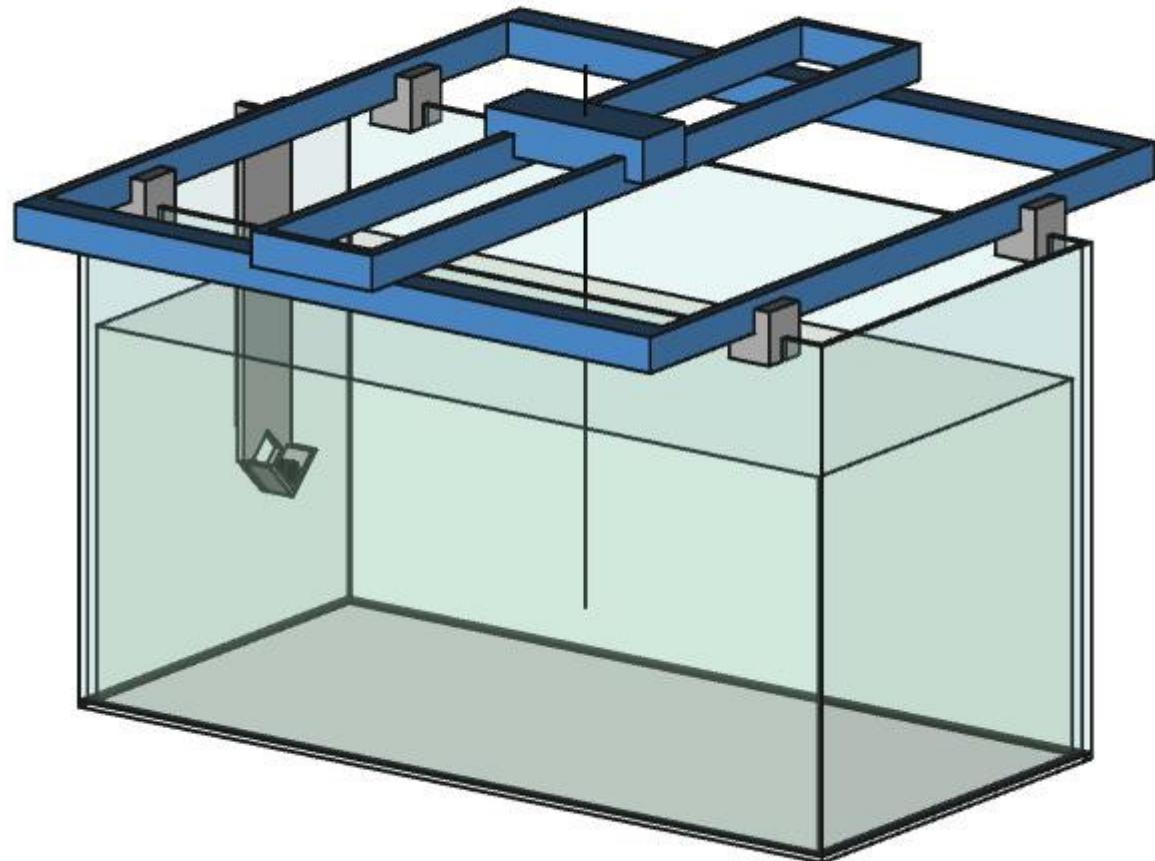


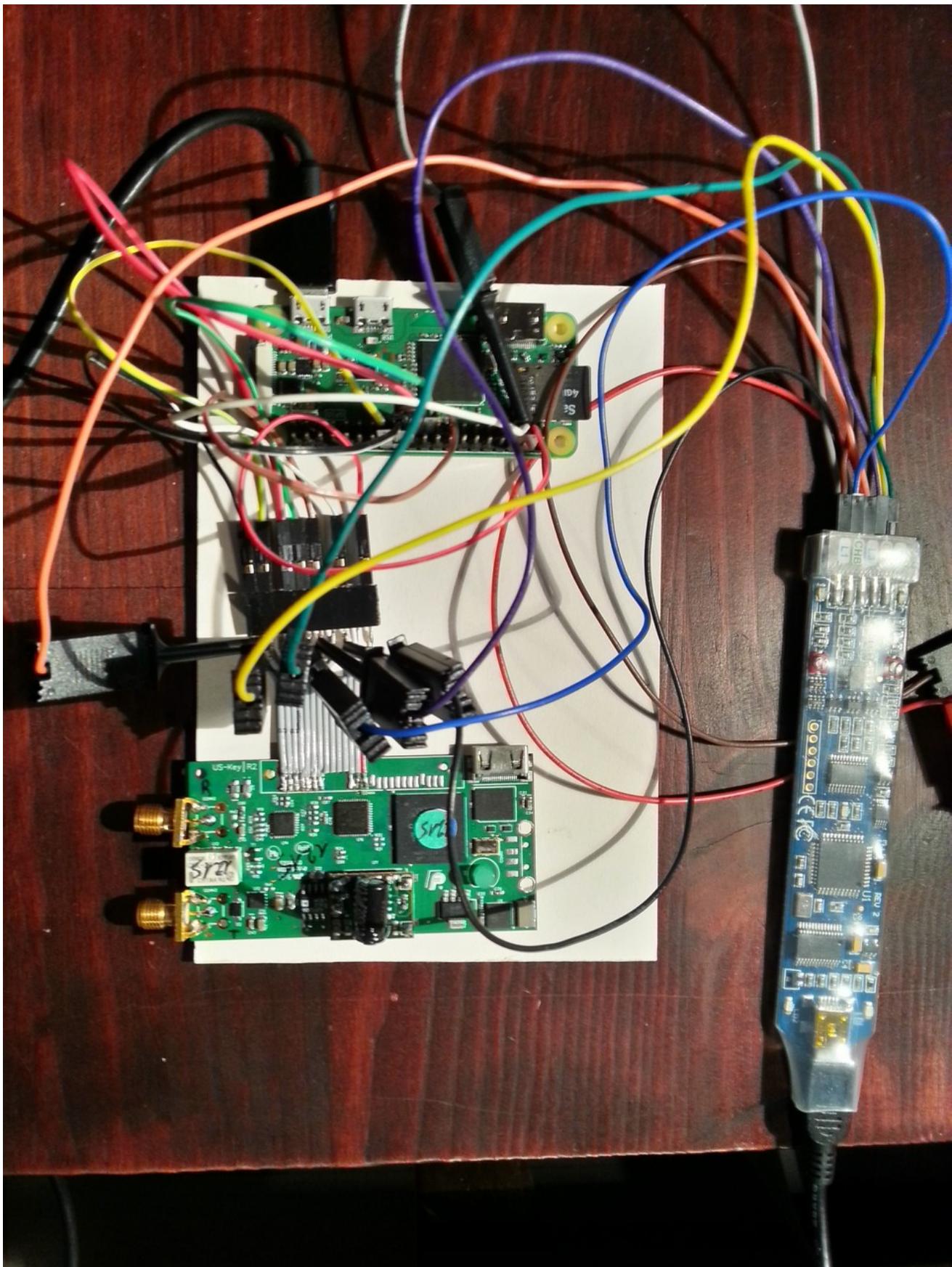
echOpen - Project log : EchoBench developpment

Daily log of the echOpen transducers test bench

Objet :	This document will show the project developpment progression with many photos and some comments.
Réf/File :	2017-09-03 PRJLOG echobench_developpment_log_en
Révision :	v1 - 2017-09-03 - BVi - Creation
Révision :	work in progress



2017-09-10 Assemblage Raspberry Pi Zero W & US-SPI



Installation RPi zero W

Installation Raspbian

Config wifi en mode headless

D'après <https://core-electronics.com.au/tutorials/raspberry-pi-zerow-headless-wifi-setup.html>

Il suffit d'ajouter un fichier `wpa_supplicant.conf` à la carte SD de boot. Il sera "consommé" au prochain boot pour surcharger le fichier de configuration Wifi `/etc/wpa_supplicant/wpa_supplicant.conf`. Il doit contenir :

```
1 network={  
2     ssid="Your SSID"  
3     psk="YourWPAPassword"  
4     key_mgmt=WPA-PSK  
5 }
```

Cette opération est à reproduire à chaque changement d'access point Wifi. On peut mettre plusieurs config :

```
1 network={  
2     ...  
3 }  
4  
5 network={  
6     ...  
7 }
```

Config mode "Ethernet Gadget"

D'après <https://learn.adafruit.com/turning-your-raspberry-pi-zero-into-a-usb-gadget/ethernet-gadget>

- Connexion SSH depuis windows
- On peut scanner et tester la connexion Wifi avec :

```
1 ifconfig  
2 iwconfig wlan0  
3 sudo iwlist wlan0 scan
```

Maj système :

```
1 sudo apt-get update  
2 sudo apt-get upgrade  
3 sudo rpi-update  
4 sudo apt-get autoremove
```

Partage répertoire de travail avec Samba !!! ne fonctionne pas !!!

D'après <http://www.framboise314.fr/partager-un-repertoire-sous-jessie-avec-samba/>

- Installation serveur Samba

```
1 sudo apt-get install samba samba-common-bin
```

- Faites une copie de sauvegarde du fichier de configuration

```
1 | sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.org
```

- Editer le fichier `/etc/samba/smb.conf`

```
1 | # The specific set of interfaces / networks to bind to
2 | # This can be either the interface name or an IP address/netmask;
3 | # interface names are normally preferred
4 | interfaces = 127.0.0.0/8 eth0
5 | bind interfaces only = yes
```

- Créez le dossier que vous souhaitez partager (s'il n'existe pas). J'ai choisi `/home/pi`.
- Maintenant rendez vous à la fin du fichier... oui, tout en bas et ajoutez :

```
1 | [pihome]
2 | comment = code folder
3 | path = /home/pi
4 | writeable = yes
5 | browseable = yes
6 | read only = no
7 | guest ok = yes
8 | guest only = yes
9 | create mask = 0755 ???
10 | create mode = 0777
11 | directory mode = 0777
12 | share modes = yes
```

- Définir le mot de passe réseau pour `pi`

```
1 | sudo smbpasswd -a pi
```

- Redémarrez Samba sous (Jessie):

```
1 | sudo systemctl restart smbd.service
```

Configuration Wifi

A défaut de pouvoir faire fonctionner SAMBA sur ethernet over USB on configure le wifi d'après <https://core-electronics.com.au/tutorials/raspberry-pi-zero-w-headless-wifi-setup.html> :

- Sous SSH éditer le fichier `/boot/wpa_supplicant.conf` en y insérant (le choix est fait en dans l'ordre et en fonction des puissances :

```

1 network={
2     ssid="Bivi's2-5G"
3     psk="...password..."
4     key_mgmt=WPA-PSK
5 }
6 network={
7     ssid="Bivi's2"
8     psk="...password..."
9     key_mgmt=WPA-PSK
10 }
11 network={
12     ssid="Bivi's"
13     psk="...password..."
14     key_mgmt=WPA-PSK
15 }

```

Avec cette config, en rebootant, la freebox attribue l'IP 192.168.1.120 au RPi et le partage pihome fonctionne correctement avec l'explorateur de fichier Windows sous \192.168.1.120\pihome

Installation Python3 et lib

- Installer Pip3

```

1 sudo apt-get install python3-pip
2 sudo python3 -m pip install --upgrade pip

```

- Installer bottle, le serveur http et websocket pour Python3

```

1 sudo pip3 install bottle
2 sudo pip3 install python-dev !!! ECHEC !!!
3 sudo pip3 install gevent-websocket

```

- Tester bottle en exécutant les fichier `python3 bottle_hello.py` :

```

1 from bottle import route, run, template
2
3 @route('/hello/<name>')
4 def index(name):
5     return template('<b>Hello {{name}}</b>!', name=name)
6
7 run(host='0.0.0.0', port=8080)

```

- Installer l'accès au bus SPI

D'après <http://deusyss.developpez.com/tutoriels/RaspberryPi/PythonEtLeGpio/>

- Lancer sudo raspi-config et valider le SPI dans Advanced options :

```

1 sudo raspi-config

```

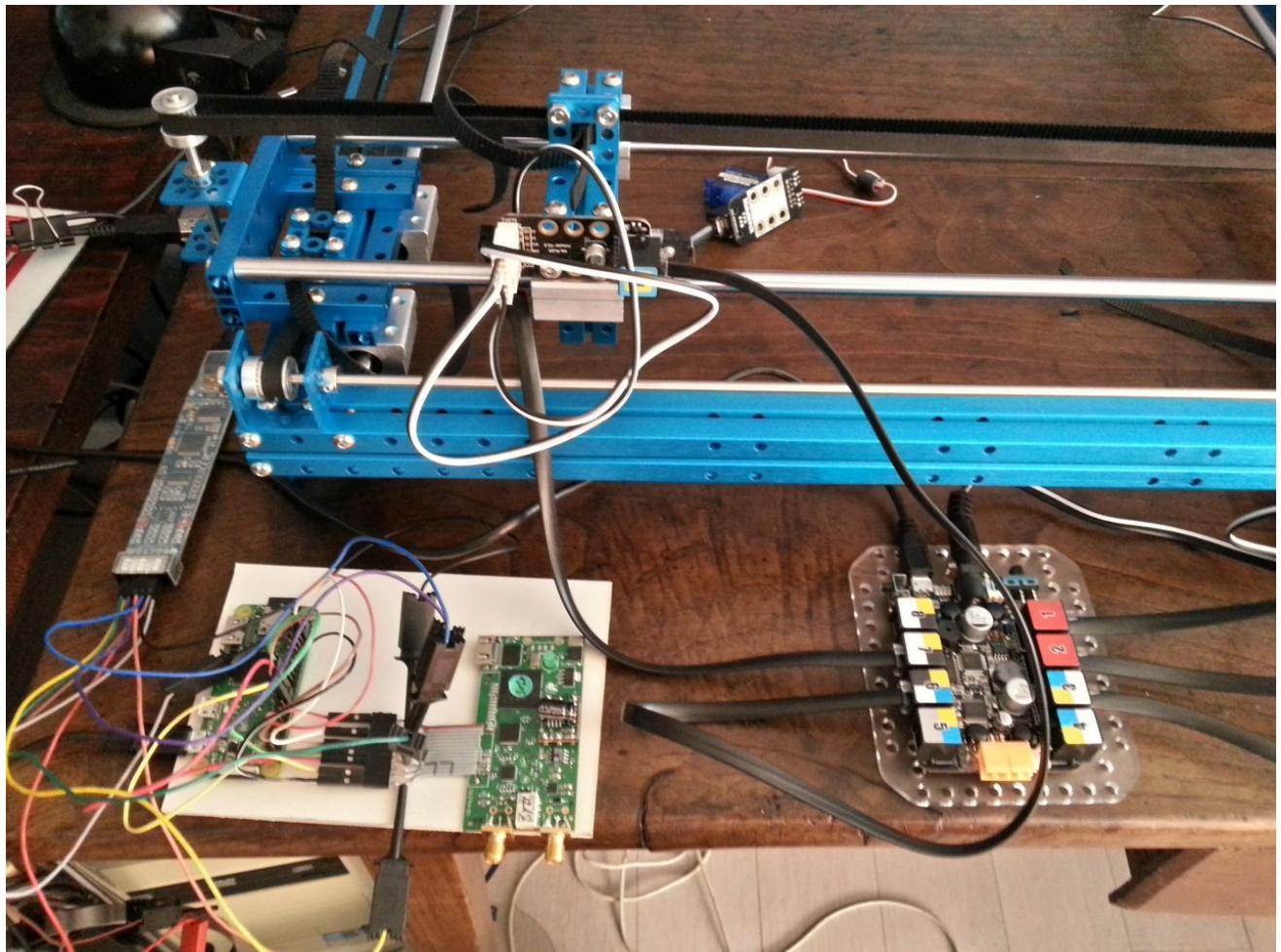
- Après reboot, tester la présence du driver SPI :

```
1 | lsmod | grep spi_
```

- Installation modules python pour SPI

```
1 | sudo pip3 install spidev
```

2017-09-15 Montage Table XY

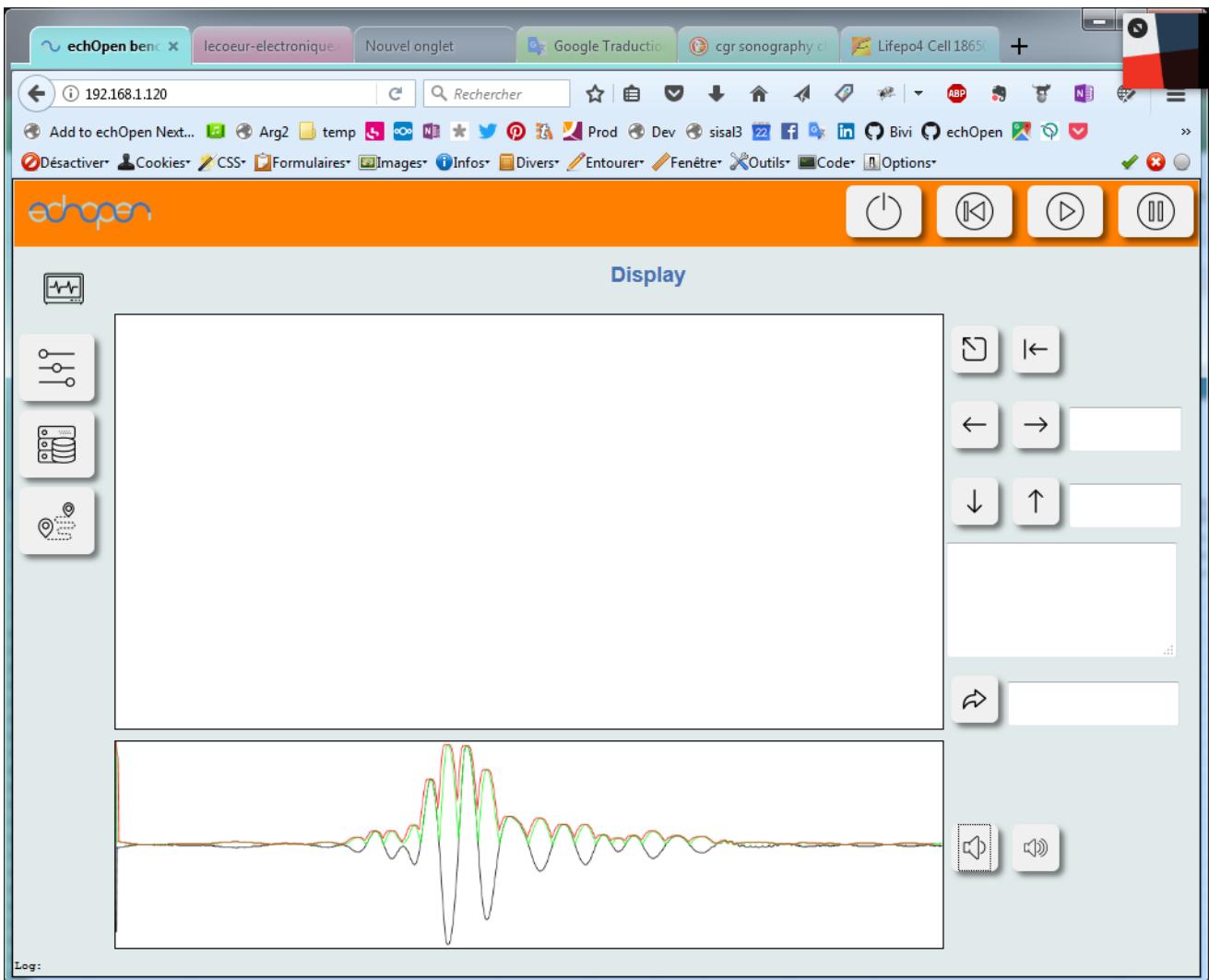


2017-10-06 Premiers essais

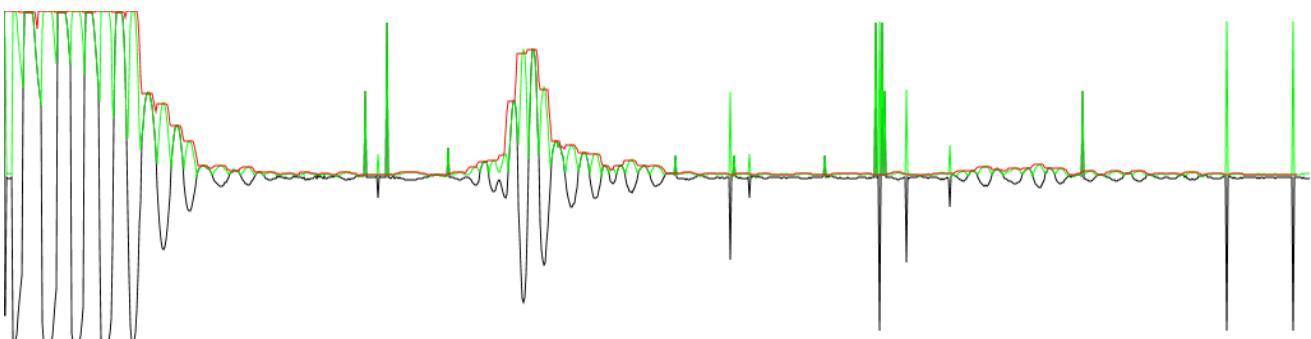
Fantôme de pôche ;-)



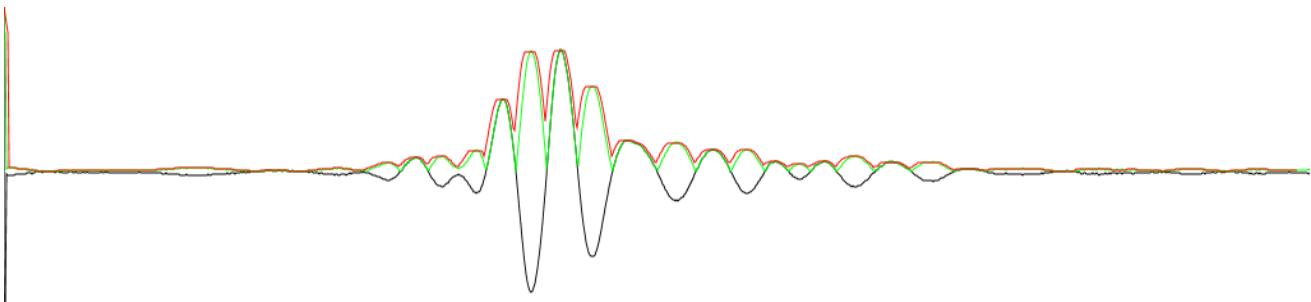
IHM sur navigateur Web



Echantillonnage à 20Msps



Echantillonnage à 160Msps



2017-11-05 Adaptation Table XY sur aquarium (ou jardinière)

Connexion Table XY <-> RPi0

On va établir une connexion série mais **attention** les niveau logiques sont à 3.3V côté RPi et 5V côté carte Me Orion (Arduino) de MakeBlock. Il faut un adaptateur de niveau :

- Le "Me Shield for Raspberry Pi" de chez MakeBlock <http://learn.makeblock.com/en/me-shield-for-raspberry-pi/> (en stock chez Roboshop <http://www.robotshop.com/en/makeblock-me-shield-raspberry-pi.html>)
- Ou un adaptateur de niveau std <https://hackspark.fr/fr/3-3v-5v-logic-level-converter.html> ou <https://hackspark.fr/fr/8-channel-bi-directional-logic-level-converter-txb0108-3-3-5v-or-others.html>
- On évitera l'adaptation de niveau par pont diviseur qui présente un danger pour le T car la pin RX côté Arduino peut rester au niveau 5V dans certains cas.
Un article complet sur la connexion Rpi Arduino : <https://oscarliang.com/raspberry-pi-and-arduino-connected-serial-gpio/>

Pilotage de la table par le RPi

Au lieu d'utiliser les logiciels de dessin de haut niveau fournis avec la table (mDraw ou BenBox), nous allons installer sur la carte Me Orion de la table le firmware de pilotage CNC GCodeParser :

<https://github.com/Makeblock-official/XY-Plotter-2.0>. Il interprète habituellement les commandes envoyées par le logiciel PC GRemote mais dans notre cas nous enverrons directement les commandes depuis un script Python (serveur HTTP Bottle) via le port série.

Codes G

```
1 G0 Xnnn Ynnn Znnn (Avance rapide)
2 G1 Xnnn Ynnn Znnn Fnnn (Avance usinage)
3 G2 Xnnn Ynnn Znnn Innn Jnnn (arc sens horaire)
4 G3 Xnnn Ynnn Znnn Innn Jnnn (arc sens anti-horaire)
5 G4 Pnnn (Pause)
6 G20 (Inch)
7 G21 (mm)
8 G28 (Home 0,0,0)
9 G30 Xnnn Ynnn Znnn (Home via XYZ)
10 G90 (Abs positionning)
11 G91 (Incr positionning)
12 G92 (Set as home)
13 $1 Xnnn Ynnn Znnn (set XYZ STEP PIN)
14 $2 Xnnn Ynnn Znnn (set XYZ DIR PIN)
15 $3 Xnnn Ynnn Znnn (set XYZ Min PIN)
16 $4 Xnnn Ynnn Znnn (set XYZ Max PIN)
17 $5 Znnn (ENABLE SERVO MOTOR FOR Z)
18 $6 Xnnn Ynnn Znnn (set XYZ STEPS PER MM)
19 $7 Xnnn Ynnn Znnn (set XYZ fast FEEDRATE)
20 $8 Snnn (set XYZ INVERT LIMIT SWITCH)
```

Usage NetBeans

Pour permettre à Netbeans d'ouvrir des projets quelconques (juste une arborescence de fichier, sans fichier projet)

- Installer le plugin "Automatic Projects"

Install Makeblock Library for Arduino

D'après <http://learn.makeblock.com/en/learning-arduino-programming/>

Liaison Série RPi0 <-> Arduino

D'après <http://learn.makeblock.com/me-shield-for-raspberry-pi/> D'après <https://electrosome.com/uart-raspberry-pi-python/>

- Libérer /dev/ttyAMA0 en désactivant le Bluetooth (cf <https://www.raspberrypi.org/documentation/configuration/uart.md>, <https://hallard.me/enable-serial-port-on-raspberry-pi/>). Depuis que les RPi sont équipés de Bluetooth (RPi3 et RPi0 W), l'UART hard est directement connecté au BT et un UART soft /dev/ttys0 est connecté au GPIO 14 et 15. Il faut ici le récupérer

```
1 sudo nano /boot/config.txt
2 ... ajouter à la fin la ligne:
3 dtoverlay=pi3-disable-bt
4 ... sauver/quitter
5 sudo systemctl disable hcuart
6 ... rebooter
7 sudo shutdown -r now
```

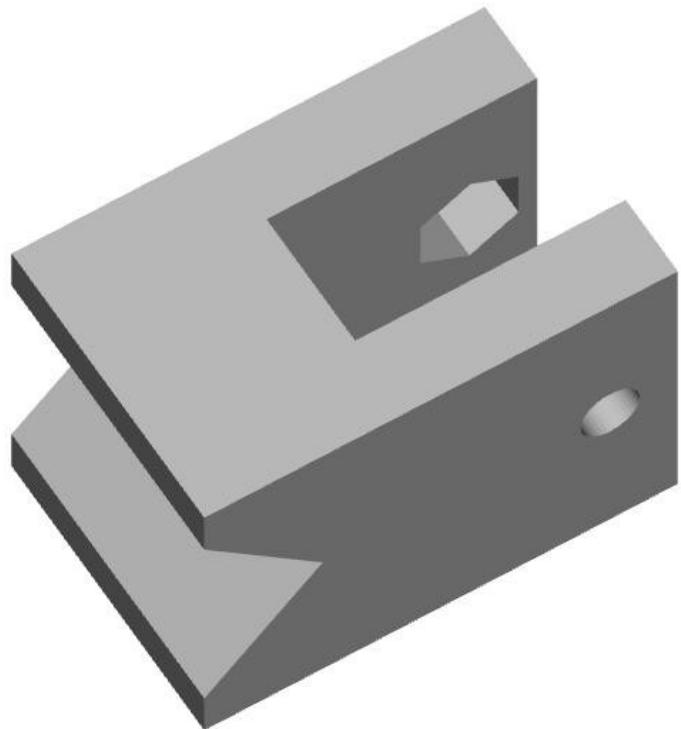
- Installer pyserial

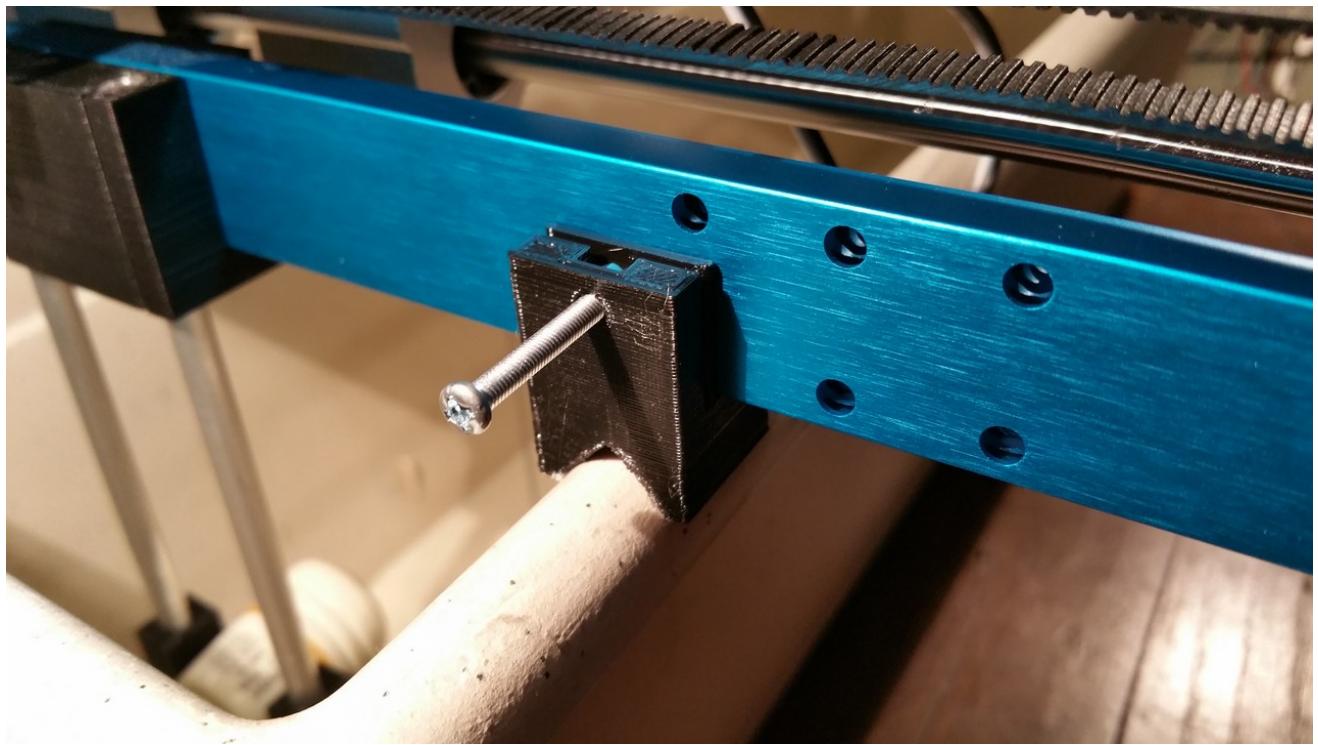
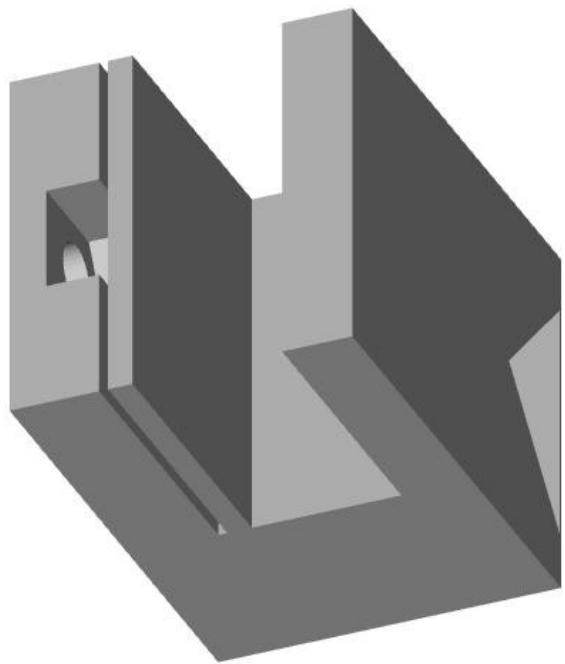
```
1 sudo apt-get install python-serial
2 sudo pip3 install pyserial
```

- Reboucler RX/TX et tester :

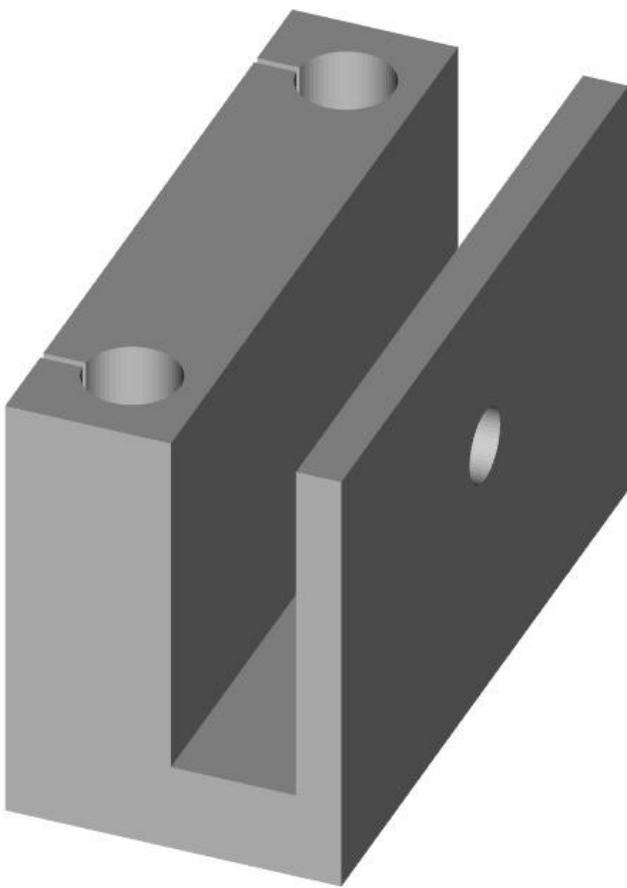
```
1 python -m serial.tools.miniterm /dev/ttys0 115200
2 ... ou
3 minicom -b 115200 -o -D /dev/ttys0
```

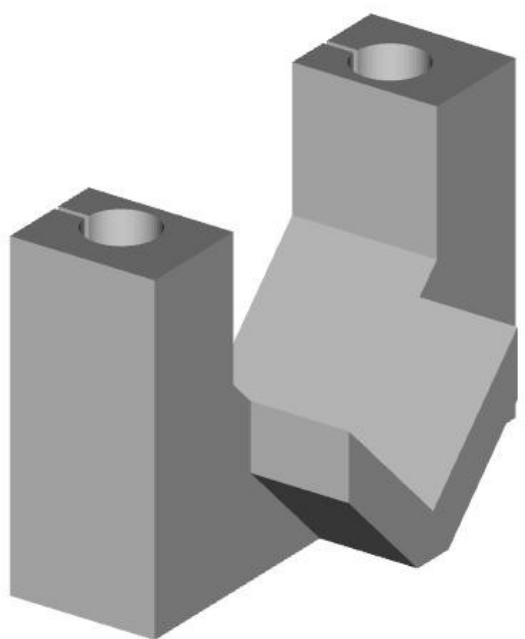
Pieds supports

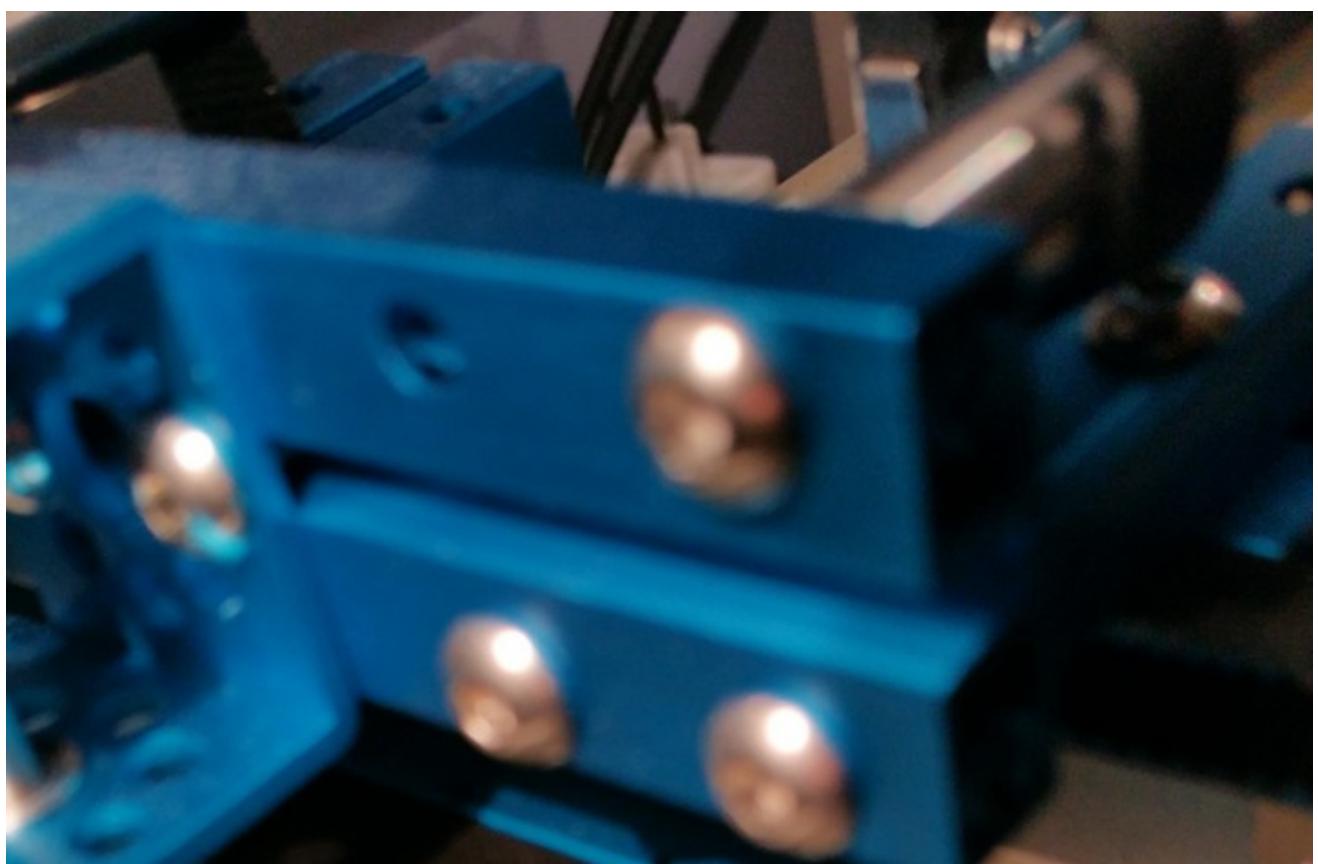
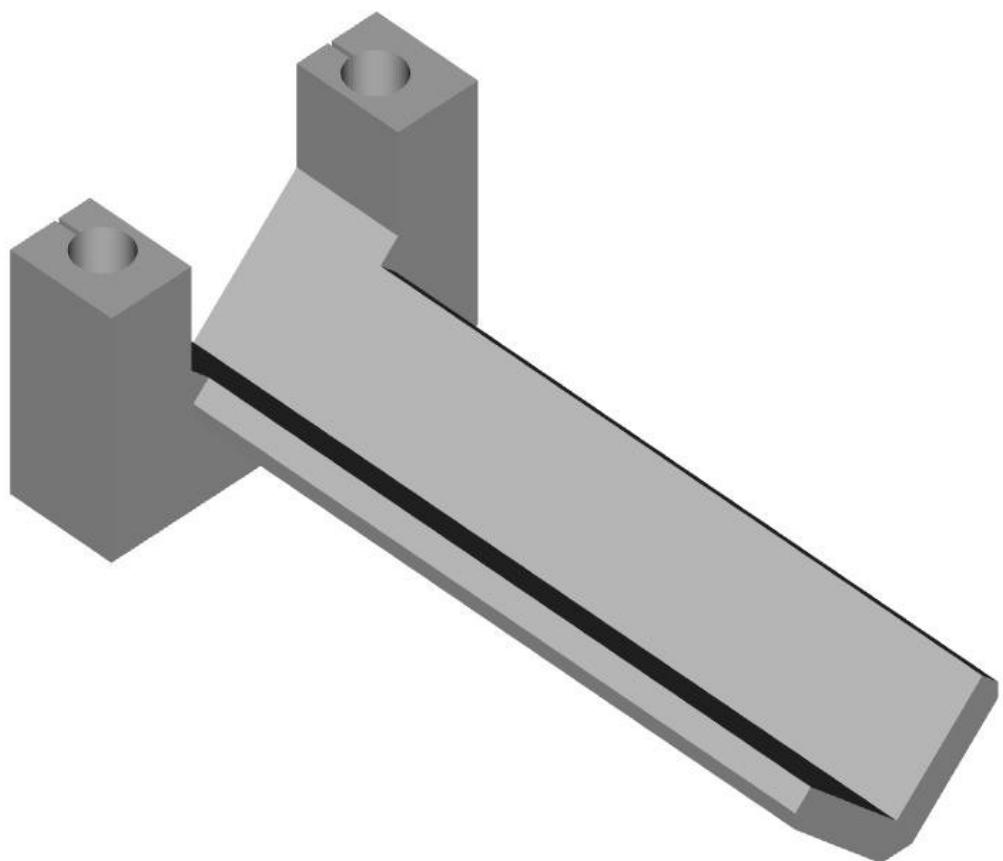




Support de transducteur

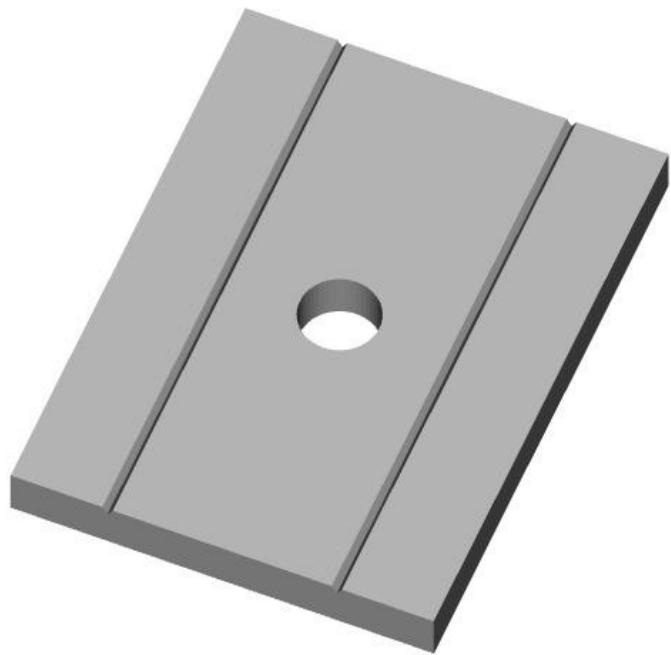




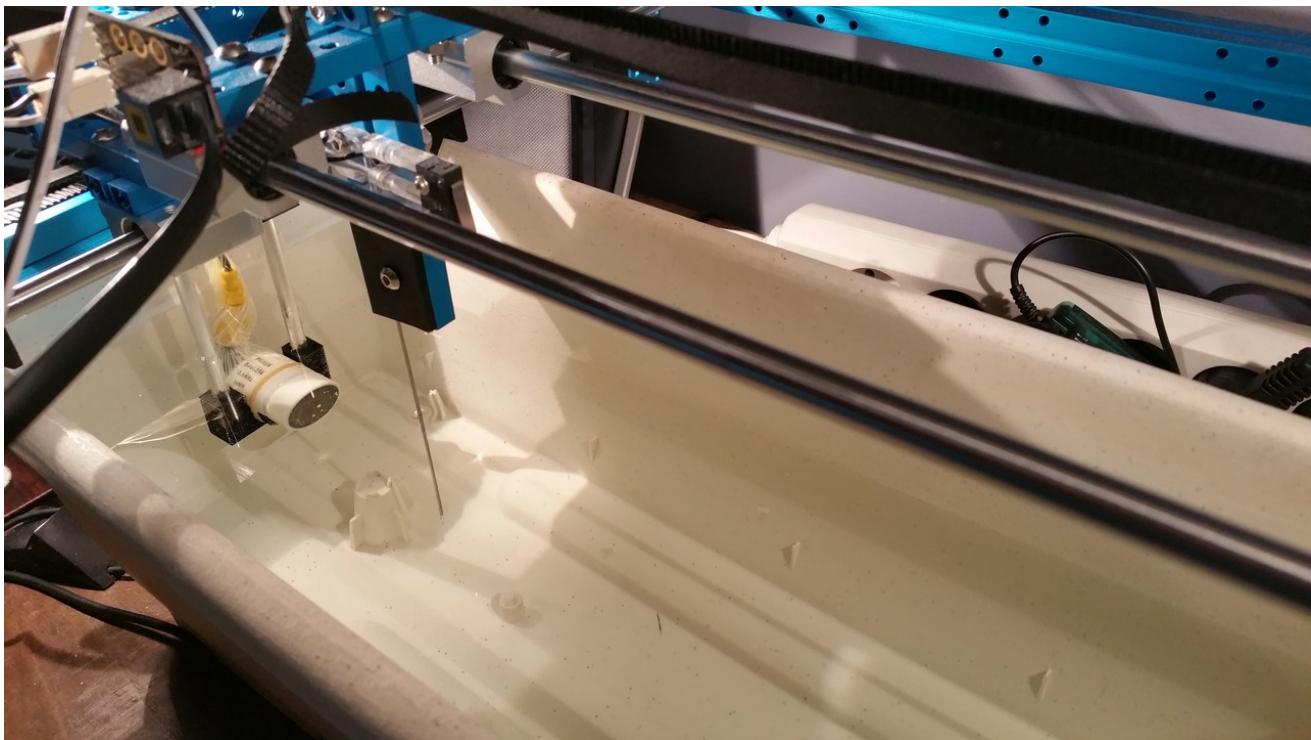




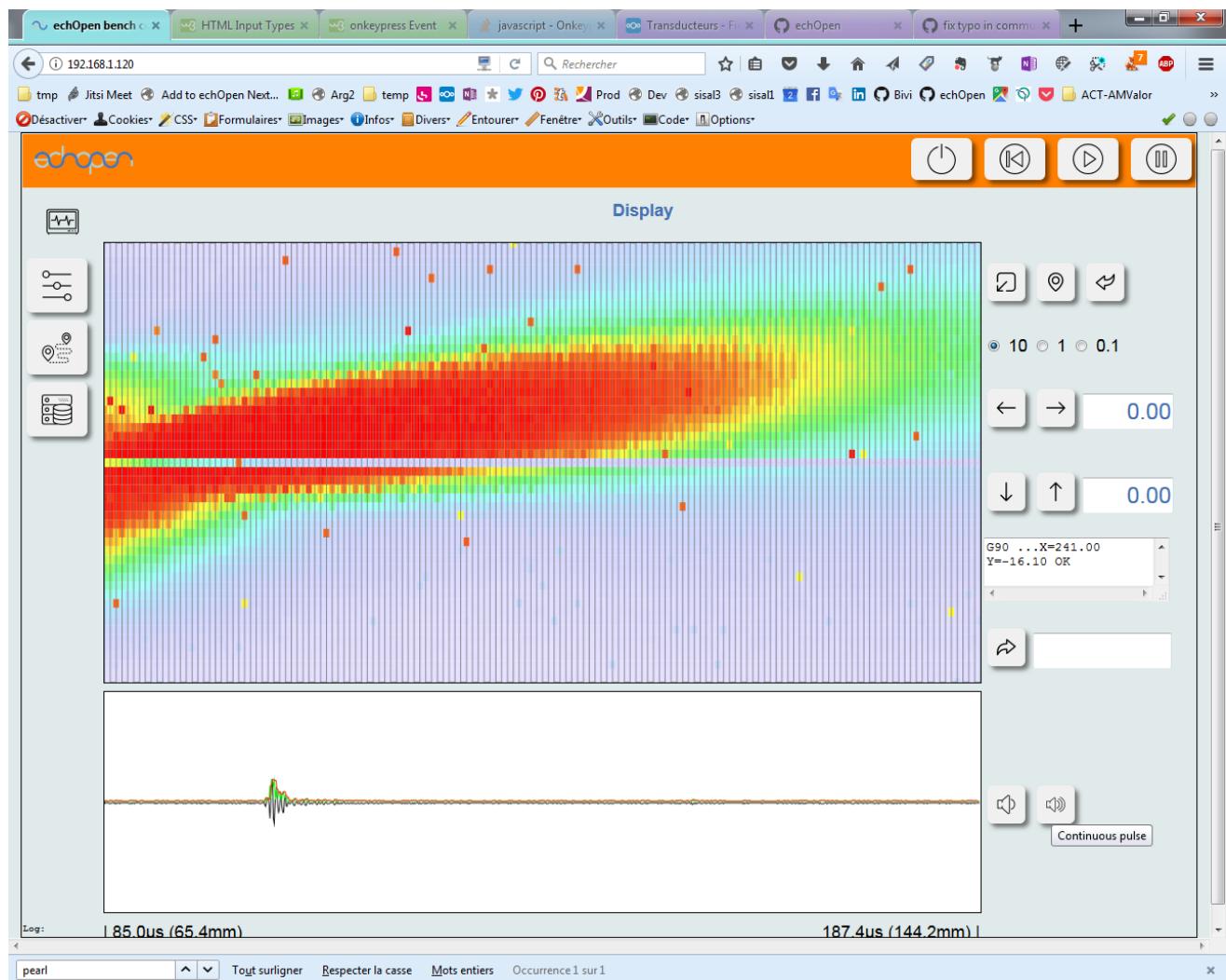
Pince de cible métallique







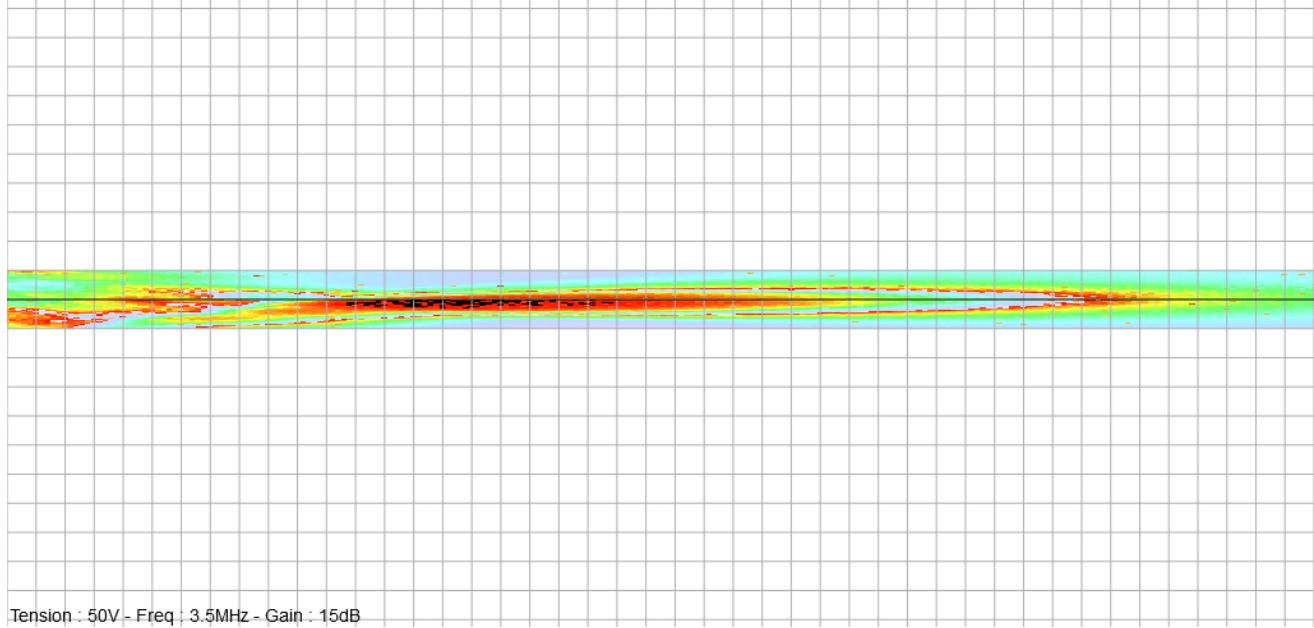
Premier essai



2017-11-16 Essais tranducteur en direct

Echelle orthonormée.

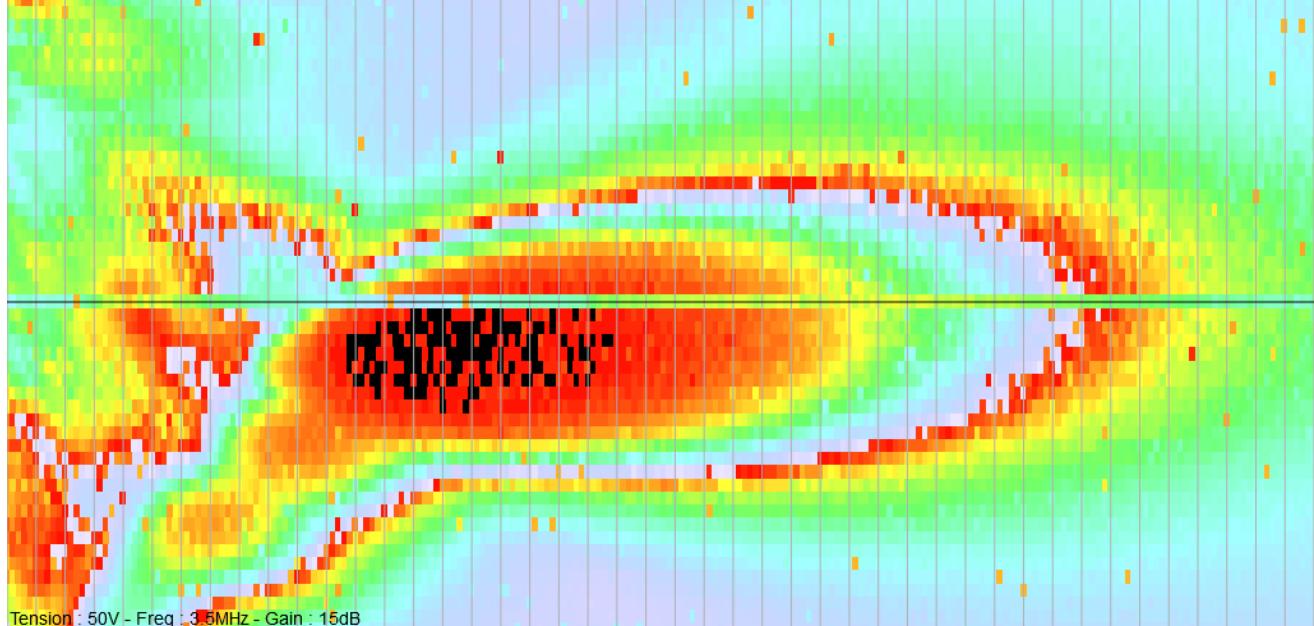
Test : Essai de scan horizontal - Size : 225mm x 10mm - Step : 1mm x 0.2mm - Speed : 50000mm/min - Scan mode : 1 - Delay : 5μs



Tension : 50V - Freq : 3.5MHz - Gain : 15dB

Echelle dilatée en Y.

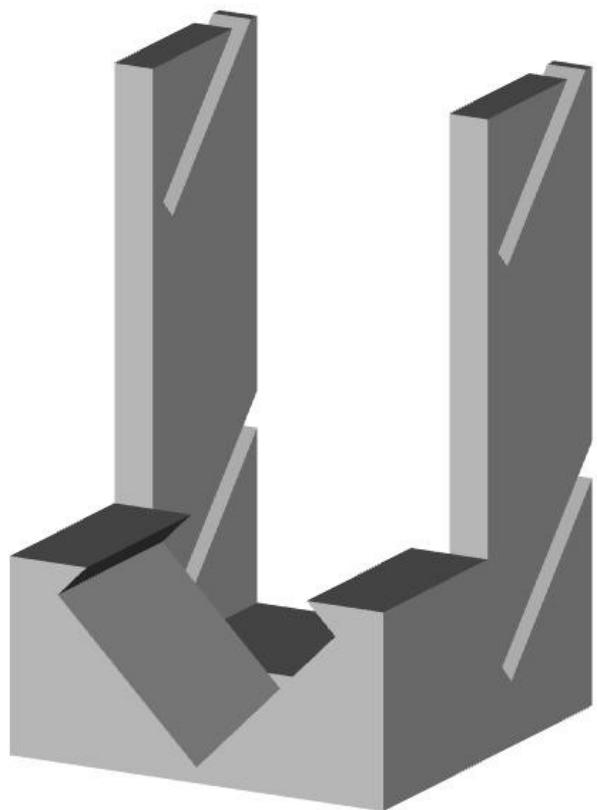
Test : Essai de scan horizontal - Size : 225mm x 10mm - Step : 1mm x 0.2mm - Speed : 50000mm/min - Scan mode : 1 - Delay : 5μs

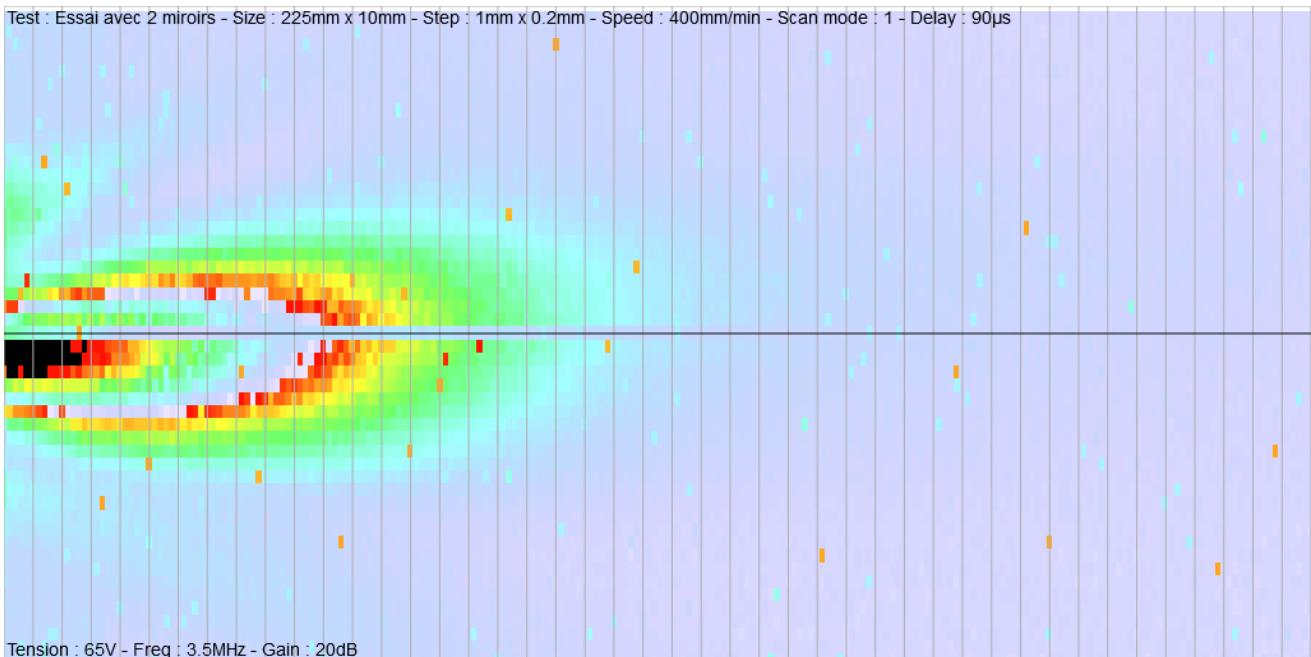


Tension : 50V - Freq : 3.5MHz - Gain : 15dB

2017-11-16 Essais avec périscope

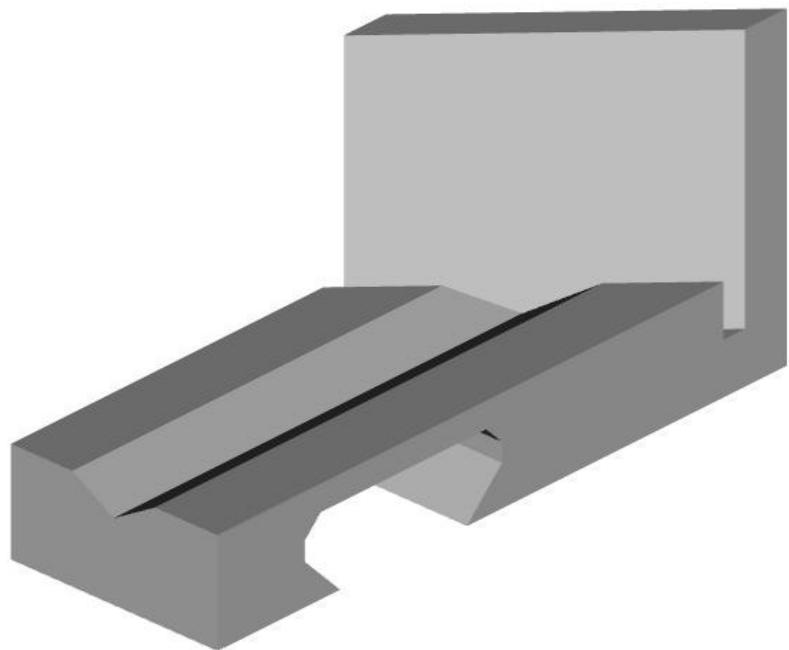
Essai avec un périscope, mais miroirs de mauvaise qualité.

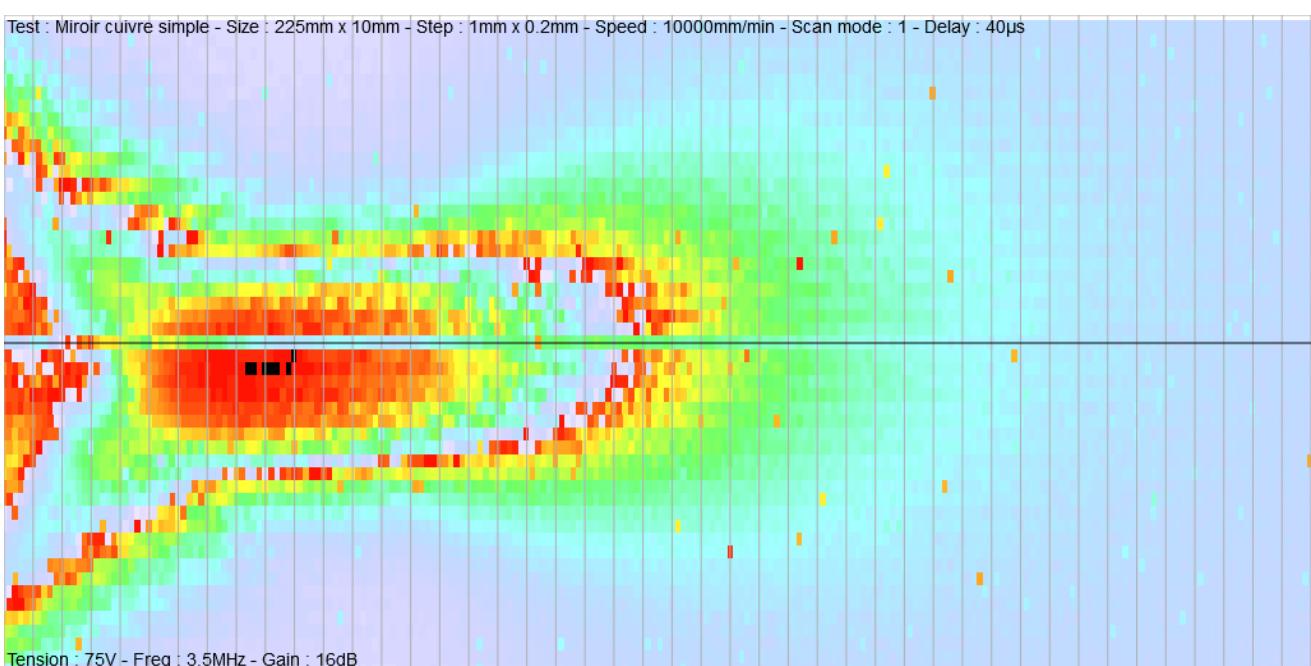




2017-11-18 Essais avec un miroir à 45°

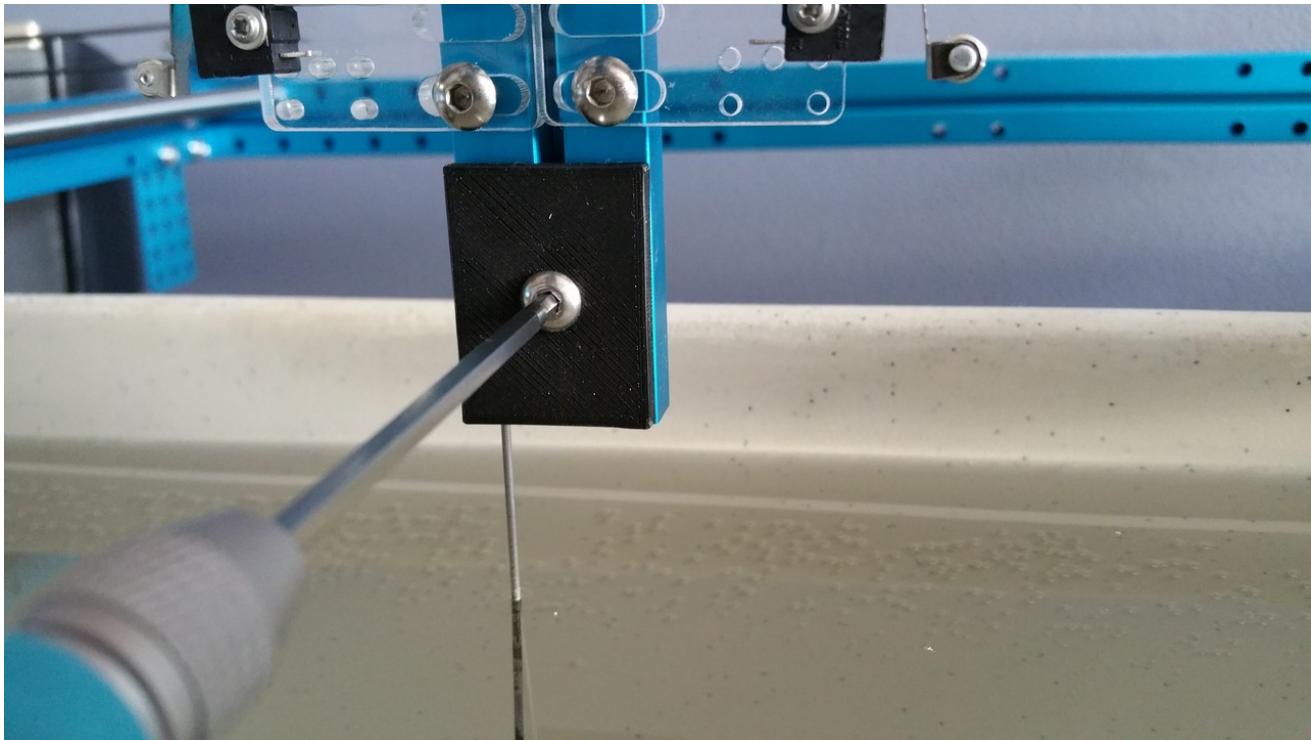
Support miroir à 45°. Le miroir est un bout de PCB simple face de 45x30.



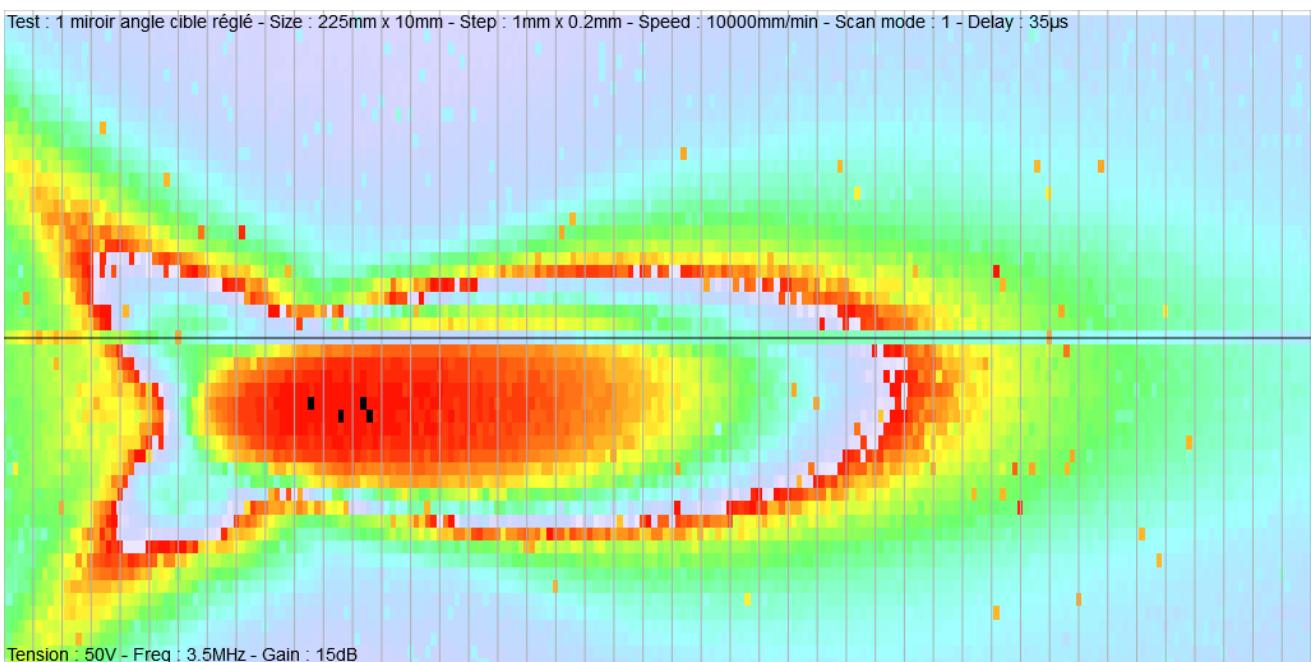


2017-11-19 Réglage verticalité de la tige cible

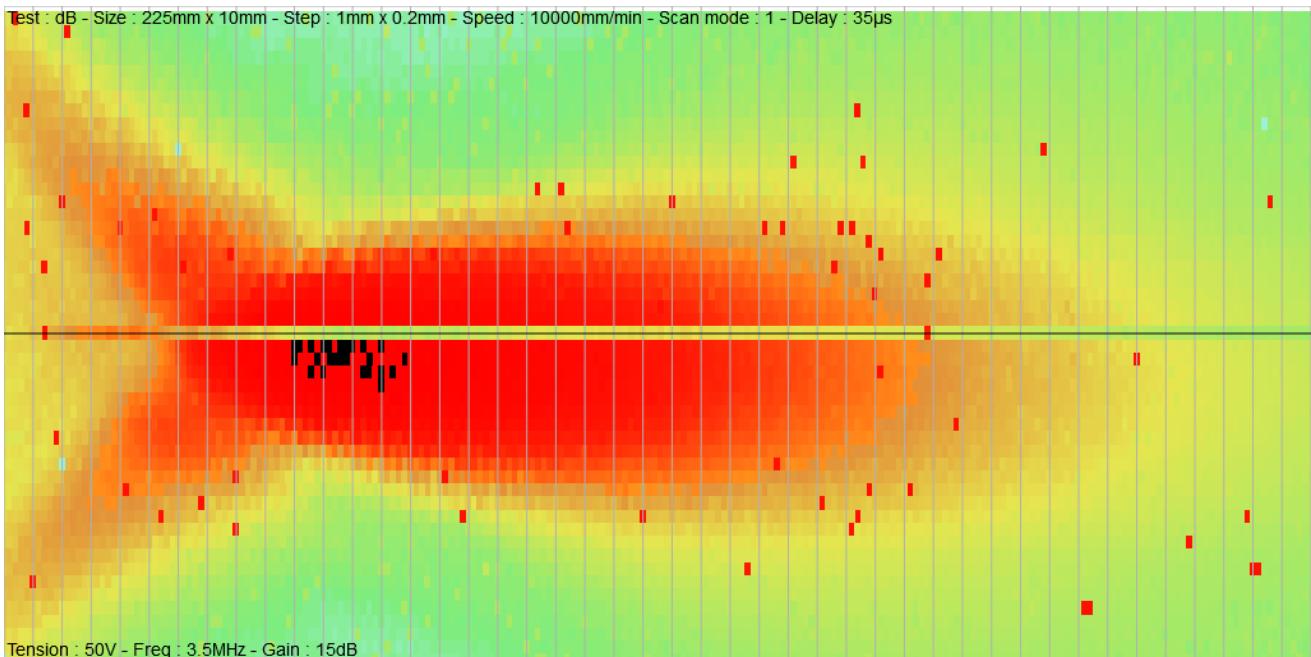
Réglage de l'angle de la tige cible par rapport à la verticale pour maximiser l'écho à 150mm.



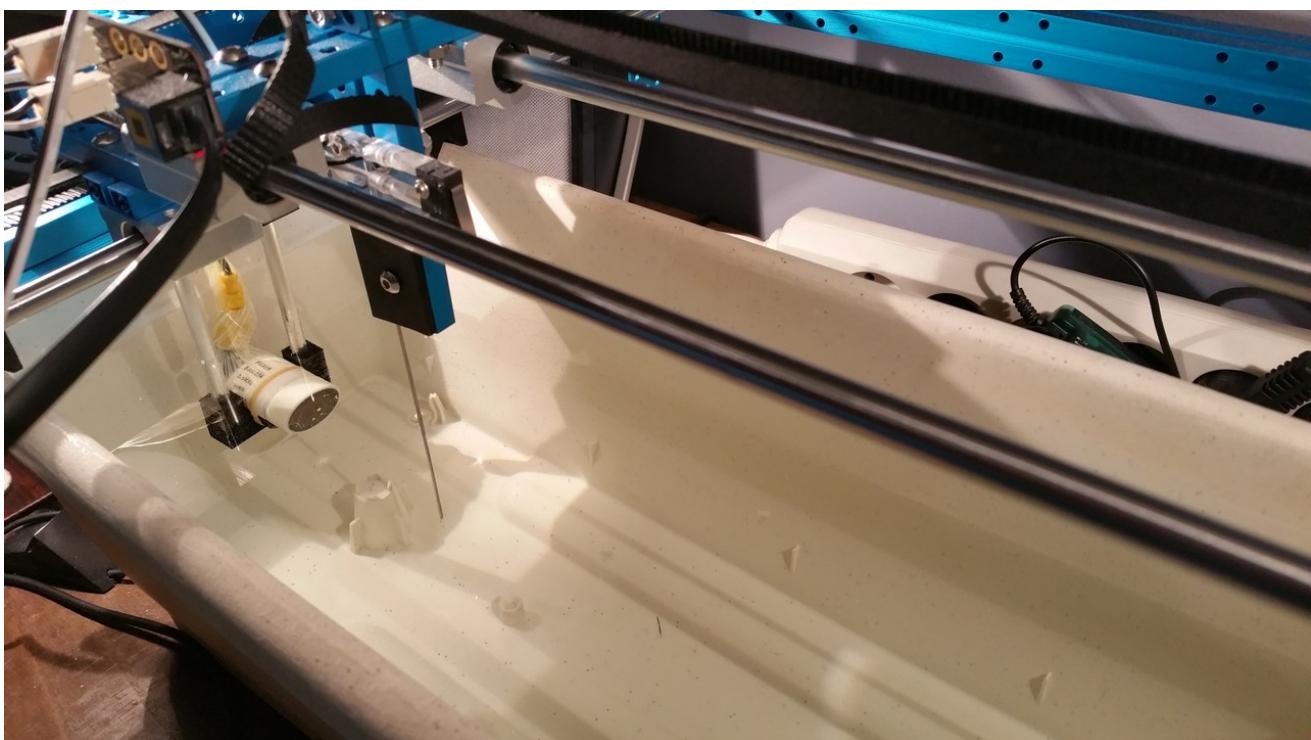
Le résultat obtenu est meilleur et comparable à la mesure sans miroir à tension et gain identiques.



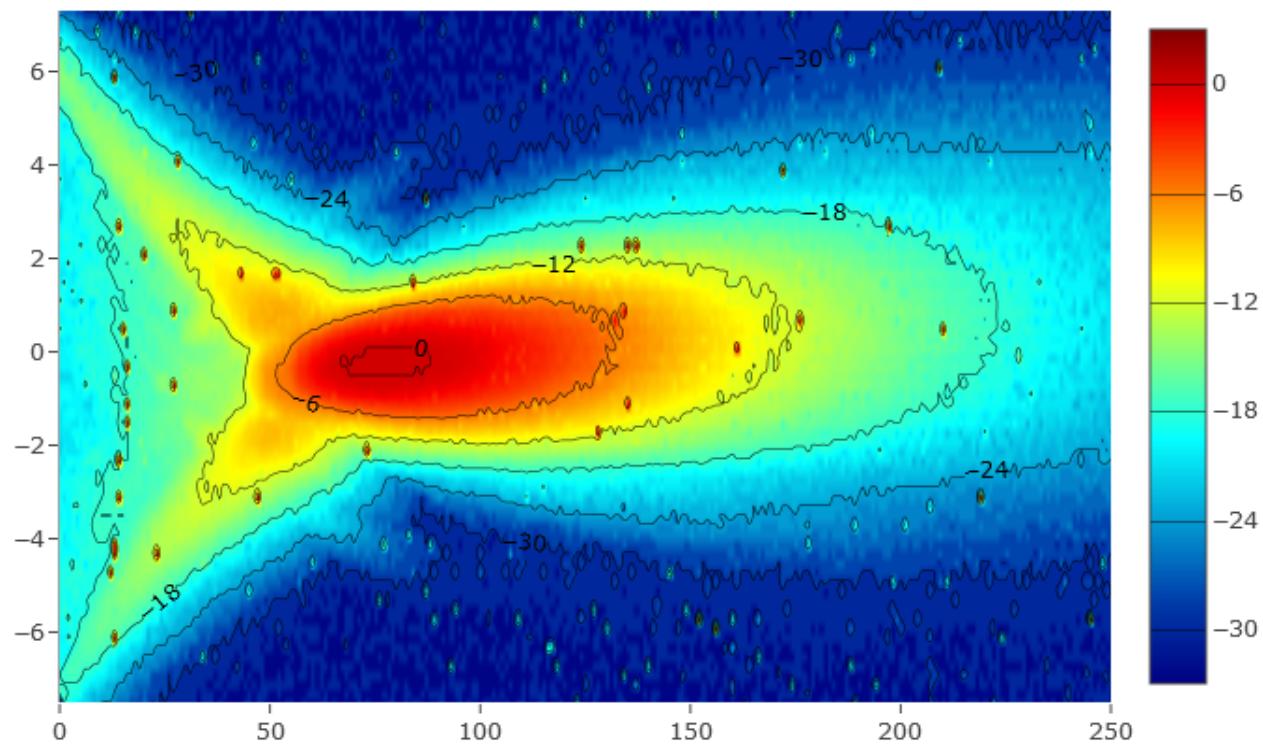
2017-11-19 Mesures affichées en dB



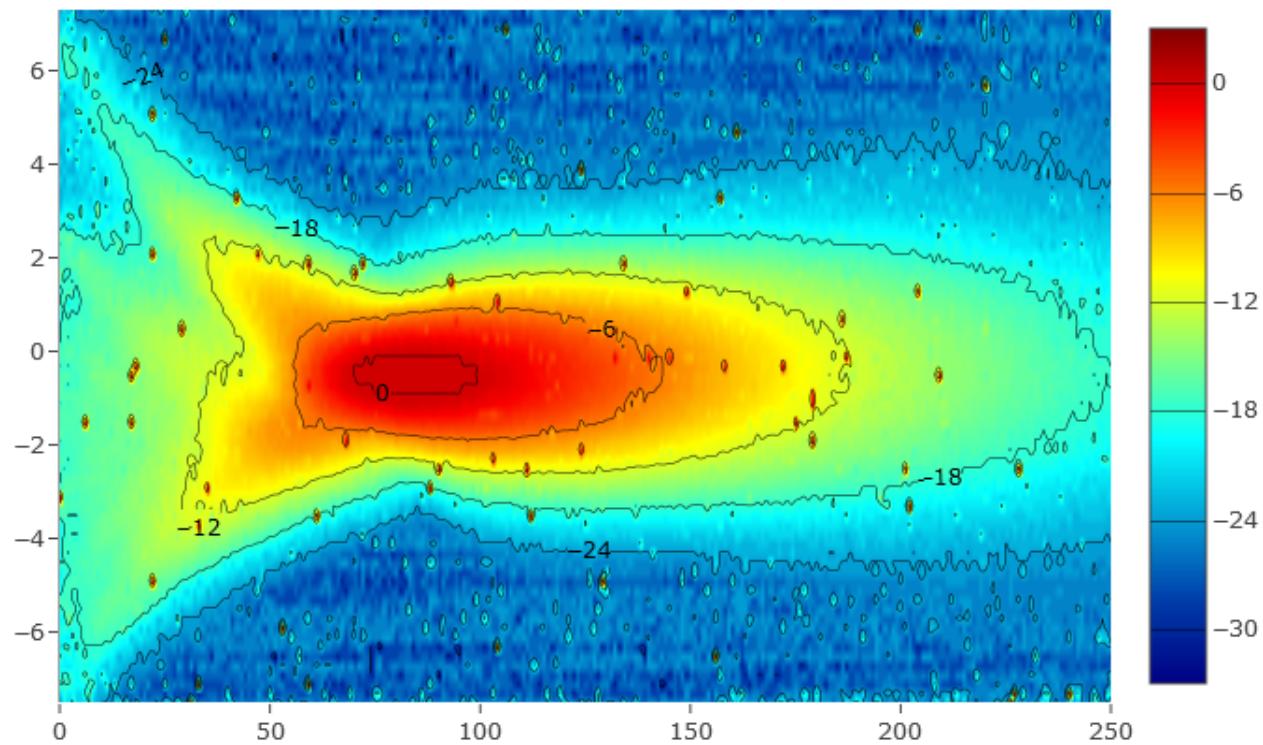
2017-11-26 Ajout Lib plotly et mesure imasonic



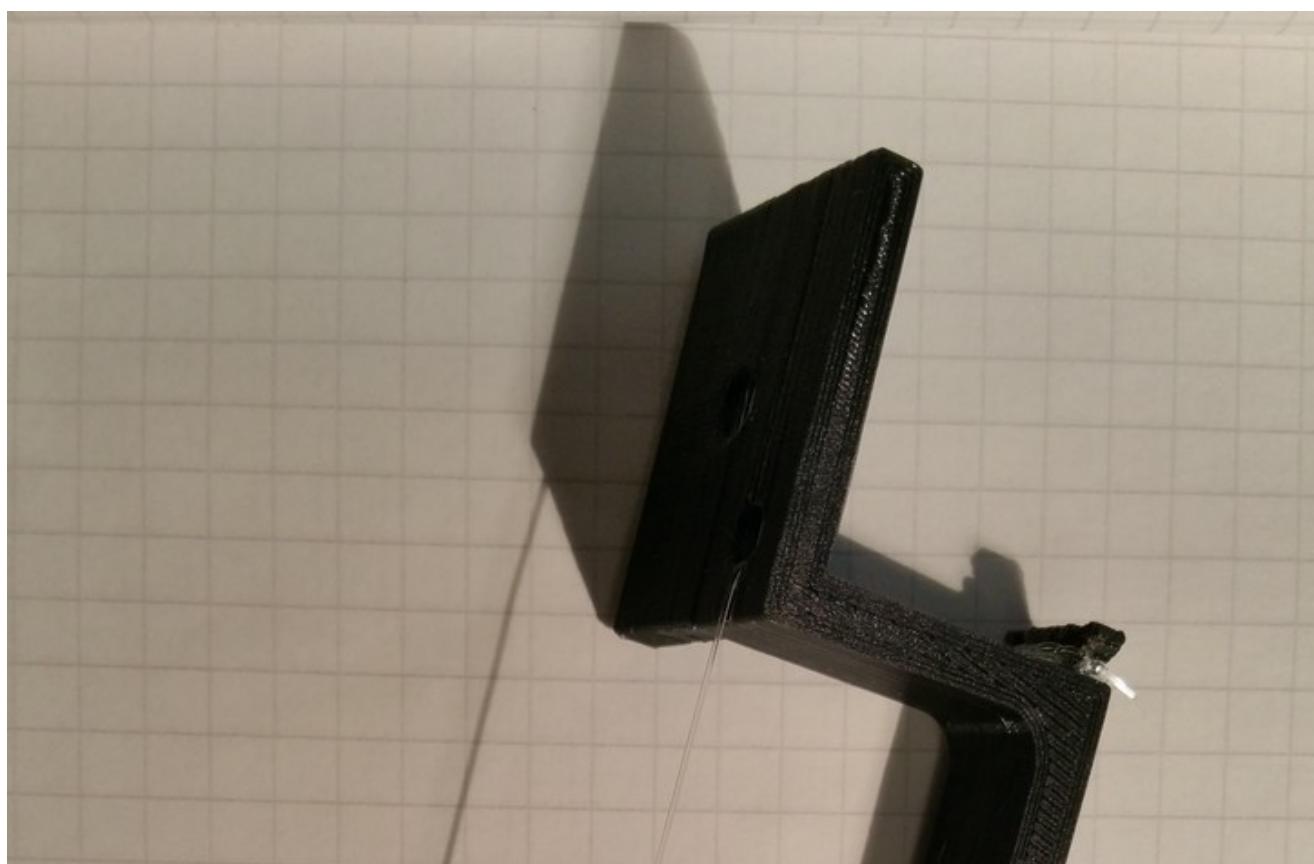
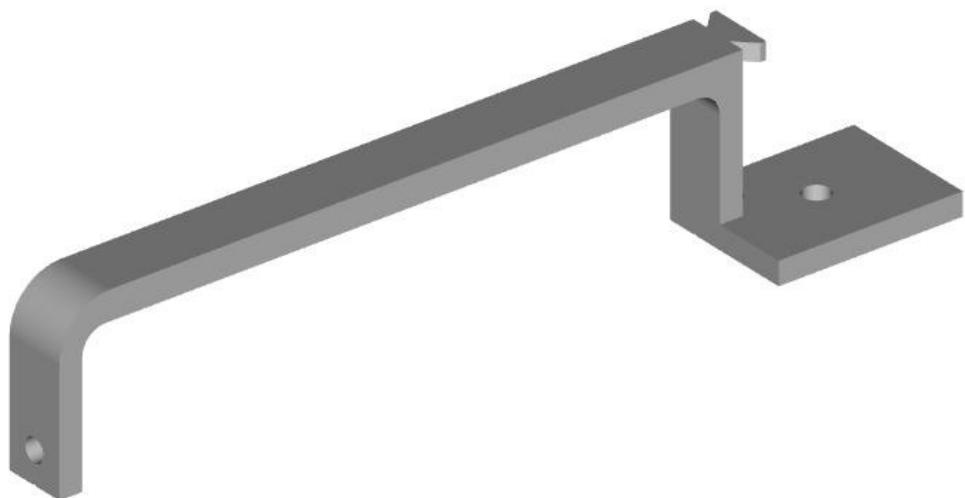
Test 3.5Labo, 250mm x 15mm, Step: 1mm x 0.2mm, Speed: 20000mm/min, Scan: 1, 5 μ s, 80V, 3.5MHz, 17dB

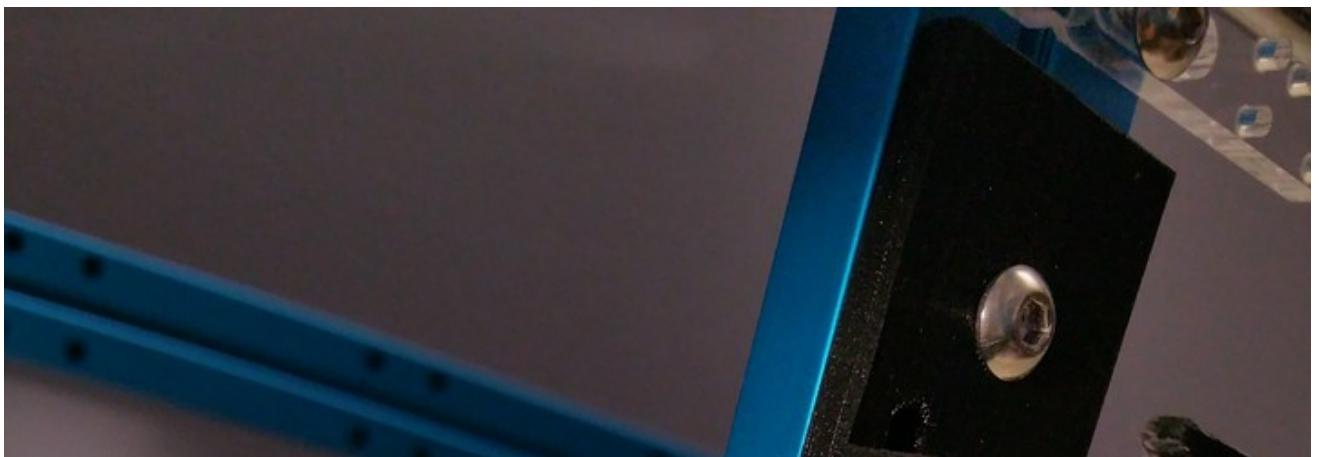


Test Imasonic, 250mm x 15mm, Step: 1mm x 0.2mm, Speed: 20000mm/min, Scan: 1, 5 μ s, 80V, 3.5MHz, 19dB



2017-11-24 Cible avec fil de nylon







Poubelle !!!

Communication RPi / PyBoard

- Installer PySerial :
`python -m pip install pyserial`
- Installer le serveur Web Bottle et l'extension pour faire du WebSocket :
`pip install bottle`

```
pip install python-dev  
pip install gevent-websocket
```

Installation RPi

Mise à jour du système

~~~~sh sudo apt-get update sudo apt-get upgrade sudo rpi-update sudo apt-get autoremove ~~~~ => avec rpi-update le firmware passe de Linux raspberrypi 4.1.19-v7+ #858 à rpi-4.4.y

Et reboot ! ... perte écran LCD :-(

### Config Samba

### Reconfiguration écran LCD

Atteindre le wiki Waveshare [http://www.waveshare.com/wiki/3.5inch\\_RPi\\_LCD](http://www.waveshare.com/wiki/3.5inch_RPi_LCD) (A) et télécharger la dernière version des drivers : <http://www.waveshare.com/w/upload/3/3d/LCD-show-160811.tar.gz>

Extraire l'archive dans `/home/pi/LCD-show` et :

~~~~sh cd LCD-show ./LCD35-show ~~~~

```
1 |
```

```
1 |
```