

```
//Ce programme étudie le fonctionnement du capteur ultrason; T=transmetteur; R=recepteur
```

```
//zone de définition des constantes ou des variables
```

```
int trig=12;           //la borne Trig du capteur est sur la broche 12
int echo=11;          //la borne Echo du capteur est sur la borne 11
long duree;           //la variable durée est un nombre entier; elle donne la durée que met le signal US
                      //pour faire l'aller-retour en microsecondes
long distance;        //la variable distance est un nombre entier; elle donne la distance, en cm, séparant l'obstacle du capteur
```

```
//zone qui contient les instructions d'initialisation
```

```
void setup()
{
  pinMode(trig, OUTPUT); //définit la borne 12 comme une sortie
  pinMode(echo, INPUT);  //définit la borne 11 comme une entrée
  Serial.begin(9600);    //vitesse de communication entre le PC et l'arduino
}
```

```
//zone qui contient les instructions du programme qui seront répétées indéfiniment
```

```
void loop()
{
  digitalWrite(trig,HIGH); //le signal part de l'émetteur
  delayMicroseconds(10);   //impulsion brève de 10 microsecondes
  digitalWrite(trig,LOW);  //fin de l'impulsion
  duree=pulseIn(echo,HIGH); //la variable durée (en microsecondes) est donnée lorsque le récepteur reçoit le signal
  distance=duree*0.017;    //calcul de la distance à partir de la valeur de la durée
  Serial.print("distance en cm:"); //texte affiché dans le moniteur série
  Serial.println(distance); //valeur de la distance affichée dans le moniteur série puis retour à la ligne
  delay(1000);             //attente de 1 s avant le prochain envoi de la salve
}
```

```
//Ce programme fait fonctionner des DEL de différentes couleurs suivant la distance séparant le capteur de l'obstacle
```

```
//zone de définition des constantes ou des variables
```

```
int delR=4;           //del rouge sur la borne 4
int delV=2;           //del verte sur la borne 2
int delO=3;           //del orange sur la borne 3
int trig=12;          //la borne Trig du capteur est sur la broche 12
int echo=11;          //la borne Echo du capteur est sur la borne 11
long duree;           //la variable durée est un nombre entier; elle donne la durée que met le signal US
//pour faire l'aller-retour en microsecondes
long distance;        //la variable distance est un nombre entier; elle donne la distance, en cm, séparant l'obstacle du capteur
```

```
//zone qui contient les instructions d'initialisation
```

```
void setup()
{
  pinMode(delR, OUTPUT); //définit la broche 4 comme une sortie
  pinMode(delV, OUTPUT); //définit la broche 2 comme une sortie
  pinMode(delO, OUTPUT); //définit la broche 3 comme une sortie
  pinMode(trig, OUTPUT); //définit la borne 12 comme une sortie
  pinMode(echo, INPUT);  //définit la borne 11 comme une entrée
  Serial.begin(9600);    //vitesse de communication entre le PC et l'arduino
}
```

```
//zone qui contient les instructions du programme qui seront répétées indéfiniment
```

```
void loop()
{
  digitalWrite(trig,HIGH); //le signal part de l'émetteur
  delayMicroseconds(10);   //impulsion brève de 10 microsecondes
  digitalWrite(trig,LOW);  //fin de l'impulsion
  duree=pulseIn(echo,HIGH); //la variable durée (en microsecondes) est donnée lorsque le récepteur reçoit le signal
  distance=duree*0.017;    //calcul de la distance à partir de la valeur de la durée
  Serial.print("distance en cm:"); //texte affiché dans le moniteur série
  Serial.println(distance); //valeur de la distance affichée dans le moniteur série puis retour à la ligne
  delay(1000);             //attente de 1 s avant le prochain envoi de la salve
  if(distance>40)          //si la distance est supérieure à 40 cm
  {
    digitalWrite(delV,HIGH); //la del verte s'allume
    digitalWrite(delR,LOW);  //la del rouge est éteinte
    digitalWrite(delO,LOW);  //la del orange est éteinte
  }
  else if(distance>=20&&distance<=40) //si la distance est comprise entre 20 et 40 cm
  {
    digitalWrite(delV,LOW);  //la del verte est éteinte
    digitalWrite(delR,LOW);  //la del rouge est éteinte
    digitalWrite(delO,HIGH); //la del orange s'allume
  }
  else                      //si la distance est inférieure à 20cm
  {
    digitalWrite(delV,LOW);  //la del verte est éteinte
    digitalWrite(delR,HIGH); //la del rouge s'allume
    digitalWrite(delO,LOW);  //la del orange est éteinte
  }
}
```

```
//Ce programme fait fonctionner un buzzer suivant la distance séparant le capteur de l'obstacle
```

```
//zone de définition des constantes ou des variables
```

```
int buzzer=3;           //buzzer sur la borne 3
int trig=12;           //la borne Trig du capteur est sur la broche 12
int echo=11;          //la borne Echo du capteur est sur la borne 11
long duree;           //la variable durée est un nombre entier; elle donne la durée que met le signal US
                       //pour faire l'aller-retour en microsecondes
long distance;        //la variable distance est un nombre entier; elle donne la distance, en cm, séparant l'obstacle du capteur
```

```
//zone qui contient les instructions d'initialisation
```

```
void setup()
{
  pinMode(trig, OUTPUT);           //définit la borne 12 comme une sortie
  pinMode(echo, INPUT);           //définit la borne 11 comme une entrée
  Serial.begin(9600);             //vitesse de communication entre le PC et l'arduino
}
```

```
//zone qui contient les instructions du programme qui seront répétées indéfiniment
```

```
void loop()
{
  digitalWrite(trig,HIGH);        //le signal part de l'émetteur
  delayMicroseconds(10);          //impulsion brève de 10 microsecondes
  digitalWrite(trig,LOW);         //fin de l'impulsion
  duree=pulseIn(echo,HIGH);       //la variable durée (en microsecondes) est donnée lorsque le récepteur reçoit le signal
  distance=duree*0.017;           //calcul de la distance à partir de la valeur de la durée
  Serial.print("distance en cm:"); //texte affiché dans le moniteur série
  Serial.println(distance);       //valeur de la distance affichée dans le moniteur série puis retour à la ligne
  delay(1000);                    //attente de 1 s avant le prochain envoi de la salve
  if(distance>40)                 //si la distance est supérieure à 40 cm, pas de son
  {
  }
  else if(distance>=20&&distance<=40) //si la distance est comprise entre 20 et 40 cm
  {
    tone(buzzer,494,100);         //fonctionnement du buzzer à la fréquence 494 Hz pendant 0,1s
    delay(100);                  //attente de 100 ms
    tone(buzzer,494,100);         //fonctionnement du buzzer à la fréquence 494 Hz pendant 0,1s
    delay(100);                  //attente de 100 ms
  }
  else                            //si la distance est inférieure à 20cm
  {
    tone(buzzer,200,1000);        //fonctionnement du buzzer à la fréquence 200 Hz pendant 0,1s
  }
}
```

