

FORMATION ACADEMIQUE
PEDAGOGIE DE PROJET
SUR LE NIVEAU PRE-BAC ET POST BAC
APPROCHE PLURIDISCIPLINAIRE (Mathématiques, Physique et SII)

DOSSIER ACTIVITE
CORRIGE



Partie 1: Validation du modèle mathématique du capteur:

Q1) $d = v \times t$ (d en m, v en m/s et t en μs)

Q2)

L'annexe 4 activité 2 nous permet de définir la célérité des ultrasons dans l'air. Pour une température moyenne de 30°C, nous pouvons identifier sur le document 1 $v = 350 \text{ m/s}$
L'annexe 3 outils mathématiques nous permet de construire l'équation de droite :

$$d_{(\text{cm})} = 350 \text{ 00} \times t / 1 \text{ 10}^6$$

$$d_{(\text{cm})} = 0,035 \times t \text{ comme les ultrasons font un aller et retour il faut diviser par 2 } d_{(\text{cm})}$$

$$d_{(\text{cm})} = 0,035 \times t / 2$$

$$d_{(\text{cm})} = 0,0175 \times t \text{ si on prend l'inverse de 0,0175 nous obtenons 58,14.}$$

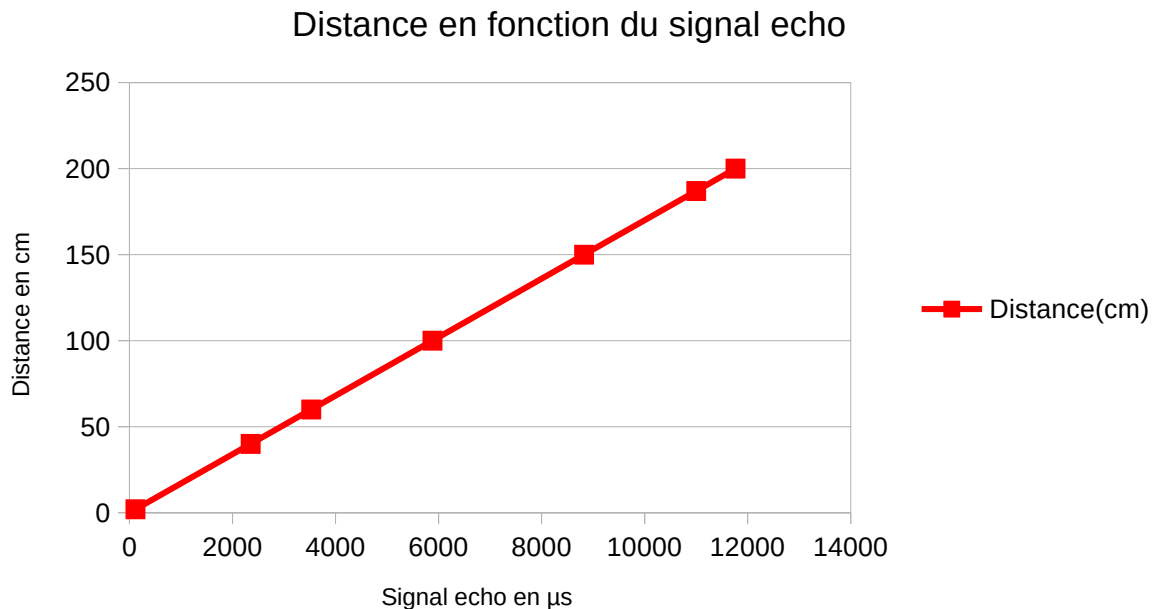
nous pouvons donc considérer que $d_{(\text{cm})} = t / 58$

Partie 2: Représentation du modèle mathématique dans un repère orthonormé

Q1)

Durée echo (μs)	116	2353	3529	5882	8823	11000	11764
Distance _(cm)	2	40	60	100	150	187	200

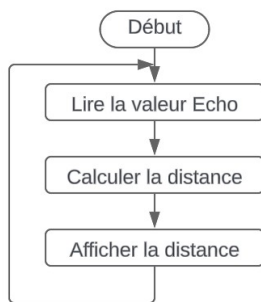
Q2)



Partie 3: Scénarios

Scénario A: Nous souhaitons afficher sur un écran LCD, la distance entre le capteur et un obstacle simulant le niveau de l'eau dynamique dans un réservoir.

Q1)



Q2)

```

#include "Wire.h"
#include "LiquidCrystal_I2C.h"

char buff[16];

int trigPin = 11; // Trigger
int echoPin = 12; // Echo
long duration, distance;

int LedRouge = 7;
int LedVerte = 6;

LiquidCrystal_I2C LCD(0x27,16,2);
    
```

```

void setup() {

    //Serial Port begin
    Serial.begin (9600);
    //Define inputs and outputs
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    pinMode(LedRouge, OUTPUT);
    pinMode(LedVerte, OUTPUT);

    LCD.init(); // initialisation de l'afficheur
    LCD.backlight();

    LCD.setCursor(0, 0);
    LCD.print("Distance :");

}
    
```

```

void loop() {

    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    pinMode(echoPin, INPUT);
    duration = pulseIn(echoPin, HIGH);

    distance = duration/58;

    /* Serial.print(distance);
    Serial.print(" cm");
    Serial.println();*/

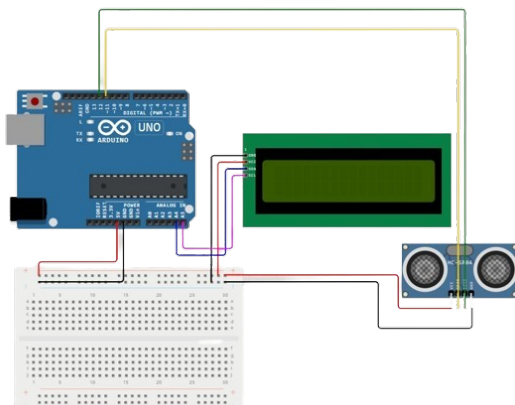
    LCD.setCursor(0, 1);

    sprintf(buff, "%3d distance ", cm);
    LCD.print(buff);

    delay(250);

}
    
```

Q3)



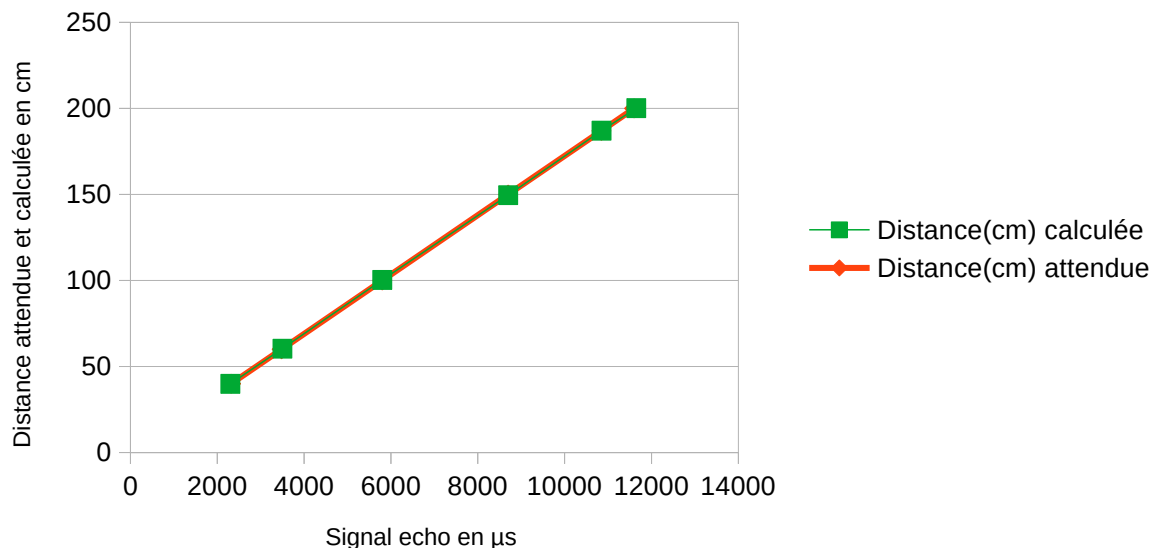
Q5)

Durée echo (μs) mesurée	2300	3500	5800	8700	10850	11600
Distance définie avec echo mes	39,65	60,34	100,0	150,00	187,1	200,0
Distance_(cm) attendue	40	60	100	150	187	200
Erreur absolue _(cm)	0,34	+0,34	0,0	0,0	+0,1	+0,0

Q6) Le capteur est bien conforme à l'exigence du constructeur par rapport à la valeur de l'erreur absolue imposée de $\pm 0,3$ cm. La moyenne des erreurs absolues nous donne MoyEa = 0,11 cm.

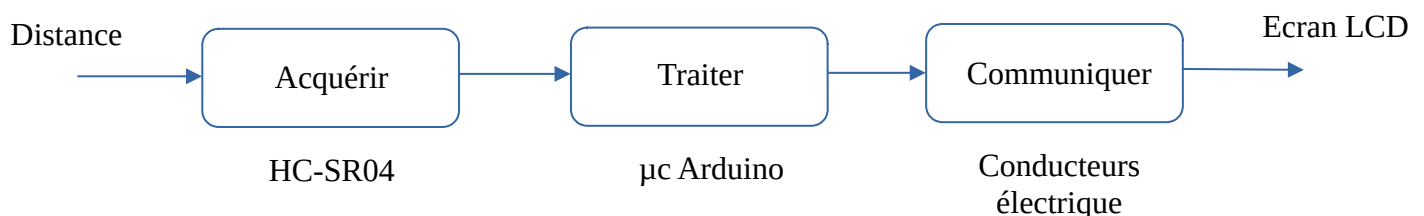
Q7)

Distance en fonction du signal echo



Nous pouvons observer la supersposition des deux droites.

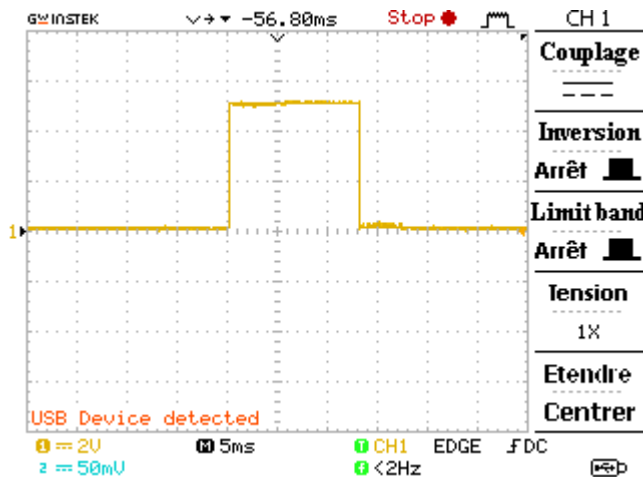
Q8)



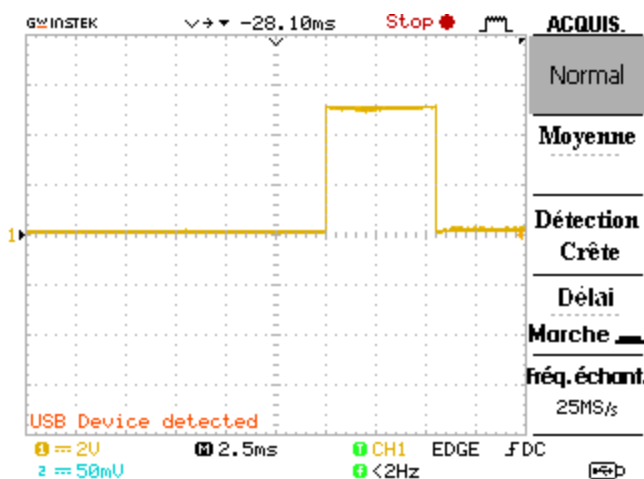
Par rapport à l'environnement industriel :

- la distance représente la hauteur du voile de boue
- le capteur HC-SR04 représente la FDU90 et FMU 90
- le μc Arduino représente l'automate
- les conducteurs électrique le réseau de câblage de la STEP
- l'écran LCD l'unité de supervision en salle de contrôle

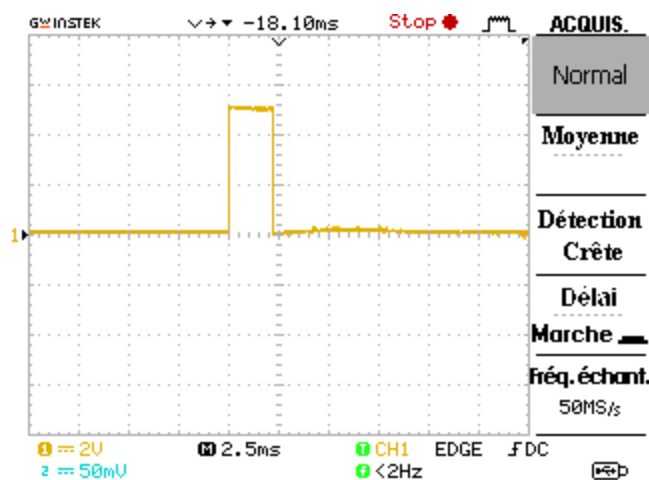
Q9) En utilisant les signaux echo de l'annexe B, retrouver les distances mesurées par le HC-SR04.



Base de temps = 5ms
 Ton = 13ms , Ton = 13000 μ s
 Amplitude du signal = 5v
 $D = 13000/58 = 224$ cm



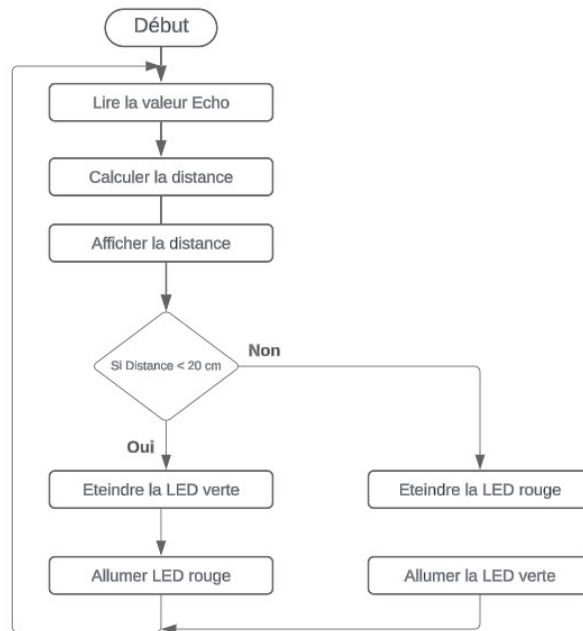
Base de temps = 5ms
 Ton = 11ms , Ton = 11000 μ s
 Amplitude du signal = 5v
 $D = 11000/58 = 189,65$ cm



Base de temps = 5ms
 Ton = 4,5ms , Ton = 4500 μ s
 Amplitude du signal = 5v
 $D = 4500/58 = 77,6$ cm

Scénario B:

Q1)



Q2) Compléter le programme du document réponse 3g en intégrant le test de la question précédente

```
#include "Wire.h"
#include "LiquidCrystal_I2C.h"
```

```
char buff[16];
```

```
int trigPin = 11; // Trigger
int echoPin = 12; // Echo
long duration, distance;
```

```
int LedRouge = 7;
int LedVerte = 6;
```

```
LiquidCrystal_I2C LCD(0x27,16,2);
```

```
void setup() {
    //Serial Port begin
    Serial.begin (9600);
    //Define inputs and outputs
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    pinMode(LedRouge, OUTPUT);
    pinMode(LedVerte, OUTPUT);

    LCD.init(); // initialisation de
    l'afficheur
    LCD.backlight();

    LCD.setCursor(0, 0);
    LCD.print("Distance :");
}
```

```
void loop() {

    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    pinMode(echoPin, INPUT);
    duration = pulseIn(echoPin, HIGH);
```

```
distance = duration/58;
```

```
If (distance < 20) {
    digitalWrite(LedVerte, LOW);
    digitalWrite(LedRouge, HIGH);
} else {
    digitalWrite(LedVerte, HIGH);
    digitalWrite(LedRouge, LOW);
}
```

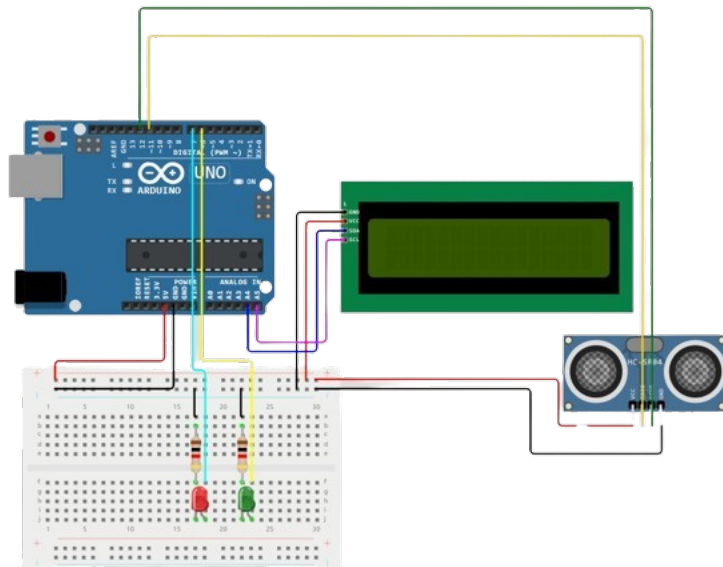
```
/* Serial.print(cm);
Serial.print(" cm");
Serial.println();*/

LCD.setCursor(0, 1);

sprintf(buff, "%3d distance ", cm);
LCD.print(buff);

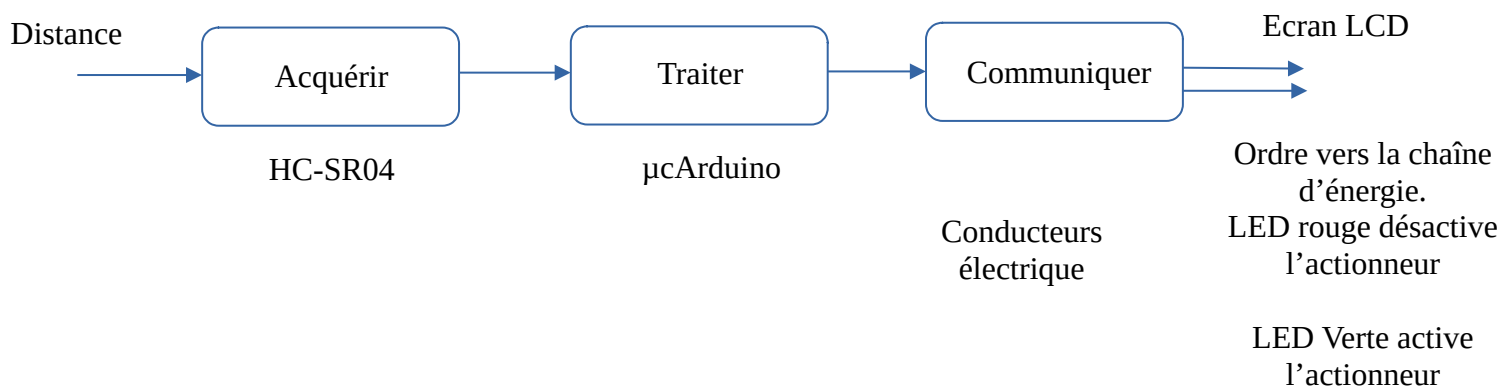
delay(250);
}
```

Q3)



Q5) Le basculement des LED se fait bien à 20cm

Q6) Compléter la chaîne d'information du document réponse 3i



Par rapport à l'environnement du laboratoire du CIRA :

- la LED rouge représente l'état de fermeture de la vanne d'arrivée d'eau
- la LED verte l'état d'ouverture de la vanne d'arrivée d'eau

Partie 4 : Principe d'exploitation de l'ultrason en STS CIRA

Q1) En vous aidant de l'**annexe1** donner l'étendue de mesure du capteur en cm et en μs pour :

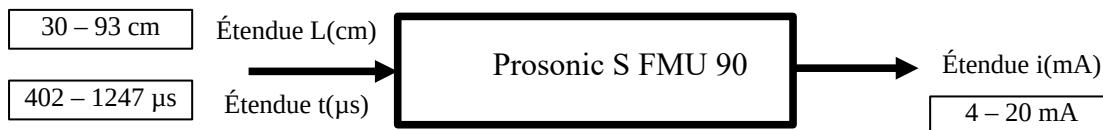
$$E = 70 \text{ cm et } DB = 70 \text{ mm}$$

Avec les conditions initiales imposées à la cuve l'encadrement de l'étendue de mesure F est donnée par :

$$\begin{aligned} F_{\min i} &< F < F_{\max} \\ 100 - 70 &< F < 100 - 7 \\ 30 \text{ cm} &< F < 93 \text{ cm} \end{aligned}$$

Ici 30 cm représente le niveau bas de référence ($L_{(\text{bas de référence en cm})}$) de la cuve et 93cm le niveau haut maximum.

Q2.1) La FMU 90 convertit l'étendue de la durée echo en un signal 4-20 mA. Ce signal est transféré par la suite à un calculateur afin d'être traité par un programme.



$$i(\text{mA}) = (\Delta i_{\text{mA}} / \Delta L_{(\text{cm})}) \times (L_{(\text{cm})} \text{ mesurée} - L_{(\text{niveau bas de référence en cm})}) + i_{\text{minimal}}$$

$$i(\text{mA}) = ((20 - 4) / (93 - 30)) \times (L_{(\text{cm})} \text{ mesurée} - 30) + 4$$

$$i(\text{mA}) = (16/63) \times (L_{(\text{cm})} \text{ mesurée} - 30) + 4$$

Définissons l'étendue de t(μs) :

Dans les conditions initiales imposées, nous pouvons identifier la distance maximale et minimale parcourue par les ultrasons dans le réservoir.

$$\begin{aligned} D_{\min i} &< D < D_{\max i} \\ 0 \text{ cm} &< D < 70 \text{ cm} \end{aligned}$$

Donc en utilisant le modèle mathématique défini dans la partie 1 avec les données de l'**annexe1** **partie 4**, on en déduit l'étendue de t(μs).

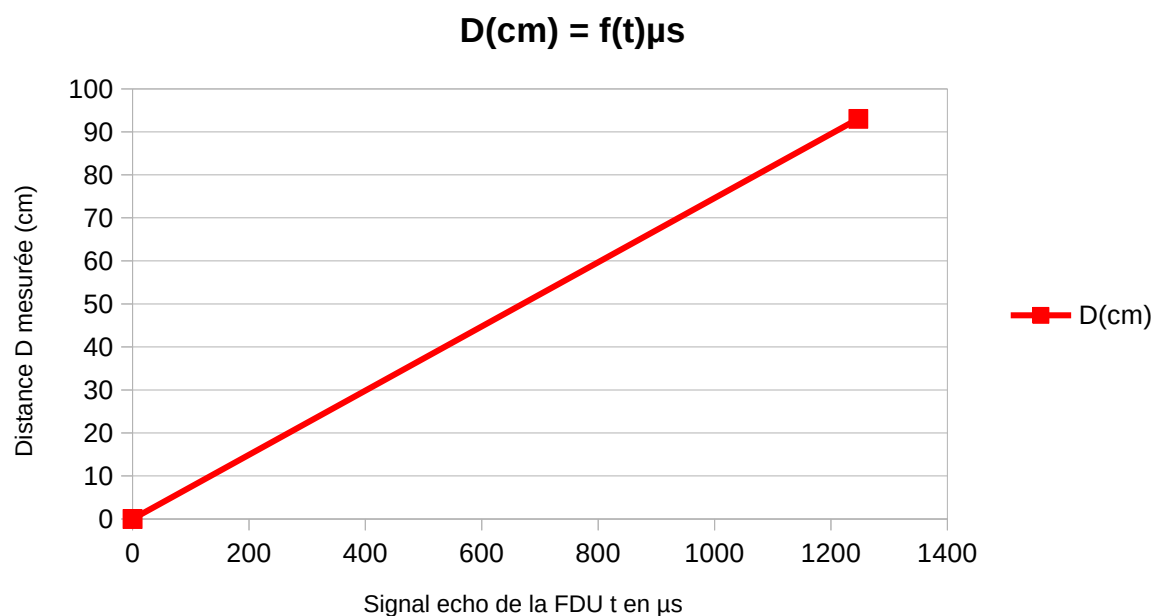
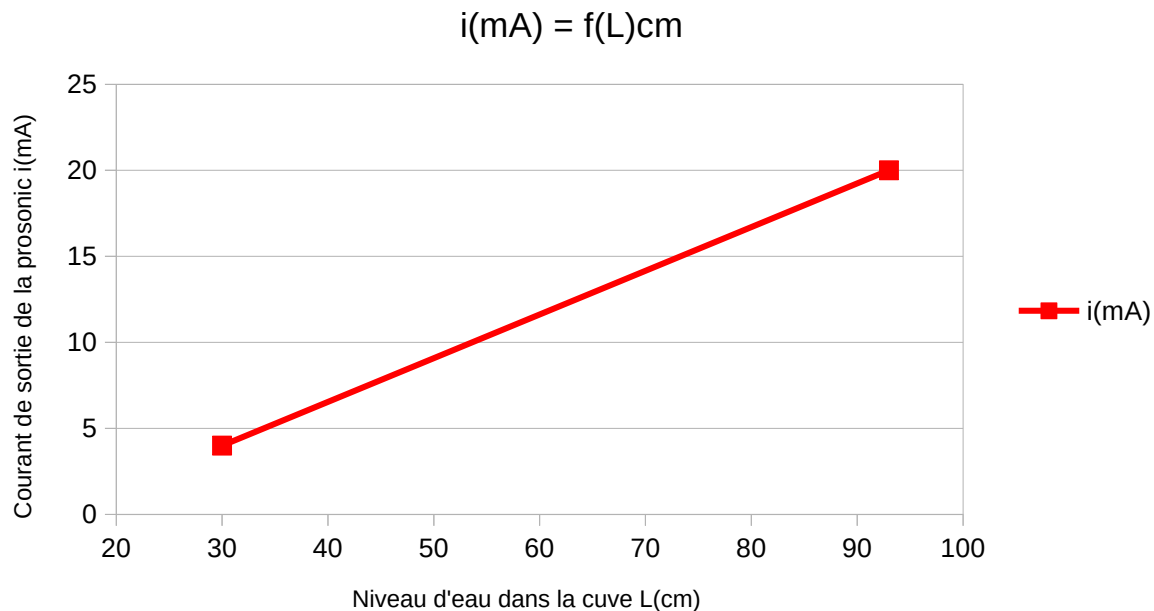
$$\begin{aligned} t_{\min i} (\mu s) &< t(\mu s) < t_{\max i} (\mu s) \\ 0 &< t(\mu s) < 1247 \end{aligned}$$

En utilisant la même démarche que pour i_{mA} , nous pouvons écrire la relation entre $D_{(\text{cm})}$ et $t_{(\mu s)}$.

$$D_{(\text{cm})} = ((70 - 0) / (1247 - 0)) \times (t_{(\mu s)} \text{ mesuré}) + 0$$

Dans ce cas le signal de l'écho part directement de la base de l'ultrason, le seuil de référence vaut donc 0 pour la mesure de D.

Q2.2 Représenter : $i_{(mA)} = f(L)_{cm}$ et $D_{(cm)} = f(t)_{\mu s}$ en utilisant le document réponse 4.



Q3) Les deux droites sont exactement les mêmes. Que nous analysons un matériel industriel ou didactique nous constatons que le principe de fonctionnement reste le même.