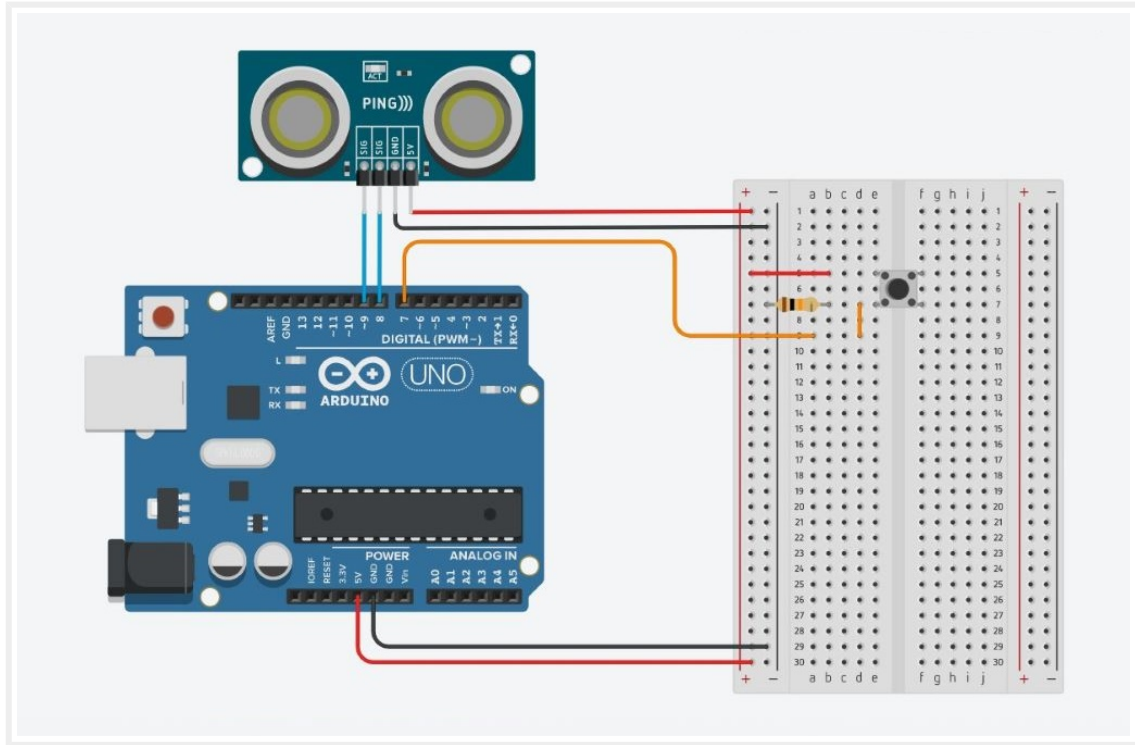


# Télémètre

## (Mesure de distances à l'aide d'un capteur ultrasonique)



### Liste des composants :

- . 1 capteur ultrasonique (HC-SR04)
- . 1 résistance de 10 kΩ (résistance du circuit du bouton poussoir)
- . 1 bouton poussoir
- . 1 plaque d'essai
- . Fils de connexion

### Objectif

Connaissant la célérité des ondes sonores dans l'air, dans cette activité, nous allons utiliser le capteur à ultrasons de notre circuit d'étude pour mesurer des distances en mesurant la durée de propagation **Dt** de l'onde sonore entre l'émetteur et le récepteur situés à une distance **d** inconnue d'un obstacle. C'est le principe du Sonar.

Les mesures de distance commencent après un appui sur le bouton poussoir et sont arrêtées en appuyant de nouveau sur celui-ci.

## . [Le programme](#)

Dans [le code de l'activité](#), la température de l'air est une variable qu'il est possible de modifier. La vitesse théorique du son dans l'air en fonction de la température indiquée est alors calculée et utilisée pour déterminer la distance entre le capteur et l'obstacle.

### Telemetre

```
// Déclaration des constantes et variables

int TRIGGER_PIN = 8;
int ECHO_PIN = 9;
const int PinButton = 7;

const unsigned long MEASURE_TIMEOUT = 25000UL;

int ValButton = 0;
int OldValButton = 0;
int State = 0;
int OldState = 0;

int Distance = 0;
int DistanceMeasure = 0;
long Dt = 0;

float Temp = 20.0;
float VitTheoUS = 20.05*(sqrt(Temp+273.15));

// Initialisation des entrées et sorties

void setup()
{
  Serial.begin(9600);
  pinMode(PinButton, INPUT);
  pinMode(TRIGGER_PIN, OUTPUT);
  digitalWrite(TRIGGER_PIN, LOW);
  pinMode(ECHO_PIN, INPUT);
  Serial.println("Appuyez sur le bouton poussoir pour mesurer \
la distance entre le capteur et l'obstacle.");
}
```

```

// Fonction principale en boucle

void loop()
{
  ValButton = digitalRead(PinButton);
  delay(10);

  if ((ValButton == HIGH) && (OldValButton == LOW))
  {
    State=1-State;
  }
  OldValButton = ValButton;

  if (State==1)
  {
    if (OldState == 0)
    {
      Serial.println("Mesure de la distance en cours.");
      Serial.println("");
      Serial.println ("Distance (cm) :");
      OldState=1;
    }

    digitalWrite(TRIGGER_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGGER_PIN, LOW);
    Dt = pulseIn(ECHO_PIN, HIGH, MEASURE_TIMEOUT);

    Distance = int(Dt / 2.0 * VitTheoUS/ 10000);

    if (DistanceMeasure != Distance)
    {
      Serial.println(Distance);
      DistanceMeasure = Distance;
    }
    Serial.flush();
    delay(100);
  }
  else
  {
    if (OldState == 1){
      Serial.println("Fin des mesures.");
      OldState = 0;}
  }
}

```

---

### Déroulement du programme :

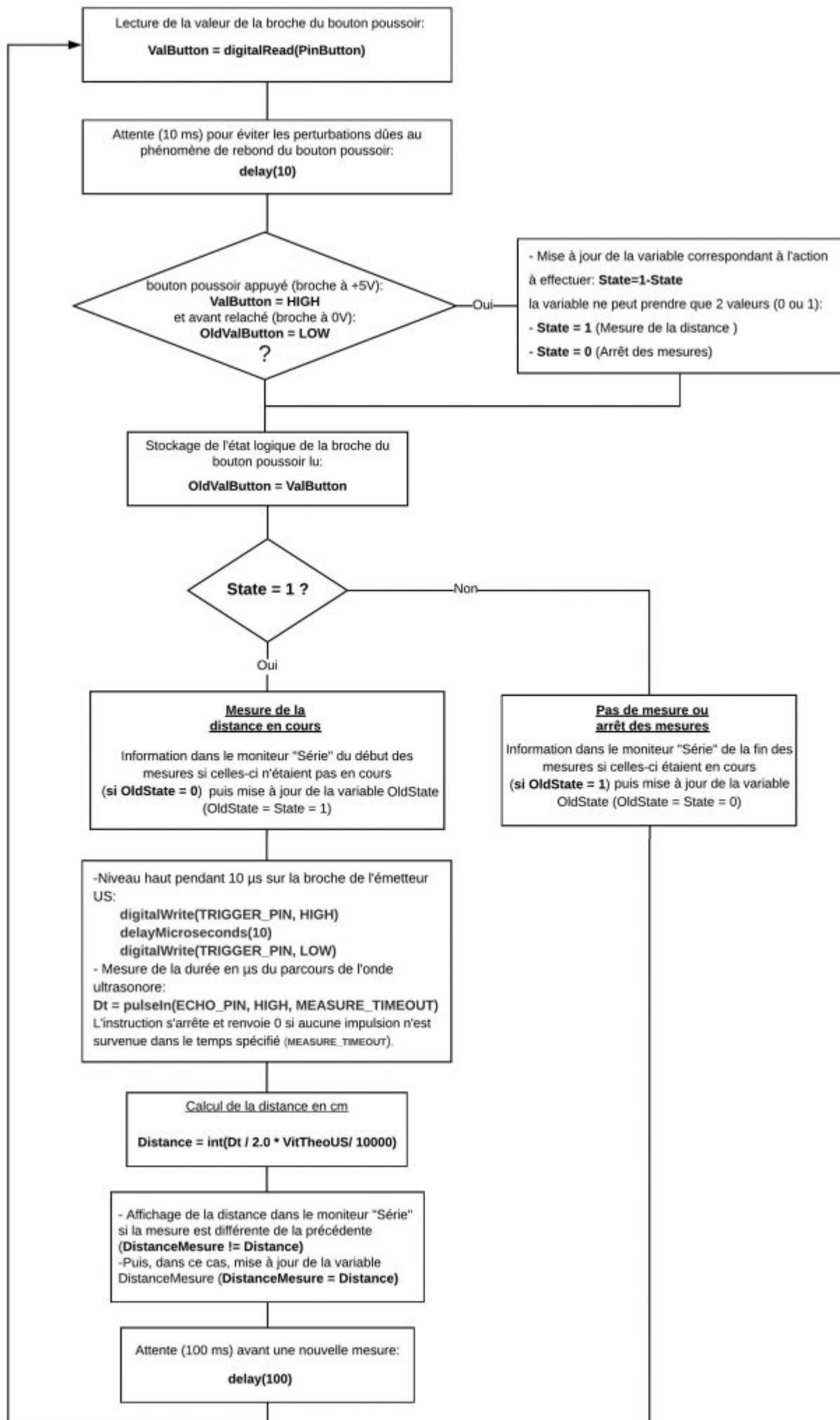
– Déclaration des constantes et variables :

- N° de la broche correspondant au bouton poussoir: **const int PinButton = 7**
- N° de la broche correspondant à l'émetteur US : **int TRIGGER\_PIN = 8**
- N° de la broche correspondant au récepteur US : **int ECHO\_PIN = 9**
- Constante pour définir la durée maximale des mesures : **const unsigned long MEASURE\_TIMEOUT = 25000UL**
- Variable pour stocker la valeur de la broche du bouton poussoir: **int ValButton = 0**
- Variable pour stocker l'ancienne valeur de la broche du bouton poussoir: **int OldValButton = 0**
- Variable correspondant à l'action à effectuer: **int State = 0**
- Variable pour stocker l'ancienne valeur de la variable correspondant à l'action à effectuer: **int OldState = 0**
- Variable correspondant à la distance mesurée: **int Distance = 0**
- Variable correspondant à la valeur précédente de la distance mesurée: **int DistanceMeasure = 0**
- Variable correspondant à la durée de parcours de l'onde ultrasonore: **long Dt = 0**
- Variable correspondant à la température de l'air: **float Temp = 20.0**
- Variable correspondant à la vitesse théorique du son dans l'air à la température définie:  
**float VitTheoUS = 20.05\*(sqrt(Temp+273.15))**


– Initialisation des entrées et sorties :

- le débit de communication en nombre de caractères par seconde pour la communication série est fixé à 9600 bauds: **Serial.begin(9600)**
- La broche du bouton poussoir est initialisée comme une entrée digitale. Des données seront donc envoyées depuis cette broche vers le microcontrôleur: **pinMode (PinButton, INPUT)**
- La broche de l'émetteur US est initialisée comme une sortie digitale. Des données seront donc envoyées depuis le microcontrôleur vers cette broche: **pinMode(TRIGGER\_PIN, OUTPUT)** et initialisée à un niveau bas (0 V): **digitalWrite(TRIGGER\_PIN, LOW)**
- La broche du récepteur est initialisée comme une entrée digitale: **pinMode(ECHO\_PIN, INPUT)**

– Fonction principale en boucle :



Résultats dans le moniteur série :

 COM3

Envoyer

Appuyez sur le bouton poussoir pour mesurer la distance entre le capteur et l'obstacle.  
Mesure de la distance en cours.

Distance (cm) :  
30  
Fin des mesures.

☒ Défilement automatique ☐ Afficher l'horodatage

Nouvelle ligne

9600 baud

Effacer la sortie