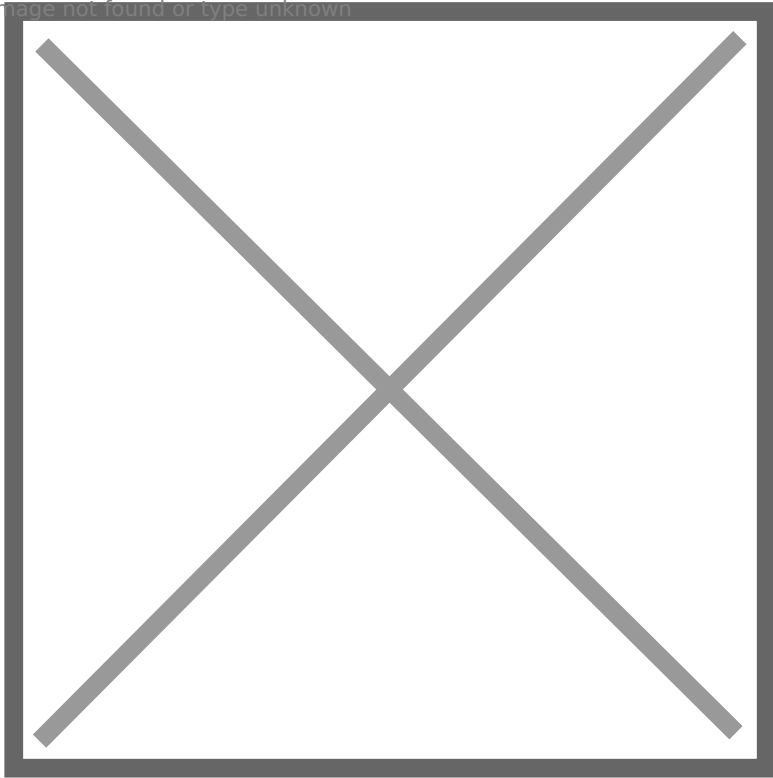


Image not found or type unknown

Pour plus de détails sur le montage vous pouvez regarder les liens, le premier concernant l'écran et le deuxième le capteur de température.



Programme:

*Remarques:* Nous avons eu des problèmes pour afficher une température qui nous semblait acceptable pour la salle du Fablab (entre 18°C et 23°C). Après avoir testé les différentes parties de

notre montage et du programme nous avons compris que l'erreur était due à la partie du programme concernant la conversion en °C de la grandeur mesurée par le capteur. En effet, notre capteur ne mesurait pas directement les différences de température. Ainsi, pour afficher une température décente, il nous fallait faire des conversions. Nous n'avons pas réussi à trouver les bons numéros car nous n'avions pas les spécifications du capteur. Mais nous avons constaté que la valeur obtenue par le capteur changeait en faisant varier sa température alentour. Cependant, bien que l'écran affichait une valeur, ce n'était pas la grandeur souhaitée.

## Semaine du 14 février 2022

Dans cette séance, nous avons choisi quel sera notre projet. Nous voulons réaliser un débitmètre à l'aide d'un moulin à eau miniature et d'un laser. A l'aide d'une photodiode placée derrière le moulin (lui-même placé entre le laser et la photodiode), nous pourrions acquérir à l'aide de la plaque programmable Arduino le nombre de tours par minute du moulin à eau. Ceci nous permettra de connaître le débit du fluide qui fait tourner le moulin de notre instrument. Nous pourrions potentiellement afficher le résultat sur un écran relié à la plaque Arduino.

Pour ce faire, nous avons commencé par essayer de modéliser notre moulin à eau en 3D sur FreeCAD. Après quelques recherches, nous sommes parvenus à trouver un moulin pré-modélisé qui correspondait à l'idée que nous en avions. Après quelques petites modifications, nous sommes parvenus à la modélisation finale de notre moulin. A la prochaine séance, nous commencerons à programmer sur la plaque Arduino afin d'effectuer la bonne conversion entre le nombre de tours par minute et le débit. Aussi, nous modéliserons le reste des pièces à imprimer qui joueront un rôle-clé concernant le fonctionnement de notre moulin ; parmi celles-ci, un axe autour duquel le moulin devra tourner.

## Semaine du 21 février 2022

*Au programme:*

**1) Listing des tâches à effectuer**

**2) Répartition du travail**

*Rita et Noah :* Pour rappel, il est prévu que le débitmètre fonctionne à l'aide d'une lumière laser pulsée. Cette pulsation doit être obtenue par l'interruption périodique du faisceau laser au contact d'un obstacle opaque. Avec le logiciel FreeCAD, Rita et moi avons perforé le moulin non loin de son bord. Le trou effectué permettra, au gré de la rotation du moulin, de générer une pulsation "artificielle" du laser ; tantôt le faisceau passera par le trou, tantôt il sera bloqué par la structure-même du moulin.

*Communément :* Nous sommes retournés tous ensemble au FabLab dans le courant de l'après-midi afin d'y effectuer notre première impression 3D : celle du moulin lui-même.

La prise en main de l'imprimante 3D n'est pas très compliquée : deux étapes sont requises. Premièrement, il faut posséder le fichier format STL de FreeCad sur une clé USB, qu'il convient ensuite de l'importer sur le logiciel *IdeaMaker* disponible sur les ordinateurs du FabLab. À l'aide de l'interface du logiciel, il est possible de choisir les paramètres d'impression : épaisseur de couche et robustesse, finitions, position de la pièce à imprimer, utilisation de supports d'impression, visualisation du résultat... . Deuxièmement, il faut exporter la modélisation sur la clé et insérer cette dernière sur le port USB de l'imprimante. L'écran tactile de la machine permet la sélection de la modélisation à imprimer, et de lancer l'opération d'impression.

Photo du 1er modèle de moulin :

[d3a56a5d-d1ae-4113-9e44-e6868d21392c.jpeg](#)  
[0948b726-6d3d-4aa2-82e3-4e874d986755.jpeg](#)

## Semaine du 8 mars 2022

Des modifications sont à apporter au moulin : le remplissage et le nombre de pales sont trop importants et ne permettent pas de mettre correctement la roue en rotation ni de mesurer le volume d'eau écoulé. Il faut également avancer sur la conception et la réalisation de la structure externe de l'instrument Répartition du travail :

**Alexandre et Noah** : Modélisation des parois externes du débitmètre.

**Rita et Guillem** : modification de La structure du moulin.

Modélisation 3D du modèle final du moulin : **(cf page annexe pour plus de détails)**

[6cfb1b9-b8dc-4c3e-b3fb-6dbc98ff736b.jpeg](#)  
[536ed39c-b349-4516-904b-b901ae6b00b8.jpeg](#)

Modélisation des parois du débitmètre où il y aura la photodiode et le laser : cavité du moulin

[070f1a5b-4ae7-4b38-9473-e0f706f47d34.jpeg](#)  
[1ad90124-0d83-4f18-8094-cf73e7f6d33d.jpeg](#)  
[52e01d43-b39a-4608-b5ea-d1ce4c561407.jpeg](#)

## Semaine du 15 mars 2022

**Ordre du jour : poursuite de la modélisation 3D des pièces de l'instrument.**

*Rita et Guillem* : modélisation de la partie électronique de notre débitmètre afin que le circuit Arduino soit protégé de l'eau. L'idée est aussi de rendre le débitmètre plus esthétique.

Plafond et paroi de la protection :

[3f2c636b-aab6-4425-b703-f3674ed8d47f.jpeg](#)

Support du circuit électronique :

[d16c930d-fbe2-4ac5-baf4-303b753953e1.jpeg](#)

*Alexandre et Noah* : modélisation de la base ( “sol” ) de l’instrument. Cette base est constituée en plusieurs parties.

——> **Plan incliné** : Il s’agit de la pièce à paroi latérale triangulaire.

——> **Excavation au pied du plan incliné** : Il s’agit d’un creux rectangulaire dont la longueur est égale à la profondeur du **plan incliné**. Ces deux premières pièces servent à l’évacuation de l’eau.

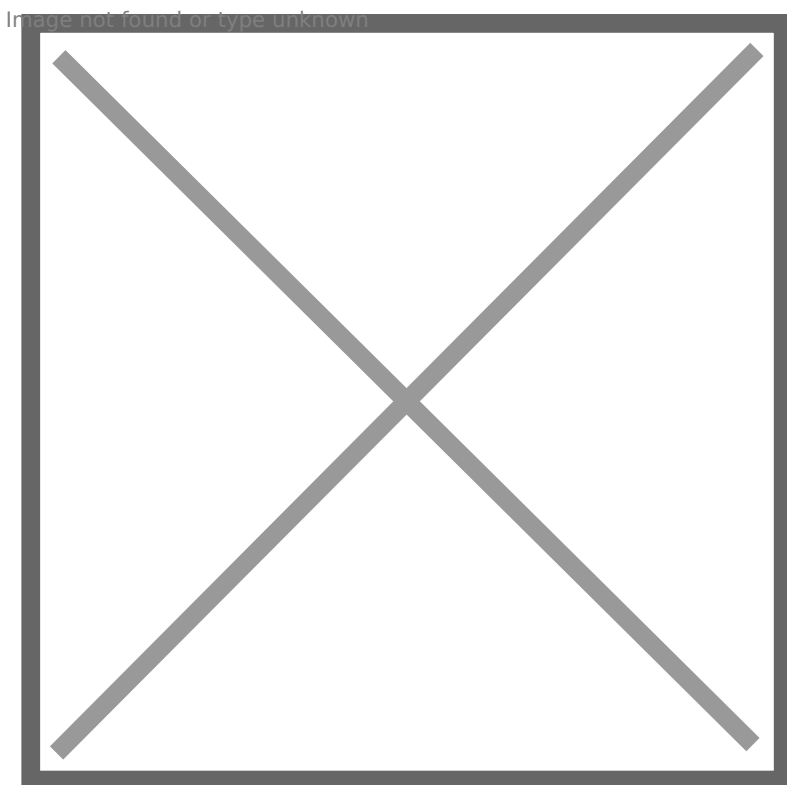
——> **Support** : Il correspond au plus gros pavé ; il soutient toute la base de l’instrument.

——> **Parois externes** : Elles s’apparentent à des pavés fins et très longs, et embrassent une bonne partie du pourtour du **support** afin de rendre plus solide la structure globale et d’éviter des quelconques décalages.

[282f78b2-51dd-4471-9562-f2a567bebb97.jpeg](#)

## Semaine du 22 mars 2022

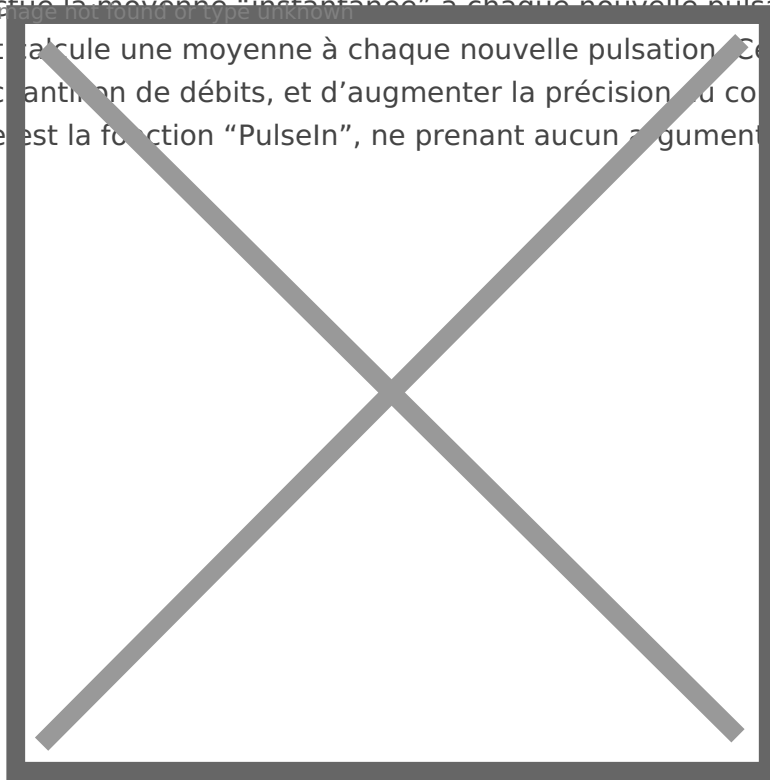
Ordre du jour : récupération des impressions 3D de la semaine passée, lancement des dernières impressions, poursuite du code Arduino.



### Code Arduino :

Petite modification dans notre montage électronique analogique : nous n'avons pas de photodiode. Par conséquent, nous devons y substituer une photorésistance. Ceci ne pose pas de contrainte supplémentaire, étant donné que la carte Arduino est dotée de pins d'alimentation ; l'alimentation de la photorésistance s'effectue sans passer par un générateur extérieur. Afin de mesurer notre débit, nous avons besoin de quantifier une durée de rotation.

À l'aide d'un débit connu (certaine force d'écoulement de l'eau) : mesurer durée d'un certain nombre pulsations ( 1 pulsation = 1 éclaircissement  $\Rightarrow$  faible tension photorésistance OU = 1 "obscurité"  $\Rightarrow$  forte valeur tension photorésistance), et en faire la moyenne. On a ainsi une corrélation entre une valeur déterminée de débit et une durée moyenne de pulsation. Cette corrélation va être utilisée comme étalon. Il suffit alors d'effectuer un produit en croix pour obtenir la valeur actuelle du débit. Quelques précisions : Calcul de la durée moyenne d'une pulsation : le programme effectue la moyenne "instantanée" à chaque nouvelle pulsation. Il accumule la durée des pulsations et calcule une moyenne à chaque nouvelle pulsation. Ceci offre l'avantage d'obtenir un plus ample échantillon de débits, et d'augmenter la précision au cours du temps. L'élément principal du code est la fonction "PulseIn", ne prenant aucun argument et renvoyant la durée



d'une pulsation.

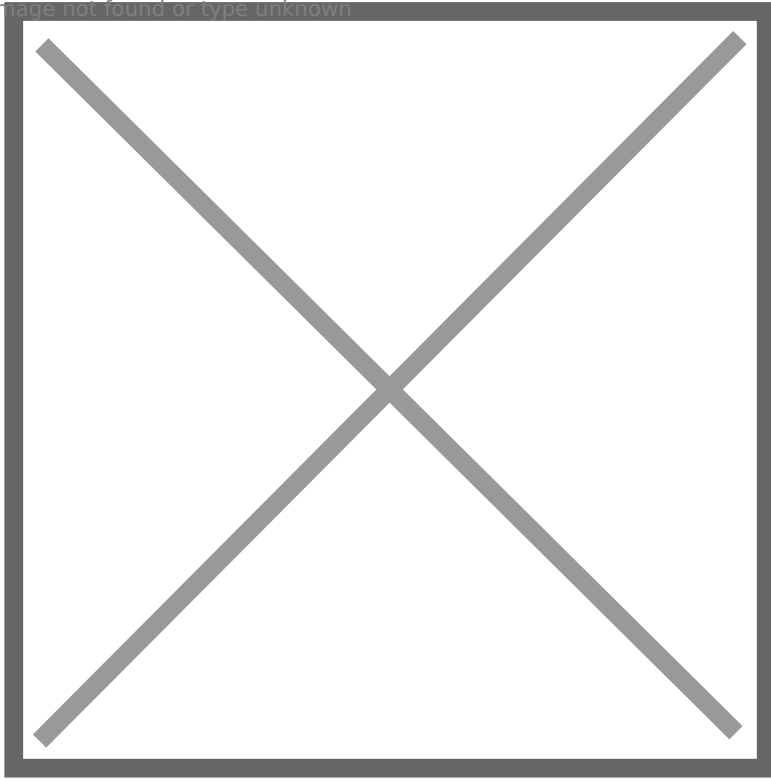
### Modification de la source de lumière :

Nous nous sommes rendus compte que la focalisation du faisceau laser ne facilite pas la détection par la photorésistance. Nous optons plutôt pour une lampe dont le faisceau est bien plus diffus et touche plus facilement la photorésistance.

### Débitmètre avec le laser

Débitmètre avec la lampe :

Image not found or type unknown



*Choix de la source d'eau :*

Jusqu'ici, nous n'avions pas de réelle idée quant à la façon dont produire un débit d'eau. Cependant, nous optons pour une jarre que Rita et Guillem ont achetée chez HEMA. Celle-ci va s'avérer très utile pour générer et contrôler un débit d'eau par l'intermédiaire d'un petit robinet. En plus de cela, la jarre n'est pas trop volumineuse et permet son déplacement dans trop de difficultés.

## Semaine du 29 mars 2022

**Ordre du jour : finalisation du projet et préparation de la soutenance.**

Nous choisissons de répartir les tâches :

*Alexandre et Guillem :* Nous avons finalement décidé de modifier la structure de notre code afin de simplifier le circuit Arduino. En effet, afin d'utiliser la fonction « pulseIn » nous aurions dû utiliser un montage comparateur. Nous avons donc opté pour une structure plus traditionnelle avec des boucles nous permettant de réaliser notre objectif.

**Pour plus de détails sur le code, le circuit ainsi que le traitement du signal, cf la page annexe.**

*Rita et Noah :* élaboration de la structure de la soutenance et participation aux tests du débitmètre.

*Tous ensemble :* nous effectuons une série de tests dans lesquels nous utilisons la jarre pour générer l'écoulement de l'eau. Au gré des rotations du moulin, nous suivons à l'écran de

l'ordinateur le nombre de trous illuminés. Nous vérifions les valeurs que l'afficheur numérique présente. Nous rencontrons beaucoup d'erreurs et d'incohérences, que nous parvenons à corriger progressivement. **Exemples d'erreurs :**

Enfin, nous vous proposons une vidéo de l'ensemble du débitmètre ainsi qu'une seconde démontrant le fonctionnement de l'instrument.

[Structure du débitmètre](#)

[Fonctionnement du débitmètre](#)

*Tâches:*

#### **Structure**

1. Axe (dimensions, maintien, matériel, conception)
2. Perforation du moulin
3. Positionnement et maintien du laser et de la photodiode
4. Achat d'un récipient à volume connu et délivrant un débit réglable
5. Réglage avec un volume d'eau connu

#### **Hardware et Software**

5. Codage arduino, affichage
  6. Réalisation du circuit
- 

## Modélisation 3D des différentes pièces:

Sur chacune des parois figure une ouverture carrée ainsi qu'une plate-forme rectangulaire accolée à la base de l'ouverture. Ceci permet d'y poser simplement le laser ou la photodiode. Quant aux ouvertures carrées, elles jouent le rôle de communication entre laser et photodiode, puisque nous souhaitons illuminer cette dernière.

*Technique utilisée pour la fixation des parois :*

La paroi de la lampe et celle de la photodiode s'imbriquent l'une dans l'autre par un système de tenons et de mortaises. Aux deux extrémités de la paroi de la photodiode se trouvent deux "protusions" appelées tenons. Sur celles de la paroi de la lampe figurent des renforcements appelés "mortaises" et dont le volume est "complémentaire" à celui des tenons. Nous avons cependant appliqué une différence de diamètre de

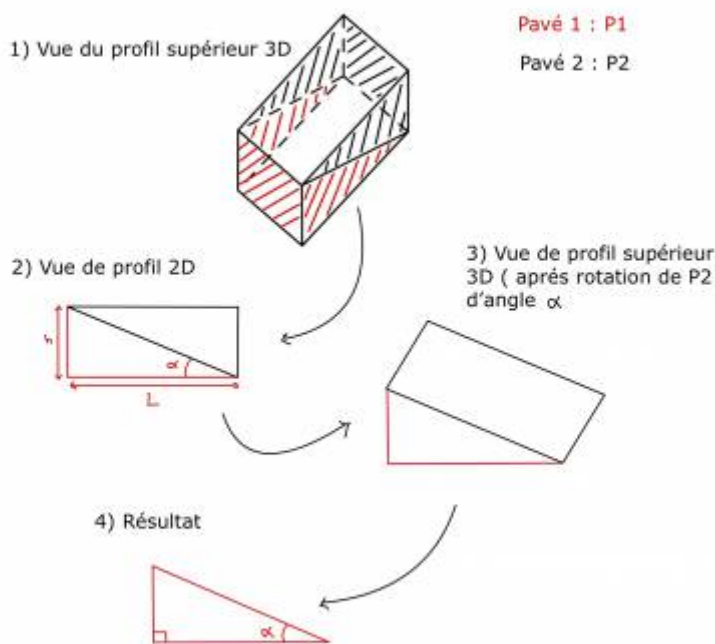
1mm entre tenons et mortaises afin de faciliter l'imbrication.

Ainsi, les deux parois font s'aligner les fenêtres carrées et permettent la "communication" du laser et de la photodiode ; le faisceau laser traverse toute la cavité d'une fenêtre à l'autre.



### Détails sur la modélisation du plan incliné :

Le plan incliné joue un double rôle : réception et évacuation de l'eau écoulee. Sa technique de conception repose toujours sur un principe de soustraction booléenne. Cependant, quelques étapes supplémentaires sont à prendre en compte. Notre objectif est d'obtenir un trapèze en 3D dont les faces latérales superposables. Chacune constitue un triangle rectangle, dont la hauteur vaut "h", et la base "L". Deux angles sont d'intérêt : L'angle droit qui se trouve au sol, et l'angle "alpha" qui se trouve être le plus éloigné. Avec l'outil de soustraction booléenne, il nous faut partir de deux pavés d'égales dimensions, l'un rouge et l'autre noir, et orienter l'un des deux d'un angle "alpha". Or, ce dernier ne peut pas être choisi au hasard, mais vaut précisément : "alpha" =  $\arctan(h/L)$ . Nous comprenons donc qu'il faut orienter le cylindre noir (par exemple) d'un angle  $\arctan(h/L)$ . Enfin, nous procédons à la soustraction : Pavé rouge - Pavé gris, afin de ne se retrouver qu'avec le trapèze 3D décrivant le plan incliné.



## Circuit électronique

*Rappel du fonctionnement:* Avec les pièces que l'on a construites, on veut faire arriver un signal sur un capteur, ce signal correspondant aux variations d'intensité lumineuse de la lumière produite

par un laser. Ces variations d'intensité lumineuses sont directement liées à la vitesse de rotation de la roue, et donc au débit. Un débit grand produira plus de variations en moins de temps, un débit faible en produira moins. Si l'on décide d'un débit de référence connu, et qu'on mesure quelle est la vitesse de rotation de la roue qu'il engendre (et en faisant l'hypothèse que la variation du débit en fonction de la vitesse de rotation est linéaire) on pourra trouver n'importe quel débit en utilisant ce débit de référence et une vitesse de rotation mesurée avec notre instrument.

*Sélection du capteur adapté:* D'après ce que nous avons décrit précédemment, nous avons besoin d'un capteur très sensible aux variations d'intensités lumineuses. Après avoir fait quelques tests, nous avons décidé de choisir une photo-résistance (à la lumière ambiante la valeur de sa résistance vaut environ 30kohm, à la lumière artificielle elle vaut environ 20kohm, et quand on la cache de la lumière elle vaut près de 100kohm). Nous nous sommes rendus compte que notre montage, tenant compte de tous les supports, cachait assez bien la photo-résistance de la lumière ambiante, donc le mode "lampe" de notre pointeur laser suffirait pour produire des variations mesurables. Par ailleurs, on évite comme cela d'avoir à aligner parfaitement la photorésistance et le laser.

*Circuit:* L'arduino est capable de lire des tensions. Il nous faudrait donc alimenter notre photo-résistance en courant, puis lire la tension à ses bornes: d'après la loi d'Ohm, la tension est proportionnelle à la valeur de la résistance (donc haute tension si obscurité → blocage de la lumière par la roue, tension faible si lumière → passage de la lumière par un trou de la roue). Cependant, on décide d'utiliser l'Arduino comme générateur aussi, et elle sert uniquement comme générateur de tension. Désormais, il ne nous suffit plus d'alimenter la photorésistance en tension puis juste lire la tension à ses bornes: on va toujours lire ce qu'est produit par l'Arduino et on ne va pas avoir de variation de tension. C'est pour cette raison que l'on fait appel à un montage pont diviseur de tension, avec une première résistance de valeur 10kohm (valeur standard, pas de choix particulier).

Image not found or type unknown



De toute façon, la tension lue par l'Arduino évolue de la même façon qu'avant en fonction des variations de résistances: elle va augmenter quand la résistance est forte, diminuer quand elle est faible.

---

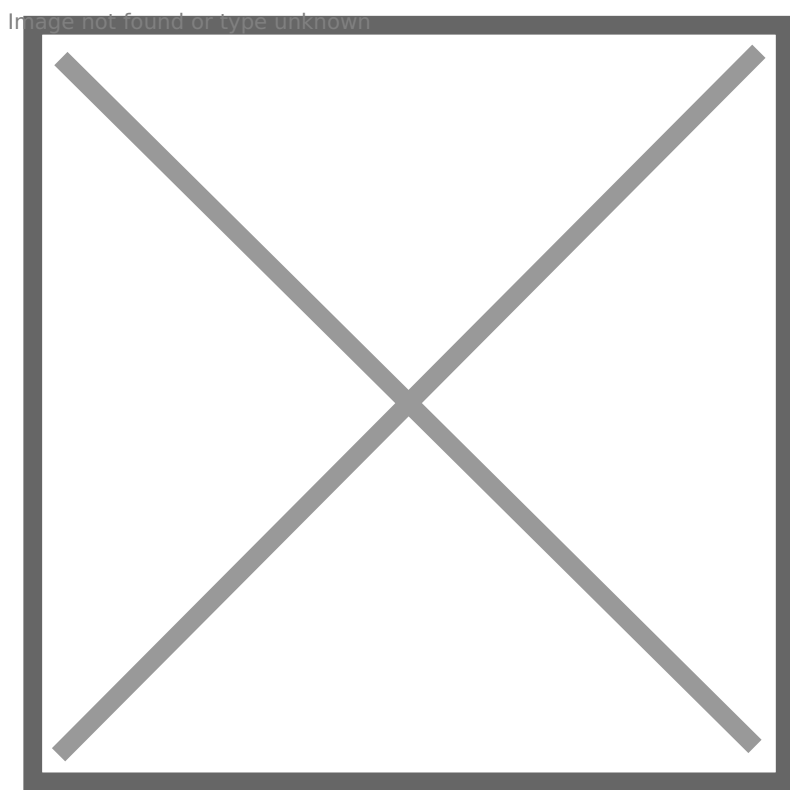
## Traitement du signal

*Signal produit en sortie du montage:* Notre but est de faire interpréter à notre programme que quand la photorésistance est éclairée, il doit mesurer un temps, et quand elle ne l'est pas, la mesure doit s'arrêter. Avec cela, on saura combien de temps dure le passage de la lumière dans un trou (temps court → rotation rapide, débit fort / temps long → rotation lente, débit faible). Le problème se pose alors de comment faire pour que l'Arduino interprète les notions "photorésistance éclairée" et "photorésistance non éclairée". Notre première idée est d'utiliser un montage comparateur: on compare la valeur de la tension aux bornes de la photorésistance et une tension de référence créée avec un montage pont diviseur et des résistances connues grâce à un AOP. Cette tension de référence serait inférieure à la tension aux bornes de la photorésistance quand elle est éclairée, mais inférieure à la tension aux bornes de la photorésistance quand elle ne l'est pas. De cette façon, en sortie de l'AOP, on a un signal binaire au lieu d'une suite de valeurs continues: soit il y a de la tension, soit non. On peut alors utiliser une fonction sur Arduino qui déclenche un compteur de temps quand elle reçoit une tension, et qui l'arrête quand elle n'en reçoit plus. On voit bien maintenant que cette idée aurait marché, mais elle était loin d'être

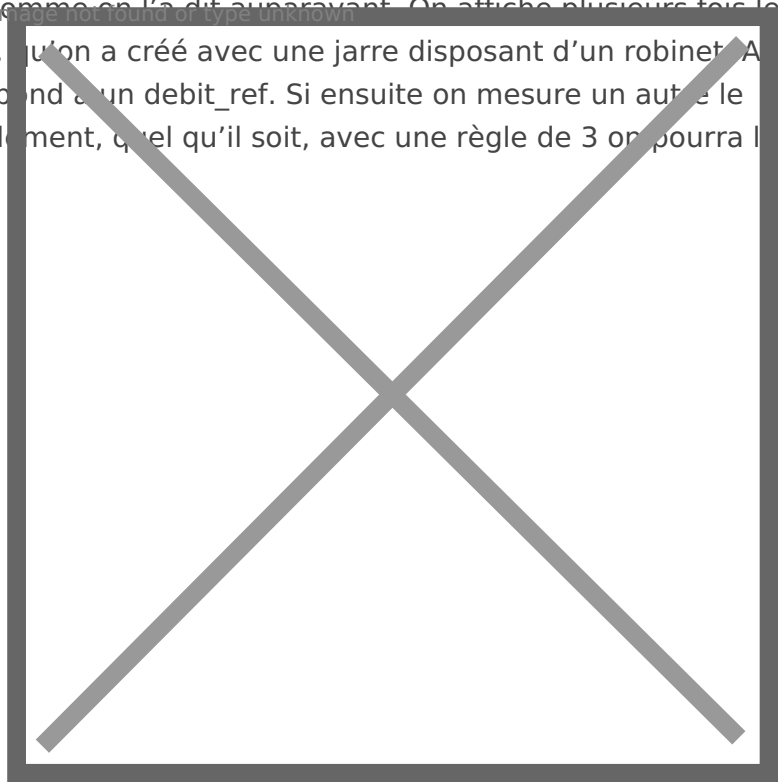
efficace. Toutefois, c'était la seule manière que l'on connaissait pour procéder: on s'était beaucoup formés en électronique au premier semestre, et beaucoup moins en numérique, et cela explique pourquoi on était plus à l'aise. Mais avec une introduction rapide au fonctionnement du codage en C, faite par notre enseignant, on a vu qu'on pouvait régler notre problème avec juste quelques lignes de codes, sans avoir à faire appel à un montage comparateur.

### **Code:**

*Initialisation et partie voidloop():* On décide premièrement de voir comment se fait la lecture des valeurs de la tension sur l'Arduino. On fera cela grâce à la fonction `Serial.print()`, qui nous sera très utile. On lit la tension aux bornes de la photorésistance et on voit que l'Arduino affiche 400 quand la photorésistance n'est pas éclairée, et 200 quand elle l'est. Il nous faut donc une tension de seuil d'environ 300. On fait le choix arbitraire de mesurer un temps quand la lumière passe à travers le trou (on aurait pu mesurer le temps d'obscurité, au lieu du temps d'éclairement). On fait cela grâce à la fonction `millis()`, qui donne le temps à la milliseconde près depuis que le programme a commencé à s'exécuter, en utilisant l'horloge de l'ordinateur. On va mesurer ce temps plusieurs fois, c'est-à-dire pour plusieurs trous. On définit alors une variable `temps_20_trous` qui sera la différence de temps entre la fin et le début d'une boucle, et cette boucle sera simplement un compteur de 20 trous, c'est-à-dire de 20 fois un passage de lumière.



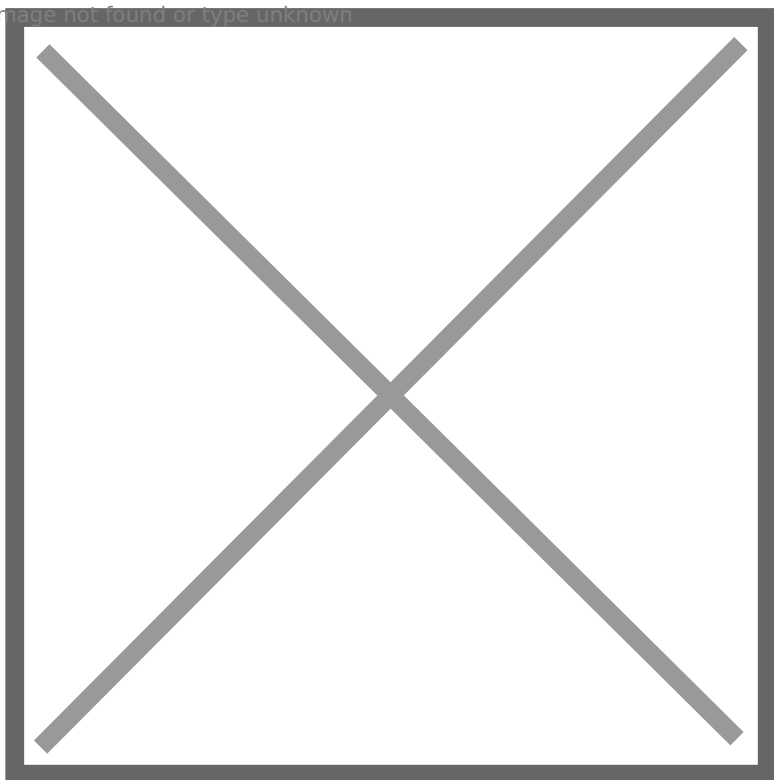
Maintenant il nous faut relier ce temps\_20\_trous au débit correspondant. Pour cela, on fait l'hypothèse de proportionnalité, comme on l'a dit auparavant. On affiche plusieurs fois le temps\_20\_trous d'un débit connu, qu'on a créé avec une jarre disposant d'un robinet. Alors, on a un temps\_20\_trous\_ref qui correspond à un debit\_ref. Si ensuite on mesure un autre le temps\_20\_trous d'une autre écoulement, quel qu'il soit, avec une règle de 3 on pourra le relier à la



valeur de son débit, et l'afficher.

*Partie voidsetup():* Pour finir, il nous faut une façon de déclencher notre boucle: on veut commencer à la faire tourner qu'au moment qui nous intéresse. On décide qu'au début, quand on ne veut pas faire de mesure, la lumière n'est pas allumée: de cette façon, la tension est toujours supérieure à la tension de seuil, on reste dans la boucle du voidsetup() et on affiche "Pas de débit". Quand on veut faire une mesure, on allume la lumière, mais on la positionne la roue de telle sorte que la lumière soit cachée: on ne sort toujours pas de la boucle, mais on est prêts. Quand on commence à verser de l'eau, la photorésistance reçoit une première fois une lumière intense, la tension passe en dessous de la tension de seuil et on sort de cette boucle, on reste désormais dans la boucle du voidloop() jusqu'à la fin, et on affiche un débit.

Image not found or type unknown



---

Revision #2

Created 13 February 2023 16:14:06 by Turcios Maya

Updated 13 February 2023 16:22:11 by Turcios Maya