

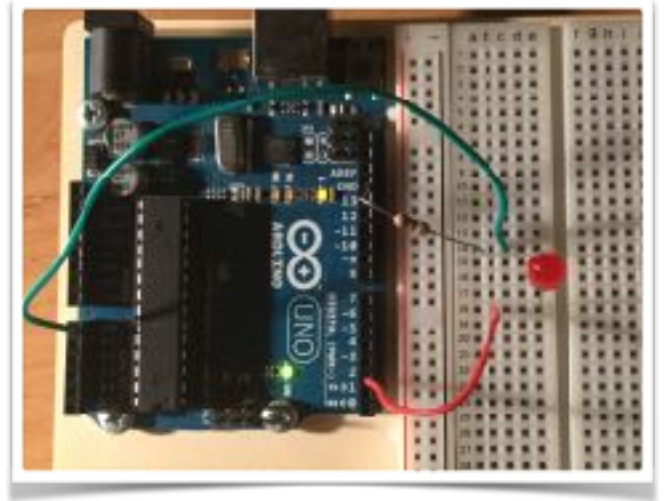
# FAIRE COMMUNIQUER ARDUINO ET P5JS !

Nous allons réaliser un petit montage avec une diode. L'objectif de notre mission sera de la commander à partir d'une application réalisée sous P5Js. Tous les programmes utiles sont fournis dans le KITARDUINOP5JS

## Mission n°1 : Réaliser le montage suivant

### Cahier des charges :

- Relier , avec un fil rouge, la borne n°2 et la borne + de la diode.
- Relier la borne - de la diode à une résistance de 220 Ohms.
- Relier cette résistance à la borne GND
- Relier un fil vert entre la borne analogique A0 et la borne - de la diode.



## Mission n°2 : Réaliser le programme suivant

```

// Arduino
// cette application va communiquer entre p5js et Arduino

int led= 2; // une led est reliée à la borne 2 avec une résistance de 220 Ohms
int valeurDuPortSerie; // a variable to read incoming serial data into

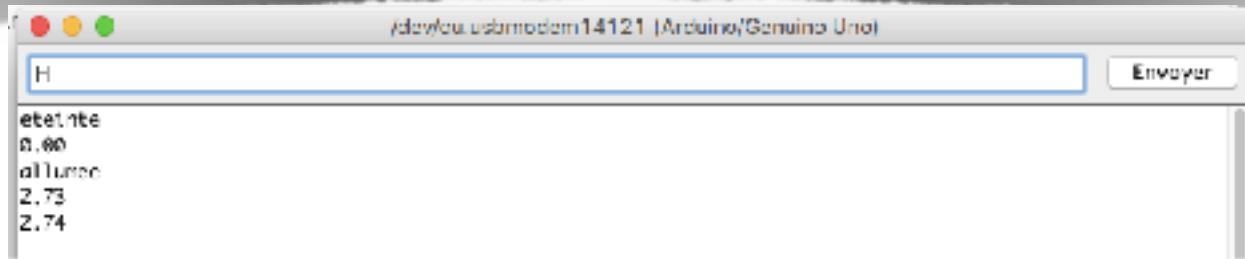
void setup() {
  // initialisation du port Série
  Serial.begin(9600);
  // mettre le pin 2 en mode sortie
  pinMode(led, OUTPUT);
}

void loop() {
  // on regarde s'il y a une donnée sur le port série
  if (Serial.available() > 0) {
    // si oui, nous lisons la dernière valeur du port série read the oldest byte in the serial buffer
    valeurDuPortSerie = Serial.read();
    // si cette valeur correspond à la lettre H , on allume la diode
    if (valeurDuPortSerie == 'H') {
      digitalWrite(led, HIGH);
      Serial.println("allumee");
    }
    // si cette valeur correspond à la lettre L, on éteint la diode
    if (valeurDuPortSerie == 'L') {
      digitalWrite(led, LOW);
      Serial.println("eteinte");
    }

    // lire la valeur sur le pin analogique 0
    int valeurDuPinAnalogique = analogRead(A0);
    // Convertir cette valeur comprise entre 0 et 1023 en une tension entre 0 et 5 V
    float tension = valeurDuPinAnalogique * (5.0 / 1023.0);
    // Envoyer cette valeur sur le port série
    Serial.println(tension);
  }
}

```

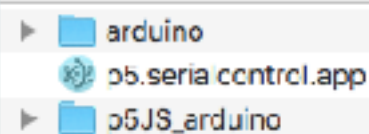
Après avoir choisi le bon port, téléverser votre programme dans la carte Arduino.  
 Tester votre programme en cliquant sur le moniteur série (en haut à droite)  
 Essayer d'envoyer la lettre H, la diode doit s'allumer  
 Essayer d'envoyer la lettre L, la diode doit s'éteindre.



## Mission n°4 : programmer p5JS

### Cahier des charges :

- Quitter le programme Arduino.
- Laisser brancher la carte.
- Lancer le programme **p5.serialcontrol.app**



```

var portSerie; // Declaration de l'objet Port Serie
var nomDuPort = "/dev/cu.usbmodem14121"; // indiquer ici le nom exact du port série utilisé par
arduino
var MessageEnvoye = 'L'; // message envoye à la carte
function setup() {
  createCanvas(600, 400); // creation d'une fenetre
  background(100,100,100); // un joli fond gris
  portSerie = new p5.SerialPort(); // creation de l'objet portSerie
  portSerie.open(nomDuPort); // ouverture du port -
  portSerie.on('data', lectureDesDonnees); // passage d'une donnée par le port série
  fill(20,20,150); textSize(20); // une belle écriture bleue
  text("Cliquez pour allumer ou éteindre la diode", 100, 100);
}

function lectureDesDonnees() { // lecture et affichage des données sur le port Série.
  var caractere = portSerie.readLine(); // lecture ligne par ligne
  noStroke();
  if (caractere !== '' && caractere !== parseFloat(caractere)) {
    fill(100,100,100); rect(90,180,100,30); fill(255,0,0); // masque pour actualiser l'affichage
    text(caractere , 100,200);}

  if (caractere !== '' && caractere == parseFloat(caractere)) {
    fill(100,100,100); rect(180,180,80,30); fill(255,0,0); // masque
    text(caractere + " V", 200,200);}
}

function mouseReleased() { // Lorsque l'on clique sur la souris , le message envoyé change
  portSerie.write(MessageEnvoye);
  if (MessageEnvoye === 'L') { MessageEnvoye = 'H'; } else { MessageEnvoye = 'L'; }
}

```