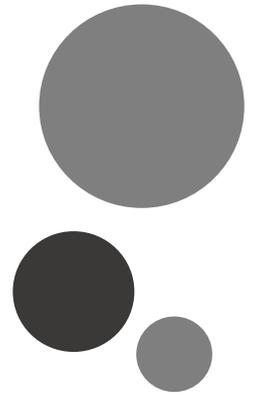


# Cahier Des Charges Technique



Composants

Programmation

CAO

Comment ça marche ?

Gasser Arthur

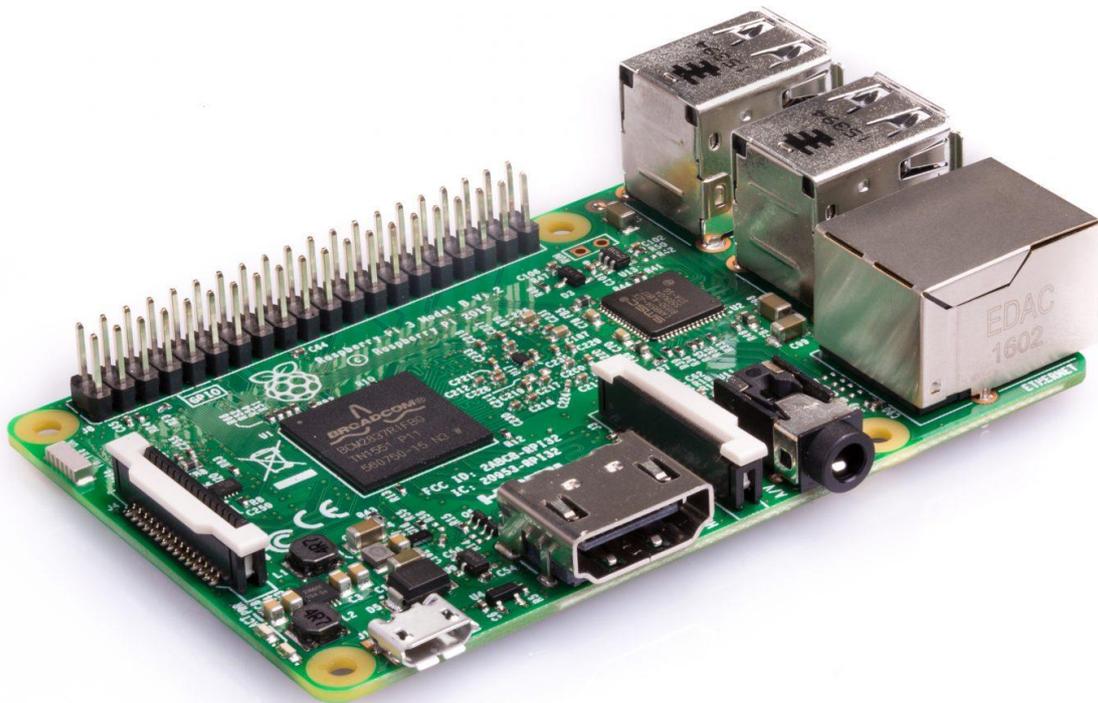
Dijoux Quentin

Jimenez Raphaël

# Composants

## Raspberry Pi 3 modèle B

C'est le cœur de notre microscope, la carte contrôle l'écran, les leds ainsi que la caméra. L'entièreté de la programmation de notre microscope a été réalisée sur notre Raspberry.



Voici les informations détaillées de notre carte :

### Caractéristiques

Processor	Broadcom BCM2387 chipset. 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
GPU	Dual Core VideoCore IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory	1GB LPDDR2
Operating System	Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V1, 2.5A

## Connectiques

Ethernet	10/100 BaseT Ethernet socket
Video Output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audi Output	Audio Output 3.5mm jack, HDMI USB 4 x USB 2.0 Connector
GPIO Connector	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
Memory Card Slot	Push/pull Micro SDIO

## Avantages

- Peu chère
- Rapide
- Bon format de la carte
- Possibilité d'ajouter des connectivités

## Applications

- PC/Tablette/PC Portable peu chères
- Industrie
- Domotique
- Caméra web
- Jeux
- Point d'accès sans fil
- Serveur
- ...

## Ecran LCD

Nous disposons d'un écran tactile pour utiliser le Raspberry. Cela est très pratique car nous n'avons pas besoin d'être à proximité d'un écran d'ordinateur par exemple, il nous faut juste une alimentation et nous pouvons interagir avec le Raspberry. Cependant, travailler sur un écran plus grand reste plus confortable sur le long terme.



Voici les informations détaillées de notre écran :

### Caractéristiques

- Ecran tactile capacitif multipoint : prise en charge jusqu'à 10 doigts
- Ecran 7 pouces tactile LCD
- Résolution 800 x 480 pixels à 60 images par secondes (ips)
- Couleur RGB 24 bits
- Angle de vue de 70°
- Fonctionnalité d'OS Raspbian complète
- Carte adaptateur
- Câble en nappe DSI
- 4 fils volants

## Avantages

- Deux connexions requises seulement
- Alimentation flexible
- Aucune interférence électronique

## RaspiCam noir V2 + Carte SD 16Go

Nous utilisons cette carte pour capturer nos images. Cette caméra est de très bonne qualité mais pour notre application, comme le nom de notre projet l'indique, nous avons dû décrocher la lentille présente sur la caméra.



Voici les informations détaillées de notre caméra :

### Caractéristiques

Type	Module appareil photo
Nombre de voies	1
Interfaces bus supportées	CSI-2
Résolution maximum supportée	3280 x 2464
Cadence maximum capture	30 fps
Dimensions	23.86 x 25 x 9mm
Température de fonctionnement maximum	60°C
Température de fonctionnement minimum	-20°C
Mise au point fixe	8 mégapixels
Format	1080p, 720p60, VGA90
Capteur d'image	CMOS Sony IMX219PQ
Câble	Câble en nappe 15 broches

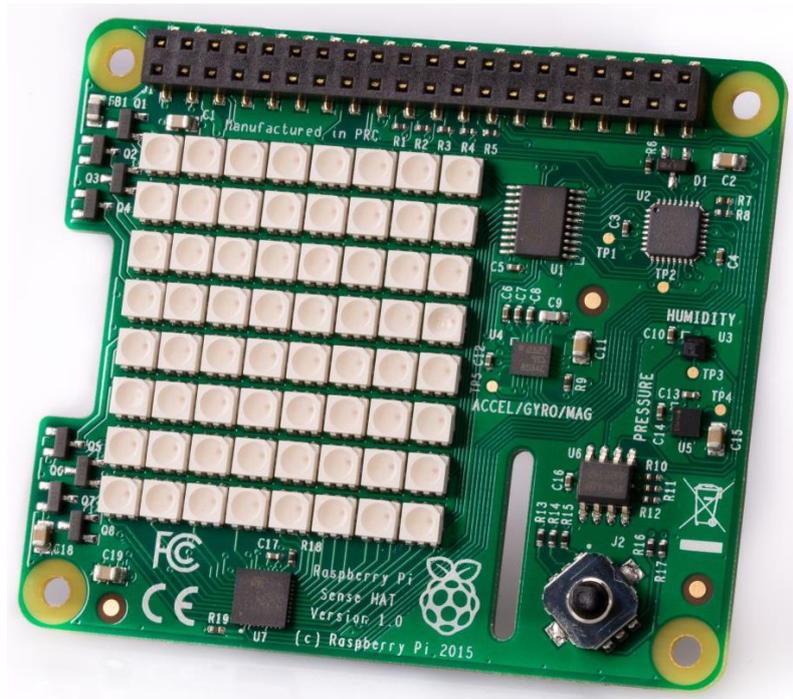
### Avantages

- Sans filtre IR
- Images de haute qualité
- Haute capacité de données

Etant donné que nous travaillons avec des images (captures et traitements) nous avons également besoin d'une carte SD de 16Go.

## Sense HAT (module Leds)

Nous avons besoin d'éclairage afin de voir l'échantillon qui sera présent dans notre microscope. Nous avons donc pensé à utiliser un module de Leds compatible avec Raspberry. Ces Leds sont pratiques à utilisées et assez simples à programmer.



Voici les informations détaillées de notre Sense HAT :

### Caractéristiques

Gyroscope	245/500/2000dps
Accéléromètre	2/4/8/16g
Magnétomètre	4/8/12/16 gauss
Baromètre	260 – 1260 hPa
Capteur de température	0°C – 65°C
Capteur d'humidité	4.5% - 20/80%rH
LED	Matrice 8 x 8
Joystick	5 boutons

### Avantages

- Polyvalence
- Directement contrôlé par le Raspberry

# Programmation

## Introduction

La programmation de notre microscope est réalisée en Python 3. Avant ce projet, nous n'avions aucune connaissance en Python et en Linux, nous avons donc dû nous adapter et apprendre ces différents langages. Nous souhaitons que notre microscope puisse être utilisé par tout le monde, c'est pour cela que nous avons annoté chacun de nos programmes. Nous utilisons trois programmes principaux et nous allons détailler leur contenu.

## Détails

### (1) Acquisition

Dans ce programme, vous pouvez gérer la caméra ainsi que les leds. Vous pouvez démarrer la prévisualisation de la caméra afin d'observer ce que vous voulez étudier, vous pouvez prendre votre photo ou arrêter la caméra. Au niveau des leds, vous pouvez les éteindre ou tout simplement régler l'intensité lumineuse comme vous le souhaitez mais nous vous conseillons de la régler à 100%. Si vous êtes satisfaits de votre capture vous pouvez passer à la suite.

### (2) Reconstruction

La capture étant réalisée, il faut maintenant améliorer le rendu de celle-ci. Pour cela veuillez cliquer sur le bouton « Reconstruction 2D » afin de lancer le programme correspondant. Ce dernier va réaliser une mise au point numérique à l'aide d'un recalage de pixel. Pour cela, il faut utiliser plusieurs FFT (Fast Fourier Transformer). Ces différents termes peuvent paraître compliqués mais en visualisant la photo une fois reconstruite vous allez très vite comprendre son impact sur l'image que vous avez prise précédemment. Une fois la reconstruction terminée vous pouvez analyser votre image.

### (3) Analyse

Pour ouvrir ce programme il vous suffit d'appuyer sur le bouton correspondant. Une fois le programme lancé, vous pouvez réaliser un certains nombres d'actions :

- Contouring
- Zoom
- Déplacement
- Annotation
- Sauvegarde
- ...

Pour accéder à ces différentes actions, il vous suffit de lire les instructions présentes sur votre écran.

## Conclusion

Malgré le manque de connaissance, nous nous sommes adaptés très rapidement à l'univers Raspberry. Nous avons réussi à créer un programme assez simple d'utilisation et nous avons atteints nos objectifs. Cependant, si nous avions eu plus de temps, nous aurions aimé programmer une « Reconstruction 3D » de nos images.

Nos différents codes ne seront pas écrits sur ce document mais vous pouvez les retrouver sur notre site internet :

<http://www.projetsgeii.iutmulhouse.uha.fr/microscopie-sans-lentille/>

(certains programmes sont en double car nous avons réalisé une version utilisable sur le Raspberry et une autre utilisable sur moniteur).

## Introduction

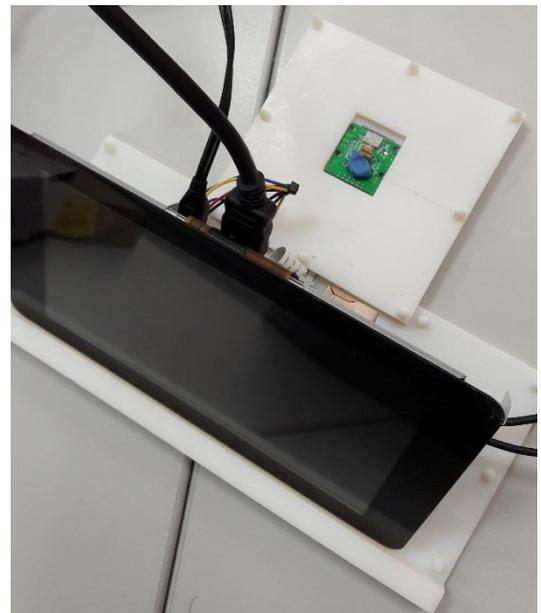
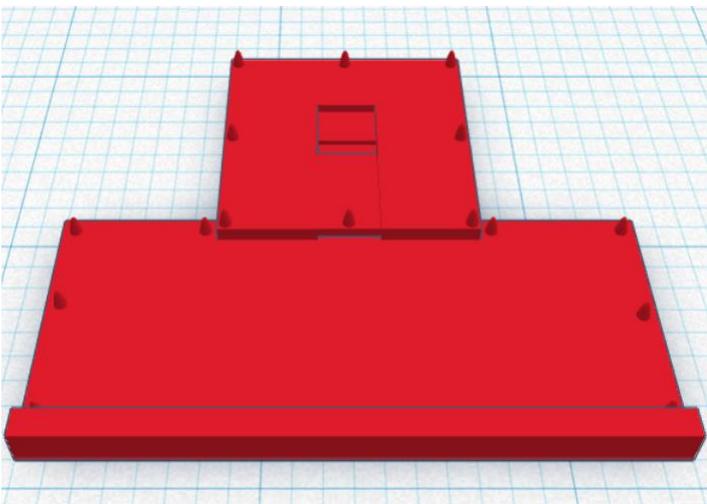
L'un des points principaux de notre projet était de réaliser une CAO (Conception Assistée par Ordinateur) pour créer notre microscope. Pour cela, nous avons utilisé le logiciel Tinkercad, qui était le plus pratique dans notre cas car c'est un logiciel en ligne, nous pouvions donc travailler chez nous et sur tous les PC de l'IUT (voici un lien vers le site [Tinkercad](#)).

## Détails

Pour une question de praticité, nous avons décidé de réaliser notre microscope en plusieurs pièces pour faciliter le montage ainsi que la maintenance. Dans cette partie, nous allons vous présenter chacune des pièces en détail.

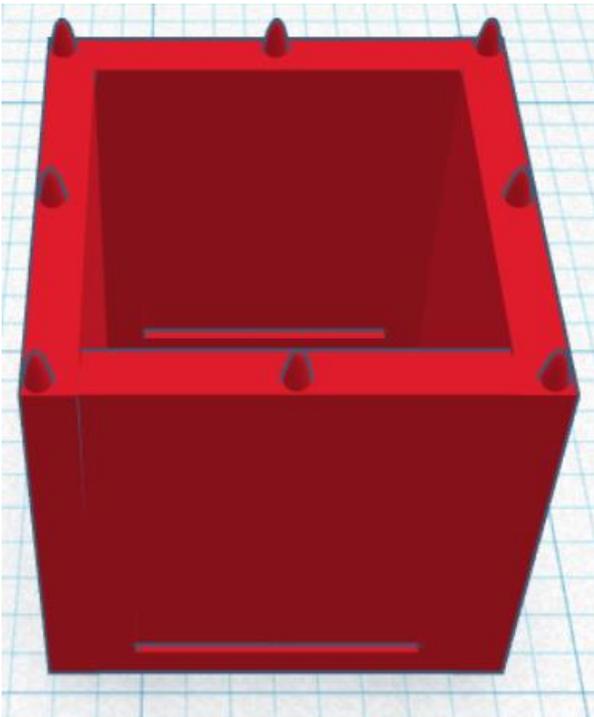
### (1) Socle

Commençons par la base de notre microscope. Ce socle a été conçu afin que la caméra et surtout le capteur soit au milieu du microscope. Il y a également un chemin de câble qui passe en dessus pour que la nappe de la caméra soit protégée et pour la connecter directement au Raspberry de manière efficace. Le Raspberry et l'écran sont donc placés sur la partie avant du socle (la plus grande) pour des questions de confort.



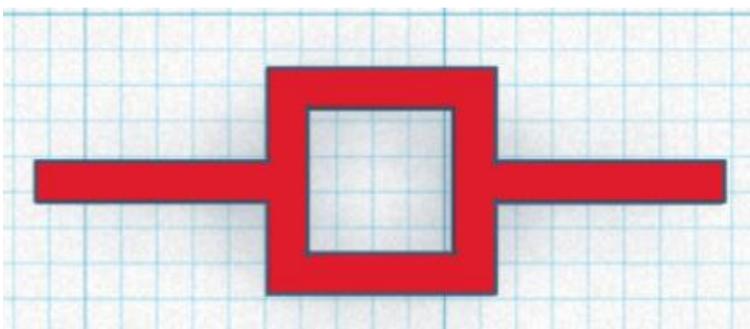
## (2) Murs pour l'échantillon

Sur ce socle, vient la partie qui permettra d'accueillir l'échantillon. Cette pièce est constituée de 4 murs et sur deux de ces murs, il y a une fente permettant d'introduire l'échantillon dans le microscope à l'aide de la pièce que nous allons voir tout de suite.



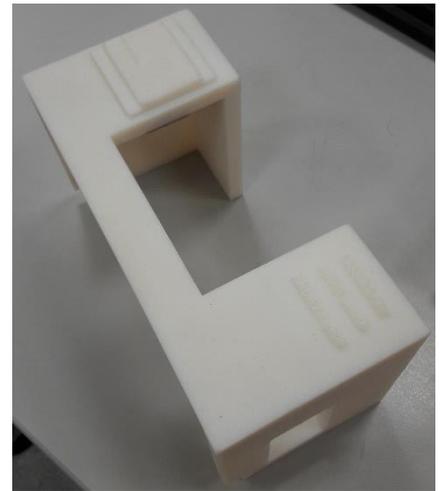
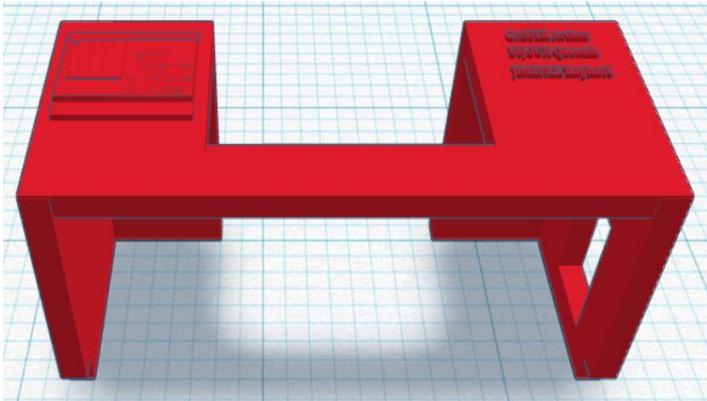
## (3) Support échantillon

On place l'échantillon sur cette pièce puis on l'introduit dans le microscope. Tout a été étudié pour que le support soit maintenu par les deux fentes vues précédemment. Cette pièce est de cette forme car l'échantillon ainsi que la lumière doivent arriver jusqu'au capteur située juste en dessous.



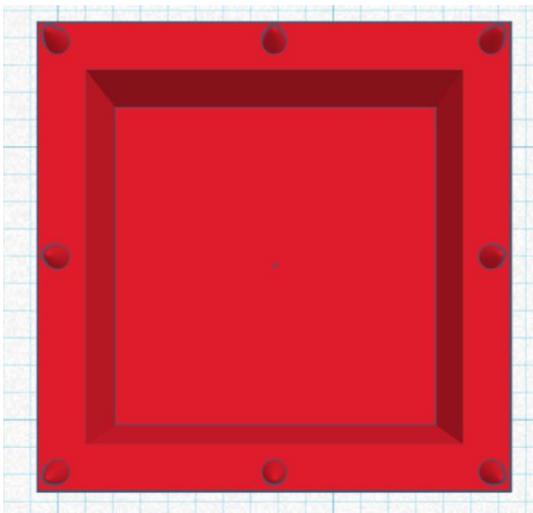
#### (4) Protection

Pour que notre écran et notre Raspberry soient protégés contre les chutes. Nous avons réalisé cette pièce permettant de les caler. Sur cette pièce nous avons également réalisé deux trous pour laisser passer les câbles provenant du Sense Hat et de l'alimentation mais aussi pour brancher le clavier, la souris et un câble HDMI. Nous avons aussi placé le logo de l'IUT ainsi que nos noms sur le dessus de la pièce.



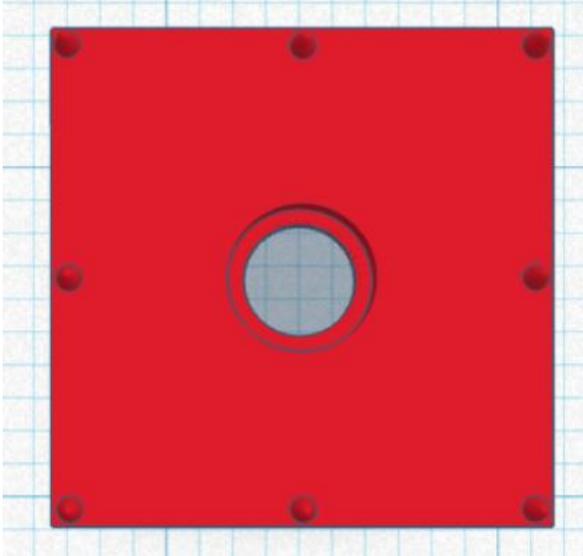
#### (5) Pinhole

Cette pièce est entièrement pleine. Mais en son centre, il y a le trou appelé « pinhole » permettant de laisser passer la lumière. Sa taille a été étudiée afin que le phénomène de diffraction soit respecté, nous verrons les détails des phénomènes physiques dans la dernière partie de ce document.



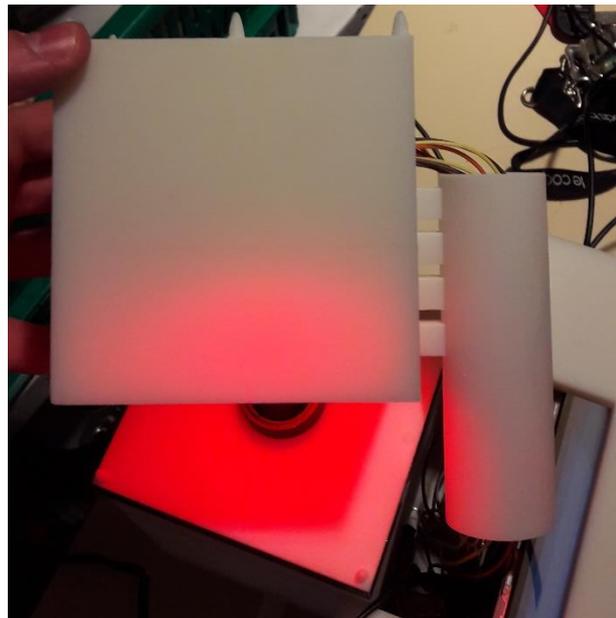
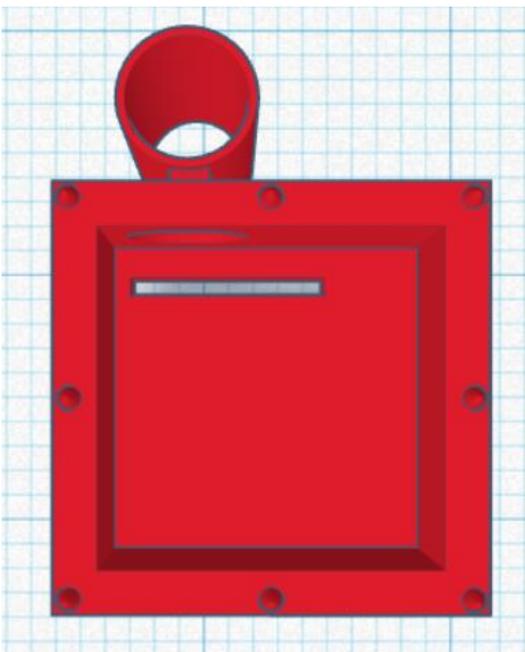
## (6) Focale

Afin que le flux de lumière soit plus conséquent, nous avons ajouté une focale pour concentrer la lumière en un seul point. Cette pièce est donc constituée d'un trou permettant de fixer parfaitement la focale au centre.



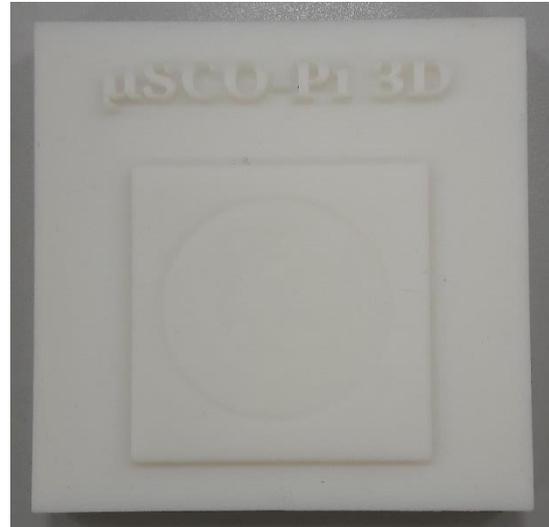
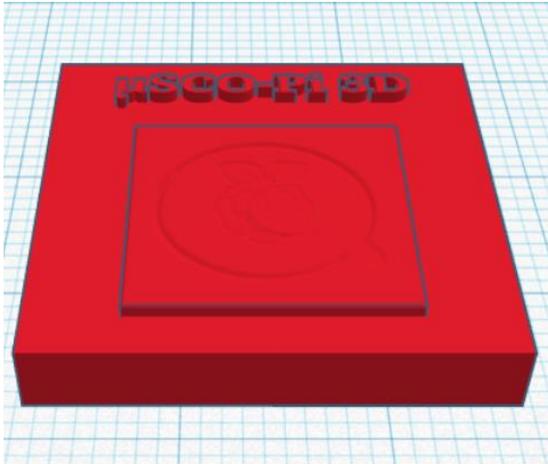
## (7) Leds + Tuyau

Notre module de Leds sera fixé sur cette pièce, nous avons percé un trou afin de laisser passer les câbles du Sense Hat. Ces câbles passeront ensuite dans le tuyau et descendront jusqu'au Raspberry.



## (8) Toit

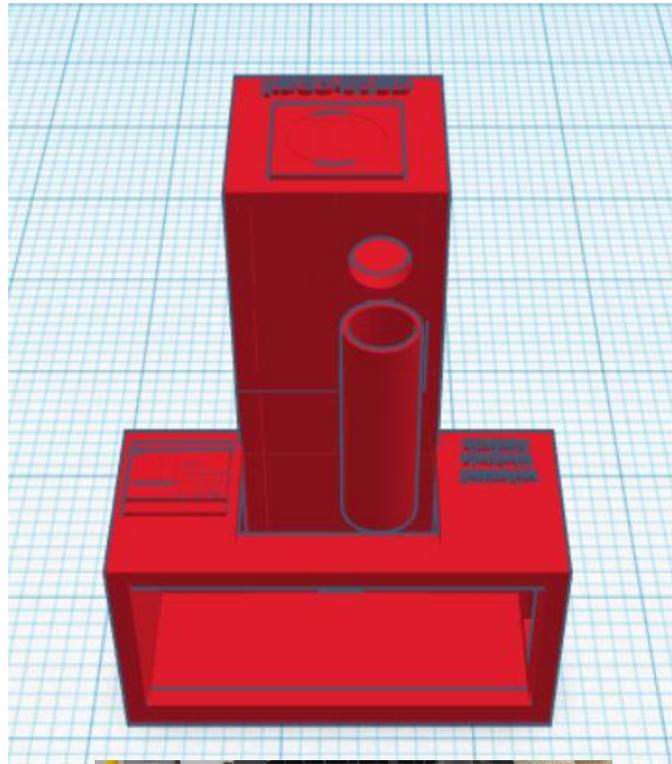
Voici la dernière pièce de notre microscope. Il s'agit d'un cube permettant de refermé le montage. Sur cette pièce nous pouvons voir le nom de notre microscope «  $\mu$ SCO-Pi 3D » ainsi que notre logo.



## Conclusion

La CAO était une étape très importante du projet, nous avons donc consacré beaucoup de temps pour la réaliser de la manière la plus précise, esthétique et ergonomique. Cependant, le rendu final présentait quelques erreurs de précision que nous avons tenté de corriger au mieux. Nous avons dû peindre notre microscope en noir pour empêcher la lumière extérieure de gêner notre acquisition. Nous avons utilisé l'imprimante 3D (résine) de l'IUT Lab, nous souhaitons donc remercier Mr De Sabbata pour son aide.

Voici le rendu final de notre microscope :



Microscope sans lentille

# Comment ça marche ?

## Introduction

Pour obtenir une image zoomée avec ce type de montage, il faut prendre en compte différents phénomènes physiques et optiques. Nous avons dû apprendre ces quelques notions pour dimensionner notre microscope. Dans cette partie, nous allons vous présenter nos recherches.

## Diffraction

Qu'est-ce que la diffraction ?

C'est un passage d'ondes linéaires dans un trou de petite taille (pinhole ; 100 $\mu$ m) qui font se transformer en ondes circulaires.

Pourquoi l'utilise-t-on ?

Ce phénomène permet d'augmenter les interférences qui conduisent à la projection de l'échantillon sur le capteur CMOS présent en aval, on obtient donc un hologramme.

Quels sont les paramètres à prendre en compte ?

- Distance Pinhole-Capteur notée **L** (ici 10cm)
- Distance Echantillon-Capteur notée **Z** (ici 1cm)
- Longueur de côté du capteur notée **W** (ici 512 (*nb pixel par côté*)  $\times$  1.4 (1 *pixel*)) = 716.8 $\mu$ m)
- Longueur d'onde notée  **$\lambda$**  (ici rouge donc environ 600nm)
- Ouverture du capteur (intervient dans l'apparition des images jumelles)
- Largeur du pinhole (calibre la cohérence des ondes circulaires) doit être de l'ordre de la longueur d'onde

Nous pouvons donc calculer la résolution de l'image :  $dx = \lambda * \frac{L-z}{W}$

Nous obtenons une résolution de 830 $\mu$ m ce qui n'est pas énorme mais suffisant pour notre application. Nous aurions pu obtenir une meilleure résolution mais nous aurions eu trop de contrainte au niveau de la CAO.

## Traitement d'image

L'image reçue sur le capteur comporte beaucoup d'interférence, si bien qu'il est difficile de distinguer clairement les contours de l'objet observé. Pour cela nous faisons appel à une méthode de reconstruction d'image. La reconstruction d'image est représentée mathématiquement par

l'équation suivante :  $k(r) = \int_{screen} \tilde{I}(\xi) e^{ikr^* \frac{\xi}{|\xi|}} d^2\xi$  avec :

- $\tilde{I}(\xi)$  contraste de l'image
- $\xi$  correspond à des coordonnées (X, Y, L) de l'image reconstruite
- $k = \frac{2\pi}{\lambda}$

Pour obtenir cette fonction numériquement, on utilise un algorithme qui transforme l'exponentielle (qui est non linéaire) en produit de convolution qui peut être résolu plus aisément par un ordinateur. En effet, il suffit d'appliquer trois FFT (transformées de Fourier) pour résoudre ce produit de convolution et ainsi connaître la valeur de la fonction  $k(r)$ .

Des approximations réduisant la lourdeur des FFT sont possibles mais au prix d'une grosse perte de précision. Nous n'avons pas opté pour cette solution car notre précision était déjà restreinte.

## Conclusion

Certaines notions étaient plutôt compliquées à comprendre. Mais grâce à l'aide de Mr Verrier, que nous remercions, nous avons réussi à estimer/calculer nos valeurs afin de réaliser notre microscope de la manière la plus précise possible.

Voici un schéma de synthèse de notre microscope :

