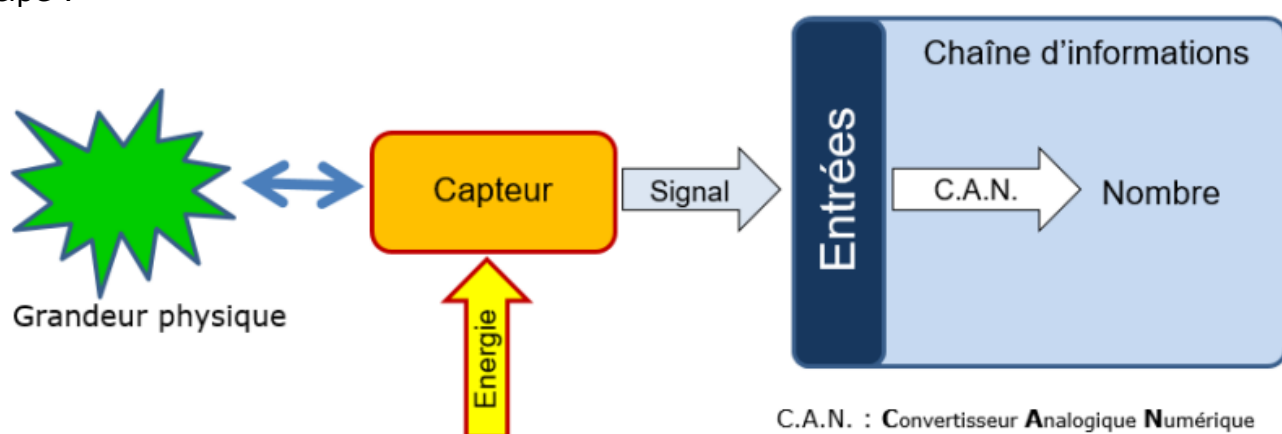


1 – Comment obtenir des informations sur notre environnement ?

La plupart des machines ont besoin d'obtenir des informations sur leur environnement. Exemples d'informations : Contact, distance, position dans l'espace (où je suis ?, comment suis-je orienté ?), température, hygrométrie, luminosité, pression, présence de gaz ou de fumées, tension électrique, présence d'une personne, comptage,...

Les machines fonctionnant presque toujours à l'électricité, ces informations doivent être reçues sous la forme d'un courant électrique pour y être traitées : c'est le rôle du capteur.

Principe :



Le capteur transforme une grandeur physique en signal électrique.
La carte électronique convertit le signal électrique en nombre.

2 – Nature du signal

2.1 Les capteurs tout-ou-rien

Seuls deux cas sont possibles : le capteur capte un phénomène ou pas.

Signal physique	Signal électrique	Valeur de l'entrée
absent	0V	0
présent	tension d'alimentation	1

Exemples : interrupteur, thermostat, détecteur de présence PIR,...

2.2 Les capteurs analogiques

Le signal électrique est une tension (ou un courant) qui varie en fonction de la grandeur mesurée. Attention : cette fonction n'est pas forcément linéaire.

Signal physique	Signal électrique	Valeur de l'entrée
minimal	0V	0
...	...	...
maximal	tension d'alimentation	1023

L'entrée peut prendre n'importe quelle valeur entière entre 0 et 1023. Cette valeur est proportionnelle à la tension d'entrée.

Exemple : un capteur de température analogique -dont la courbe est linéaire- connecté à une entrée de l'Arduino (donc alimenté en 5V) donnera :

- 0V pour 0 degré donc l'entrée vaudra 0;
- ...
- 2,5V pour 50 degrés donc l'entrée vaudra 511;
- ...
- 5V pour 100 degrés donc l'entrée vaudra 1023;

Pourquoi 1023 ? Parce que le convertisseur convertit la tension en un nombre binaire composé de 10 bits. 10 bits donnent 1024 (1023 plus le zéro qui compte pour un nombre) combinaisons possibles ($1024 = 2^{\text{puissance } 10}$).

2.3 Les capteurs numériques

Ils comportent leur propre processeur qui traduit la valeur mesurée en nombre. En général, ces capteurs sont interrogés par la carte via un protocole spécifique (liaison série, i2c, spi,...)

3 – Quelques capteurs courants et leur mise en œuvre

Je ne présente ici que les quelques capteurs dont vous pourriez avoir besoin. Il en existe évidemment des millions d'autres.

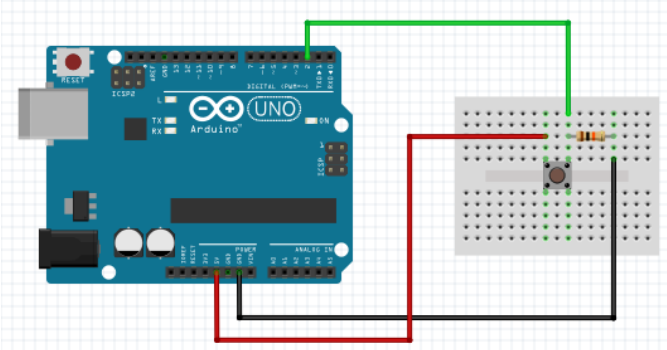
Précisions importantes :

- **vérifiez la tension d'alimentation du capteur AVANT de le brancher;**
- **un capteur devant être alimenté en 3,3V SERA DÉTRUIT par une alimentation 5V;**
- **un signal de 5V envoyé par le capteur DETRUIRA le microcontrôleur (en particulier le micro:bit) si celui-ci n'accepte que 3,3V sur ses entrées.**

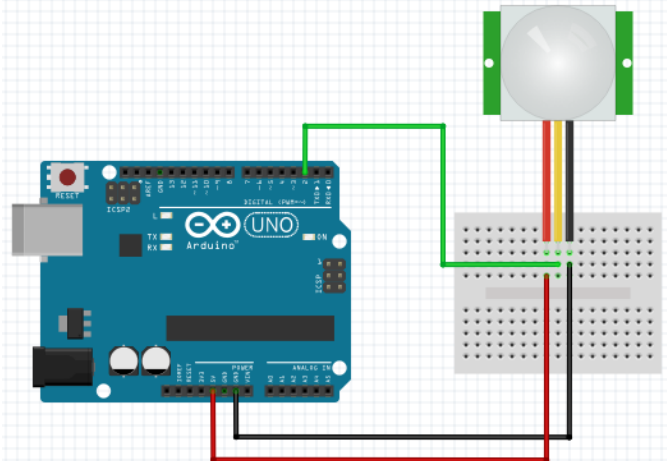
Les programmes sont donnés "bruts". Ils sont évidemment perfectibles (calcul de la moyenne d'une série de mesure pour améliorer la fiabilité par exemple). Pour comprendre le code, référez-vous au cours sur la programmation.

3.1 Les capteurs tout ou rien

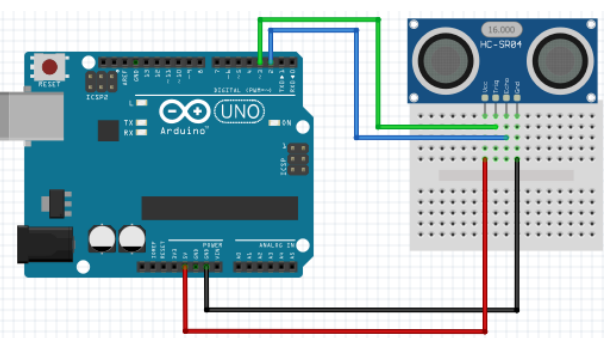
3.1.1 Les interrupteurs

Capteur	Montage	Code
	 La résistance à une valeur de l'ordre de 10K ohms.	<pre>int interrupteur; void setup() { pinMode(13, OUTPUT); pinMode(2, INPUT); } void loop() { interrupteur=digitalRead(2); if (interrupteur == HIGH) { digitalWrite(13, HIGH); } else { digitalWrite(13, LOW); } }</pre>

3.1.2 Le détecteur de présence infrarouge (PIR)

Capteur	Montage	Code
		<pre>int interrupteur; void setup() { pinMode(13, OUTPUT); pinMode(2, INPUT); } void loop() { interrupteur=digitalRead(2); if (interrupteur == HIGH) { digitalWrite(13, HIGH); } else { digitalWrite(13, LOW); } }</pre>

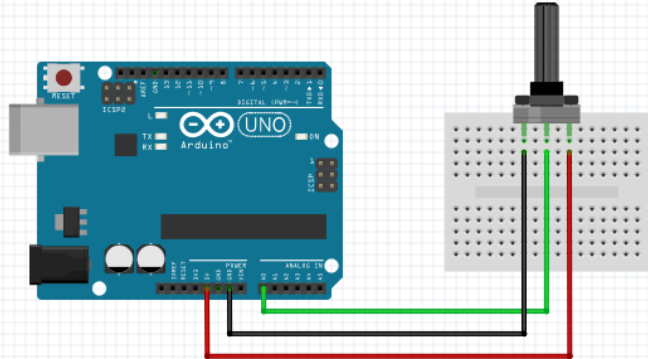
3.1.3 Le radar ultrason HC-SR04

Capteur	Montage	Code
		<pre>byte trig=3; byte echo=2; long temps; long mesure; void setup(){ pinMode(trig, OUTPUT); digitalWrite(trig, LOW); pinMode(echo, INPUT); Serial.begin(115200); } void loop(){ digitalWrite(trig, HIGH); delayMicroseconds(10); digitalWrite(trig,LOW); temps = pulseIn(echo,HIGH); }</pre>

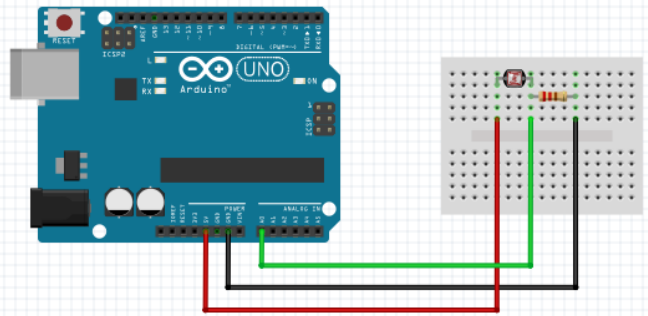
		<pre> mesure = temps/58; Serial.print("Distance en cm : "); Serial.println(mesure); delay(1000); } </pre>
--	--	---

3.2 Les capteurs analogiques

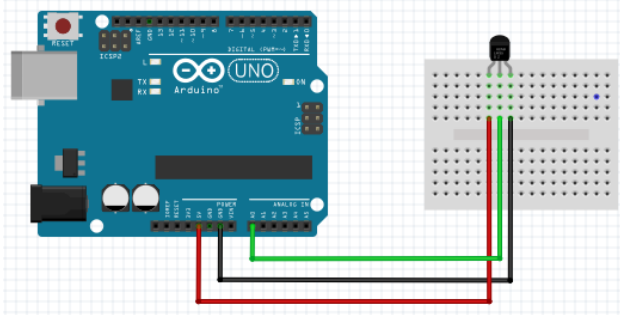
3.2.1 Le potentiomètre

Capteur	Montage	Code
		<pre> int potentiometre; void setup() { Serial.begin(115200); } void loop() { potentiometre=analogRead(0); Serial.println(potentiometre); delay(1000); } </pre>

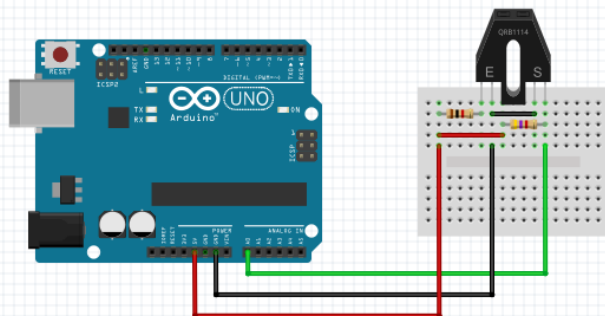
3.2.2 La photorésistance (LDR)

Capteur	Montage	Code
		<pre> int luminosite; void setup() { Serial.begin(115200); } void loop() { luminosite=analogRead(0); Serial.println(luminosite); delay(1000); } </pre>

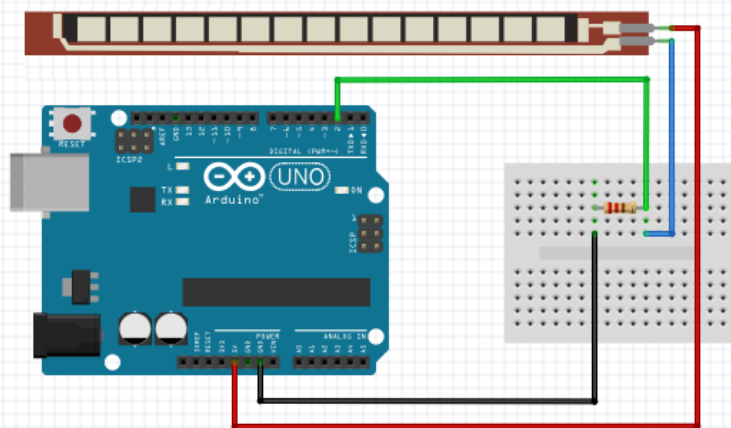
3.2.3 La thermistance LM35

Capteur	Montage	Code
		<pre> int mesure; float temperature_celcius; void setup() { Serial.begin(115200); } void loop() { mesure = analogRead(A0); temperature_celcius = mesure * (5.0 / 1023.0 * 100.0); Serial.println(temperature_celcius); delay(1000); } </pre>

3.2.4 Le détecteur de proximité infrarouge

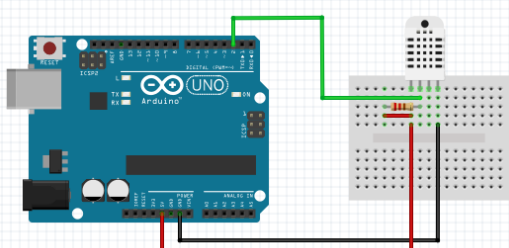
Capteur	Montage	Code
		<pre> int mesure; float distance; void setup() { Serial.begin(115200); } void loop() { mesure = analogRead(0); distance = (float)mesure * 5.0 / 1023.0; Serial.println(distance); delay(1000); } </pre>

3.2.5 Le capteur de force

Capteur	Montage	Code
		<pre> int force; void setup() { Serial.begin(115200); } void loop() { force=analogRead(0); Serial.println(force); delay(1000); } </pre>

3.3 Les capteurs numériques

3.3.1 Le thermomètre/hygromètre DHT22

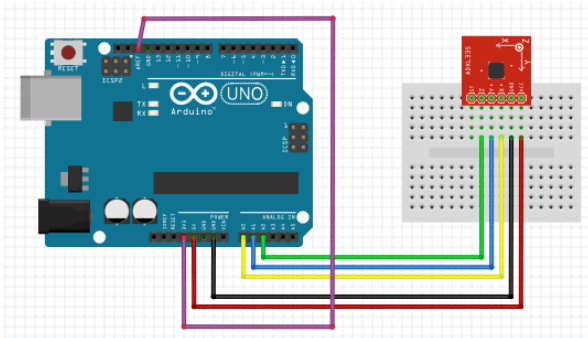
Capteur	Montage	Code
		<pre> #include "DHT.h" float hygrometrie; float temperature; DHT capteur(2, DHT22); void setup() { Serial.begin(115200); capteur.begin(); } void loop() { hygrometrie = capteur.readHumidity(); temperature = capteur.readTemperature(); Serial.print("Humidite: "); Serial.print(hygrometrie); Serial.print(" %\t"); Serial.print("Temperature: "); Serial.print(temperature); } </pre>

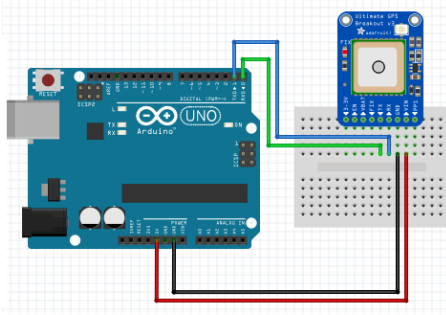
```

Serial.print(" *C ");
delay(2000);
}

```

3.3.2 Les accéléromètres

Capteur	Montage	Code
		<pre> int mesure_x; int mesure_y; int mesure_z; long x_calibre; long y_calibre; long z_calibre; float acceleration_x; float acceleration_y; float acceleration_z; void setup() { analogReference(EXTERNAL); Serial.begin(115200); } void loop() { mesure_x = analogRead(2); mesure_y = analogRead(1); mesure_z = analogRead(0); x_calibre = map(mesure_x, 0, 1023, -3000, 3000); y_calibre = map(mesure_y, 0, 1023, -3000, 3000); z_calibre = map(mesure_z, 0, 1023, -3000, 3000); acceleration_x = x_calibre / 1000.0; acceleration_y = y_calibre / 1000.0; acceleration_z = z_calibre / 1000.0; Serial.print("X, Y, Z : "); Serial.print(mesure_x); Serial.print(", "); Serial.print(mesure_y); Serial.print(", "); Serial.print(mesure_z); Serial.print(" :: "); Serial.print(acceleration_x,0); Serial.print("G, "); Serial.print(acceleration_y,0); Serial.print("G, "); Serial.print(acceleration_z,0); Serial.println("G"); delay(200); } </pre>

Capteur	Montage	Code
		<pre> #include <TinyGPS++.h> #include <SoftwareSerial.h> char data; double latitude; double longitude; double alt; //altitude double vitesse; unsigned long nbre_sat; TinyGPSPlus gps; SoftwareSerial GPS(2, 3); void setup() { GPS.begin(9600); Serial.begin(115200); } void loop() { while (GPS.available()) { data = GPS.read(); Serial.print(data); if (gps.location.isUpdated()) { latitude = gps.location.lat(); longitude = gps.location.lng(); alt = gps.altitude.meters(); vitesse = gps.speed.kmph(); nbre_sat = gps.satellites.value(); Serial.println(" "); Serial.println("----- DONNEES GPS DECODEES -----"); Serial.print("Latitude="); Serial.println(latitude); Serial.print("Longitude="); Serial.println(longitude); Serial.print("Altitude (m) ="); Serial.println(alt); Serial.print("Vitesse (km/h)="); Serial.println(vitesse); Serial.print("Nbre de satellites="); Serial.println(nbre_sat); } } } </pre>