Capteurs et Pont en H

I - Capteurs



1. Capteurs ultrason

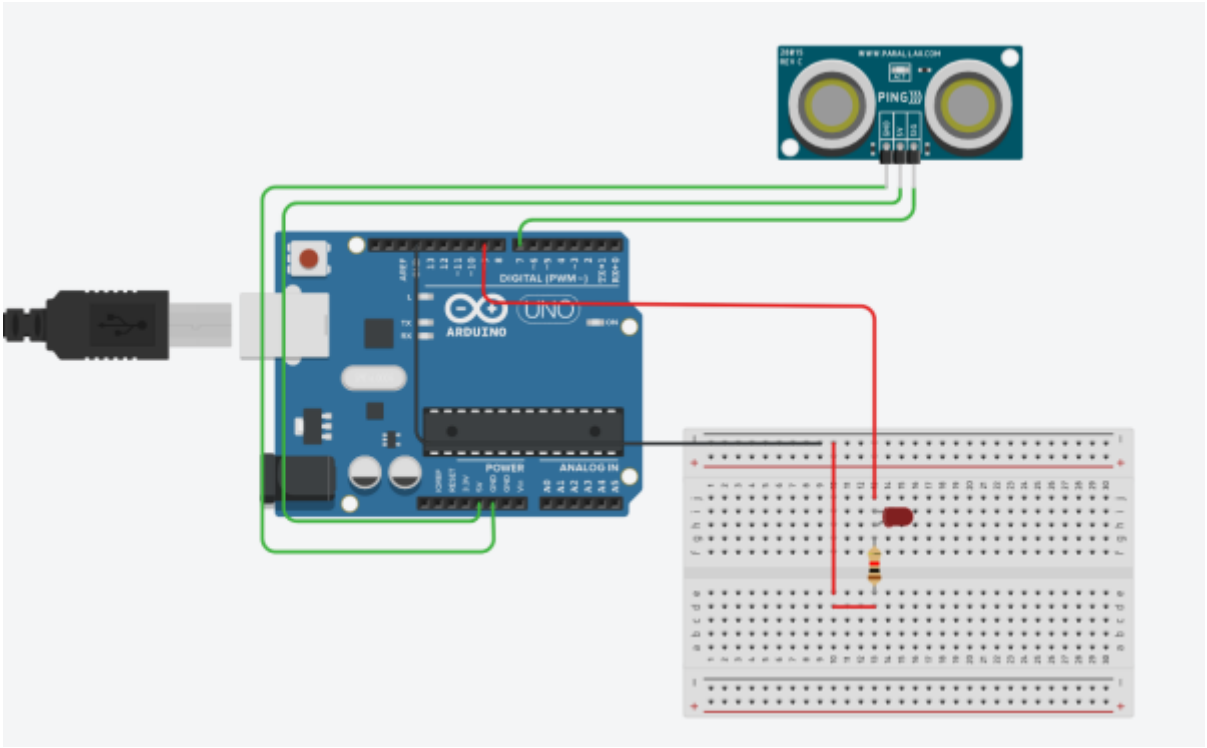
Pour cela, on va se servir de la bibiothèque “NewPing”, qui se trouve sur : https://playground.arduino.cc/Code/NewPing/

Installez-la et renseignez-vous sur les méthodes que propose cette bibliothèque.

Maintenant, réalisez un circuit, avec une LED et un capteur ultrason qui, lorsque le capteur capte un mouvement, allume la LED pour 1 seconde.

```
int pinTrigger1 = 7;
int pinEcho1 = 8;
int pinLed1 = 9;
float distance_max = 200;
boolean retour1 = 1;
NewPing sonar1(pinTrigger1, pinEcho1, distance_max);

void setup() {
  pinMode(pinLed1, OUTPUT);
  Serial.begin(9600);
  delay(10);
}

void loop() {
  retour1 = sonar1.ping();

  if (retour1 !=0 and allume1 == 0) {
    digitalWrite(pinLed1, HIGH);
    allume1 = 1;
    delay(1000);
  }
  if (allume1 == 1 ) {
```

```
    digitalWrite(pinLed1, LOW);
    allume1 = 0;
    delay(1000);
  }
}
```



La fonction delay() a un inconvénient majeur qui fait qu'elle est rarement utilisée en pratique : lors de son exécution, elle ordone à la carte de ne plus rien faire jusqu'à ce que le temps soit écoulé, et donc, on peut rater du mouvement, ou pire, si il y a deux capteurs, le système ne pourra fonctionner. On utilise alors la fonction millis() qui renvoit le temps en millisecondes (fonctionne de manière similaire que la fonction time en python). Essayez de modifier le programme précédent en utilisant la fonction millis().

```
float t1 = 0;
int allume1 = 0;
int pinTrigger1 = 7;
int pinEcho1 = 8;
int pinLed1 = 9;
float distance_max = 200;
boolean retour1 = 1;
NewPing sonar1(pinTrigger1, pinEcho1, distance_max);

void setup() {
  pinMode(pinLed1, OUTPUT);
  Serial.begin(9600);
  delay(10);
}
```

```
void loop() {
  retour1 = sonar1.ping();

  if (retour1 !=0 and allume1 == 0) {
    digitalWrite(pinLed1, HIGH);
    t1 = millis();
    allume1 = 1;
  }
  if (allume1 == 1 and millis()-t1>2000) {
    digitalWrite(pinLed1, LOW);
    allume1 = 0;
  }
}
```

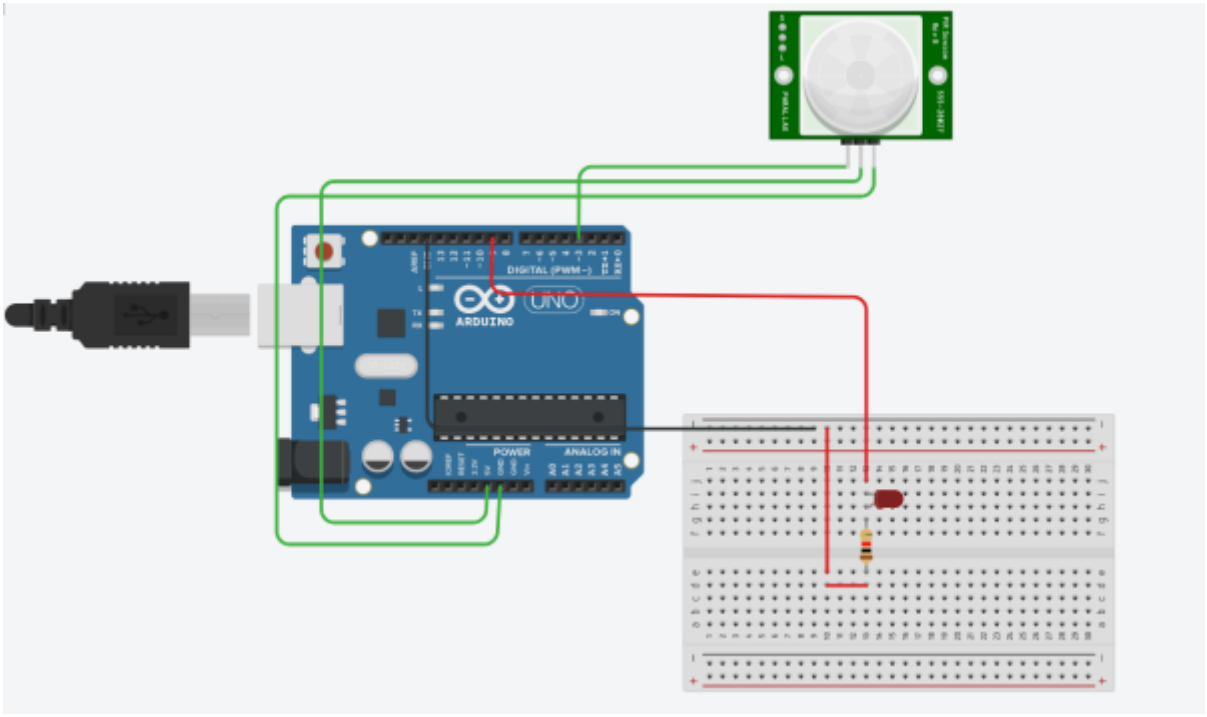


2. Capteurs infrarouge

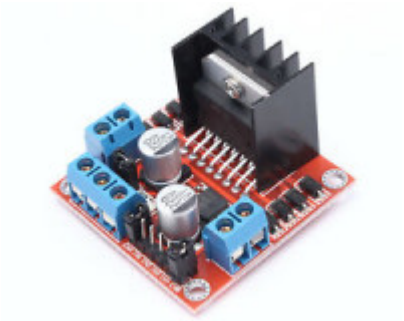
Ici, nous n'avons pas besoin d'importer de librairie. Adaptez le programme précédent pour un capteur infrarouge (bien sûr, recherchez les références du modèle sur Internet !).

```
float t1 = 0;
int allume1 = 0;
int pinIR1 = 3;
int pinLed1 = 9;
boolean retour1 = 1;

void setup() {
  pinMode(pinLed1, OUTPUT);
  pinMode(pinIR1, INPUT);
  Serial.begin(9600);
  delay(10);
}

void loop() {
  retour1 = digitalRead(pinIR1);
  Serial.println(retour1);

  if (retour1 !=0 and allume1 == 0) {
```

```
    digitalWrite(pinLed1, HIGH);
    t1 = millis();
    allume1 = 1;
  }
  if (allume1 == 1 and millis()-t1>2000) {
    digitalWrite(pinLed1, LOW);
    allume1 = 0;
  }
}
```

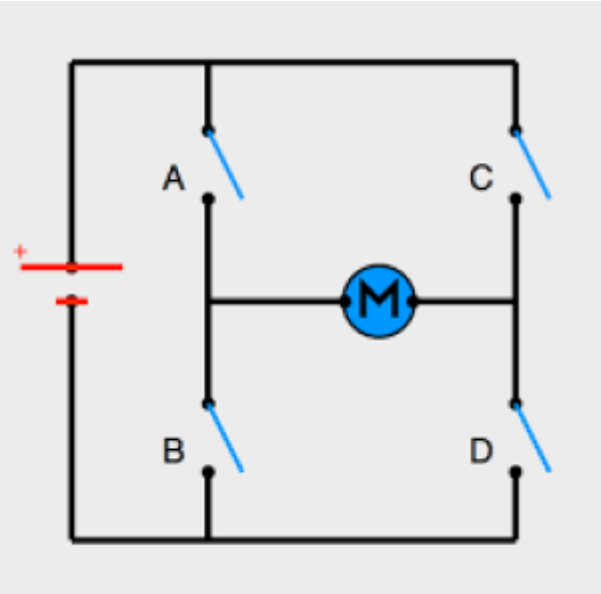


II - Ponts en H



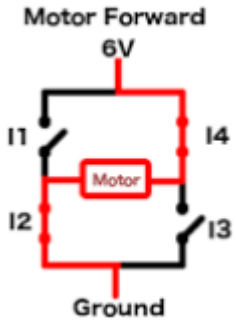
1. Principe

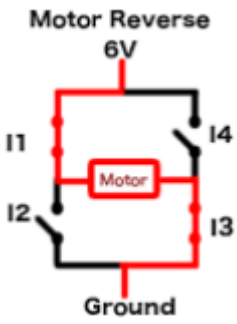
Un pont en H permet de faire tourner un moteur dans le sens des aiguilles d'une montre ou dans le sens inverse des aiguilles d'une montre en inversant les tensions et courants.



Pour cela, il faut fermer deux broches comme dans le schéma ci-contre.

Si on fermait les broches A et C (respectivement B et D) ou A et B (respectivement C et D), soit le moteur ne fonctionne pas, soit on crée un court-circuit.



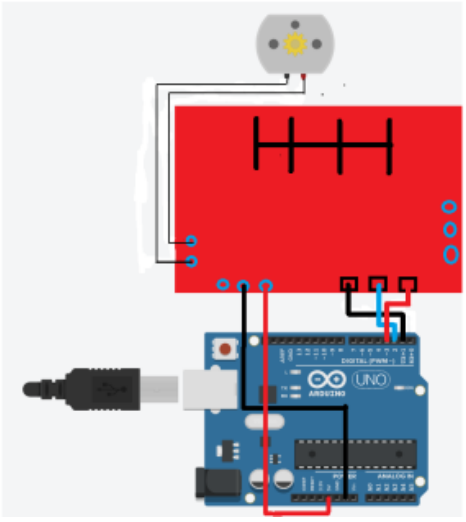


2. Partie Arduino

Code de base

```
int enA = 2; // Pin pour prévenir qu'on branche un moteur
int in1 = 3; // Pin pour le sens du moteur
int in2 = 4;

void setup() {
    pinMode(enA, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    digitalWrite(enA, HIGH); // On prévient qu'on doit activer le moteur A
}
```

Branchement



```
void loop() {
    digitalWrite(in1, HIGH); // On
fait tourner le moteur A
    digitalWrite(in2, LOW);
}
```

3. TP

On vous demande d'effectuer plusieurs taches :

1. Faire tourner le moteur dans les deux sens.
2. Inverser le sens de rotation du moteur après un laps de temps.
3. Faire un montage qui permet d'effectuer le programme précédent mais en séparant les chaînes d'énergie et de contrôle.

From:
https://wiki.centrale-med.fr/egab/ - **E-Gab**

Permanent link:
https://wiki.centrale-med.fr/egab/formation:capteurs_et_pont_h

Last update: **13/10/2020 17:38**