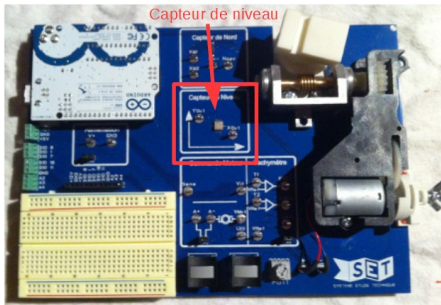


Objectif :

Dans de nombreux cas, il s'avère utile de caractériser le fonctionnement d'un capteur en vue d'une utilisation future.

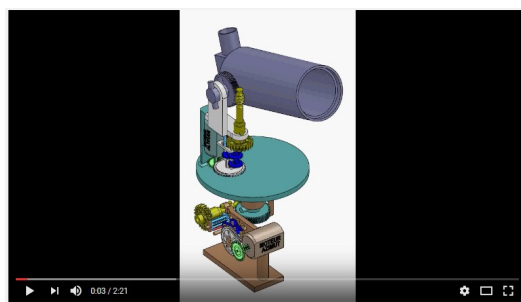


La présente activité a pour finalité de vous apprendre à caractériser les signaux de sorties du capteur de niveau (photo ci-contre) en vue d'afficher le niveau en X et Y sur un écran LCD.



Mise en situation

Le télescope Astrolab est un télescope capable de suivre les astres choisis au préalable malgré la rotation de la terre. Pour ce faire il a en mémoire les trajectoires de tous les astres. Cela peut fonctionner à condition que le télescope sache où il est situé et qu'il puisse corriger les trajectoires selon son inclinaison qui n'est jamais parfaite. Pour cela il utilise entre autres un capteur de niveau.



A disposition :

Vous disposez des matériels suivants :

- une platine de prototypage du télescope avec le capteur de niveau ;
- Oscilloscope numérique ;
- Multimètre FI225MP ;
- Boîtier Picoscope et le logiciel du même nom ;
- Sondes d'oscilloscope ;

Remarques préambulaires

L'oscilloscope vous permettra de visualiser les formes des signaux (rappel : la position DC vous montre le signal dans son intégralité, la position AC élimine la composante continue du signal).

Par ailleurs les oscilloscopes mis à disposition disposent de la fonctionnalité « mesure » vous permettant de mesurer différents caractéristiques (valeur efficace, moyenne, fréquence, etc). Cependant tous les oscilloscopes ne possèdent

pas cette fonctionnalité aussi vous préférez l'utilisation des multimètres pour mesurer les grandeurs moyennes et efficaces.

Remarque relative au multimètre FI225MP : Ne fournit pas la valeur efficace VRAIE mais uniquement la valeur moyenne du signal ainsi que la valeur efficace de la partie alternative.
La valeur efficace VRAIE devra donc être calculée au besoin.

Remarque relative au boîtier Piscope et au logiciel associé : Le boîtier et le logiciel remplacent un oscilloscope de table avec plusieurs avantages :

- 4 voies d'entrées au lieu de 2 ;
- générateur de fonctions intégré ;
- Sauvegarde de courbe et utilisation aisée pour exploitation

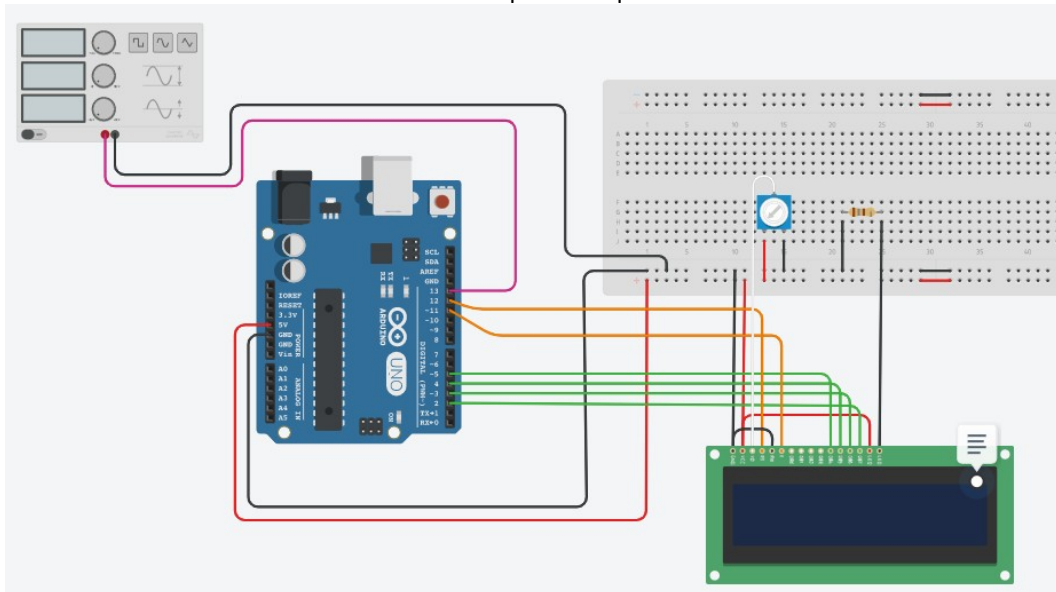
Cependant vous ne vous livrez pas à l'utilisation des Picoscopes une fois les oscilloscopes de table maîtrisés.

IMPORTANT : toute mise sous tension devra être au préalable faire l'objet d'une vérification par le professeur.

Travail demandé :

1. Le capteur de niveau possède deux sorties notées « X » et « Y ». **Définir** à partir de mesures si les sorties sont analogiques ou digitale (binaire).
2. **Réaliser** un tableau permettant de consigner la valeur de tension u_x issue de la sortie « X » pour différentes inclinaisons du capteur. Remarque : la sortie « Y » se comporte à l'identique que la sortie « Y ».
3. **Tracer** la caractéristique de transfert $u_x = f(\text{inclinaison})$ et définir la zone de linéarité du capteur.
- 4.
5. Modéliser de manière mathématique la fonction de transfert (en d'autres termes, trouver l'équation mathématique) dans la zone linéaire.
6. A partir du schéma ci-dessous, présentant le câblage d'un afficheur LCD, **réaliser** l'affichage de l'inclinaison pour selon X (et selon Y selon le temps qu'il vous reste).

Dans le montage, le potentiomètre simule le capteur analogique (attention dans le programme fourni, la grandeur analogique varie de 20 à 80, ce qui n'est pas le cas de votre capteur. Il faudra donc adapter le code). Par ailleurs, la fonction de transfert le potentiomètre est linéaire dans toute la plage de variation, ce qui n'est peut-être pas le cas du capteur de niveau. Le cas échéant vous ne utiliserez le capteur uniquement dans la zone linéaire.



Cliquer sur le schéma pour accéder au schéma et au code dans le simulateur en ligne « Tinkercad.com »

Le code disponible sur TinkerCad

```
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
//Déclaration des constantes et variables
const int pinGbf=13; //GBF connecté sur broche 13
unsigned long Ton=0; // Temps au niveau haut du signal en microsecondes
unsigned long Toff=0; // Temps au niveau bas du signal en microsecondes
float T=0;
float alpha=0; // rapport cyclique
float f=0; // fréquence du signal

void setup() {
  //Configuration des entrées sorties
  pinMode(pinGbf, INPUT);
  //initialisation communication Série pour affichage sur
  //le moniteur série
  Serial.begin(9600);
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  lcd.setCursor(0,0);
  // Print a message to the LCD.
  lcd.print("Mon LCD marche!");
  lcd.setCursor(0,1);
  lcd.print("Blah blah blah");
  delay(1000);
}

void loop() {
  Ton=pulseIn(pinGbf,HIGH); // acquisition de la durée au niveau HAUT
  Toff=pulseIn(pinGbf,LOW); // acquisition de la durée au niveau BAS
  alpha=float(Ton)/float((Ton+Toff)); // Calcul du rapport cyclique
  T=(Ton+Toff)*0.000001;
  f=1/(T); // Calcul de la fréquence

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("alpha : ");
  lcd.setCursor(9,0);
  lcd.print(alpha);
  lcd.setCursor(0,1);
  lcd.print("frequence : ");
  lcd.setCursor(10,1);
  lcd.print(f);
  // Turn off the display:
  lcd.noDisplay();
  delay(500);
  // Turn on the display:
  lcd.display();
  delay(500);
  //Affichage sur le moniteur Série
  Serial.print("alpha = ");
  Serial.print(alpha);
  Serial.print("\t");
  Serial.print("frequence = ");
  Serial.println(f);
}
```