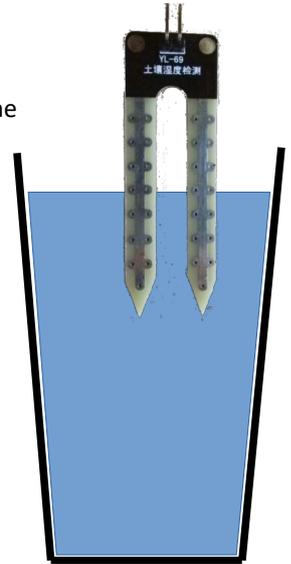
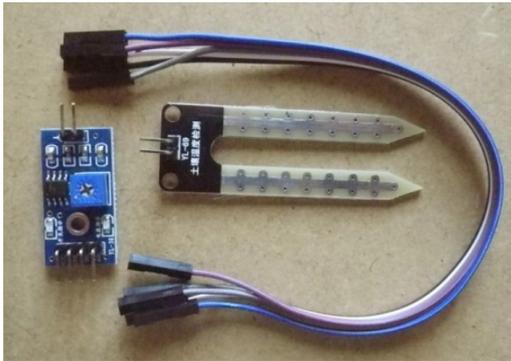


### Objectif :

---

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 d'humidité de sol (photo ci-dessous) en vue d'assurer la détection du niveau d'un liquide dans un petit récipient pour ainsi afficher le niveau sur un écran LCD.



### A disposition :

---

Vous disposez des matériels suivants :

- le capteur d'humidité de sol YL-69 et son conditionneur de signaux ;
- Oscilloscope numérique ;
- Multimètre FI225MP ;
- Boîtier Picoscope et le logiciel du même nom ;
- Sondes d'oscilloscope ;
- un verre en guise de petit récipient ;

### 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

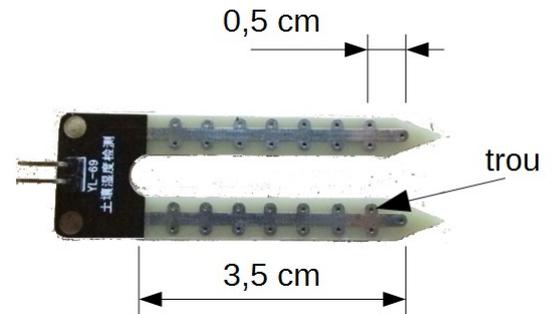
Cependant vous ne vous livrez à l'utilisation des Picoscopes une fois les oscilloscopes de table maîtrisés.

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

Travail demandé :

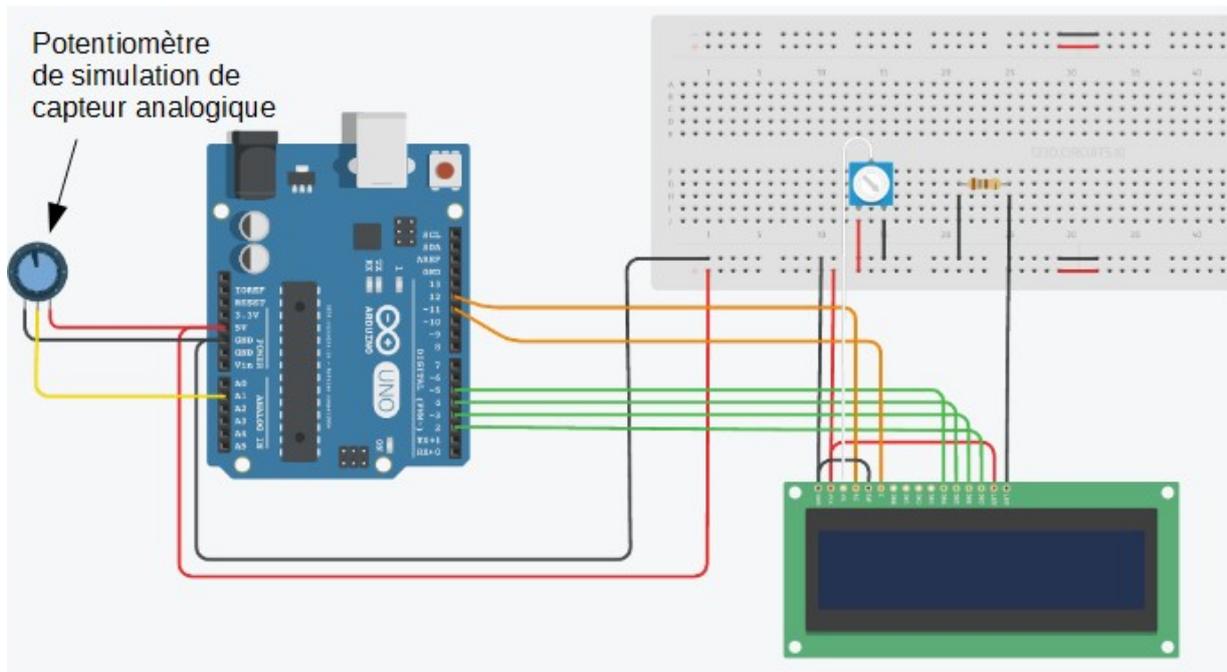
1. Le capteur possède deux sorties notées « DO » et « AO ». **Définir** à partir de mesures si les sorties sont analogiques ou digitale (binaire).

2. **Réaliser** un tableau permettant de consigner les valeurs de tension  $U_{AO}$  issues de la sortie « AO » lorsque le capteur est hors de l'eau, juste en contact avec les deux électrodes, puis à chacun des trous aménagés sur les électrodes (il y en a sept au total).



3. **Tracer** la caractéristique de transfert  $u_{AO} = f(\text{profondeur immergée des électrodes})$  et définir la zone de linéarité du capteur.
4. Modéliser de manière mathématique la fonction de transfert (en d'autres termes, trouver l'équation mathématique) dans la zone linéaire.
5. A partir du schéma ci-dessous, présentant le câblage d'un afficheur LCD, **réaliser** l'affichage de la hauteur immergée.

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);
float pinPot=A1;

void setup() {
  // 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(2000);
}

void loop() {
  int valPot=analogRead(pinPot);
  valPot=map(valPot,0,1023,20,80);
  lcd.clear();
  lcd.setCursor(0,1);
  lcd.print(valPot);
  // Turn off the display:
  lcd.noDisplay();
  delay(500);
  // Turn on the display:
  lcd.display();
  delay(500);
}
```