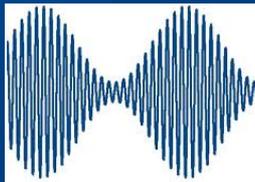


1ère STI2D

La nature de l'information



acquérir



traiter



communiquer

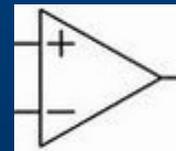
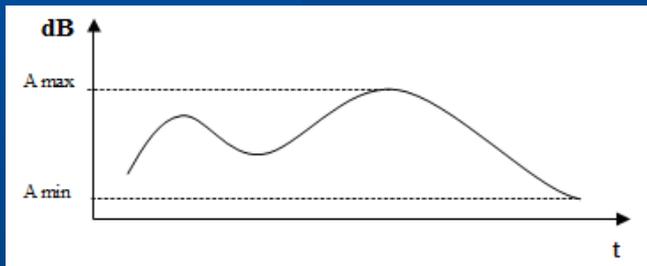


1ère STI2D

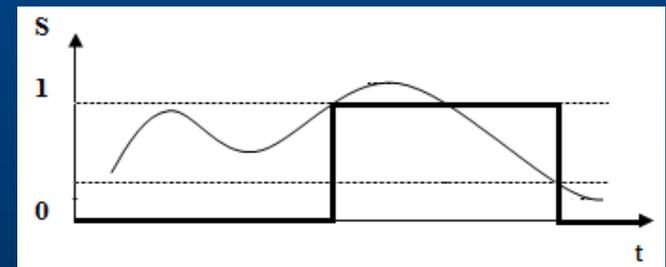
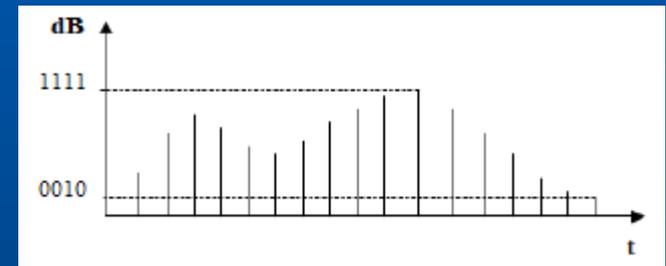
La nature de l'information



Analogique



Numérique



Logique

1ère STI2D

La nature de l'information



- **Analogique** : le signal varie de manière continue au cours du temps (mesure d'une grandeur physique).
- **Numérique** : la représentation du signal varie de façon discrète (ie : discontinue) dans une liste de valeurs.
- **Logique** : le signal est convertit dans un état binaire qui ne prend que deux valeurs, notées par convention 0 et 1 (logique Tout ou Rien, TOR).

1ère STI2D

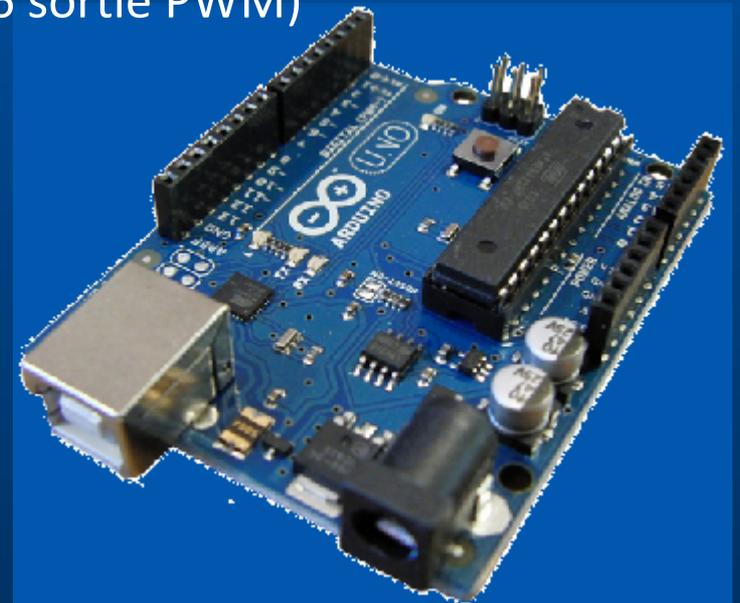
La nature de l'information



Nano automate Arduino

Modèle UNO

- Microcontrôleur : ATmega328
- Tension de fonctionnement : 5V
- Broches E/S numériques : 14 (dont 6 sortie PWM)
- Broches d'entrées analogiques 6
- Intensité maxi : 40 mA
- Mémoire Programme Flash : 32 KB
- Mémoire SRAM : 2 KB
- Mémoire EEPROM : 1 KB
- Vitesse d'horloge : 16 MHz



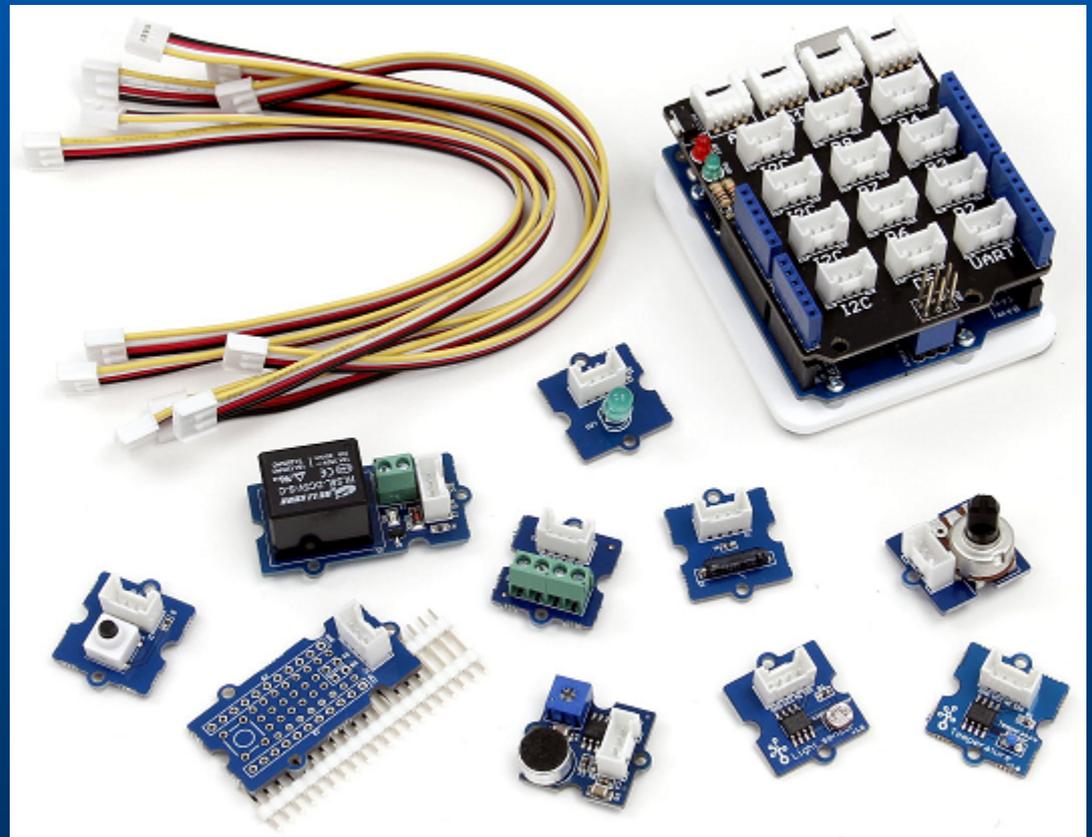
1ère STI2D

La nature de l'information



Capteurs et shield

Les shields sont des cartes d'interfaces qui s'empilent sur des cartes Arduino.

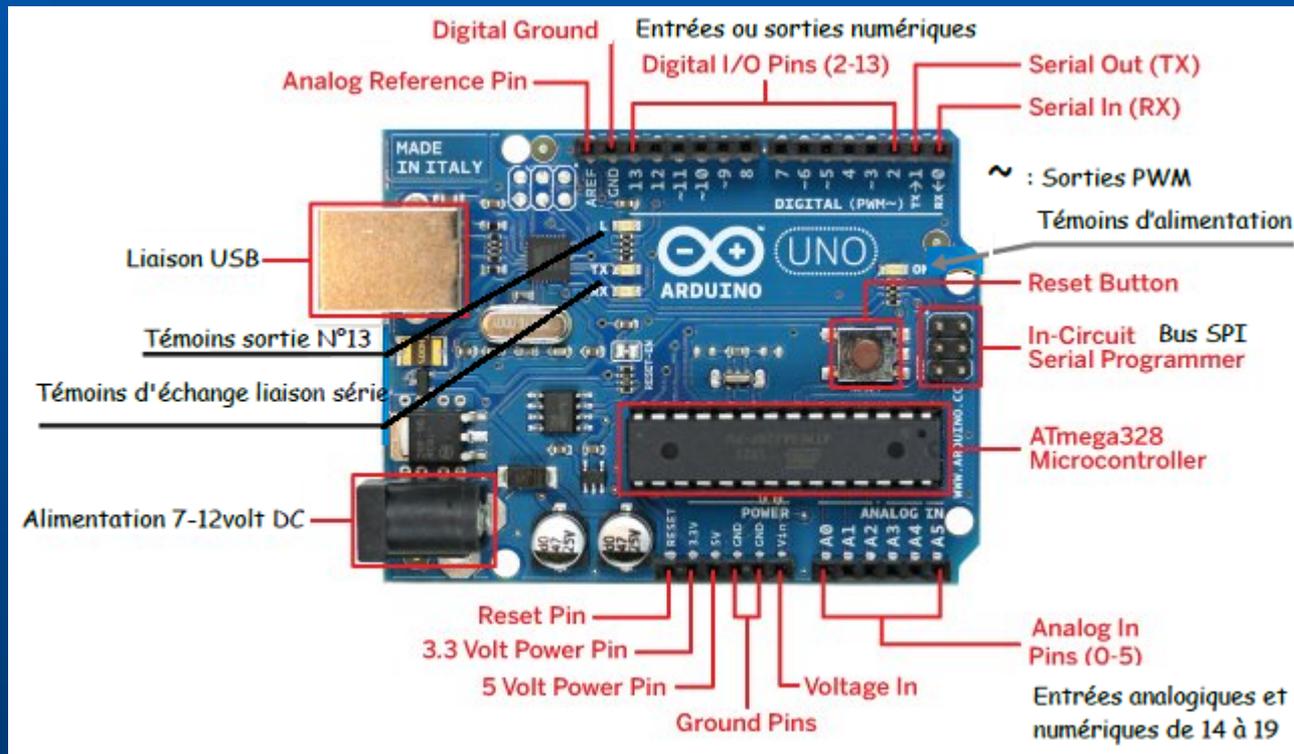


1ère STI2D

La nature de l'information



Arduino UNO



1ère STI2D

La nature de l'information



Entrées et sorties numériques

Les 14 broches numériques de la carte UNO (numérotées des 0 à 13) peuvent être utilisées comme entrée ou sortie numérique :

instructions `pinMode()`, `digitalWrite()` et `digitalRead()`

Ces broches fonctionnent en 5V avec un maximum de 40mA.

Les résistances internes des broches :

- s'activent avec l'instruction `digitalWrite(broche, HIGH)`
- se désactivent avec l'instruction `digitalWrite(broche, LOW)`.

1ère STI2D

La nature de l'information



Broches spécialisées

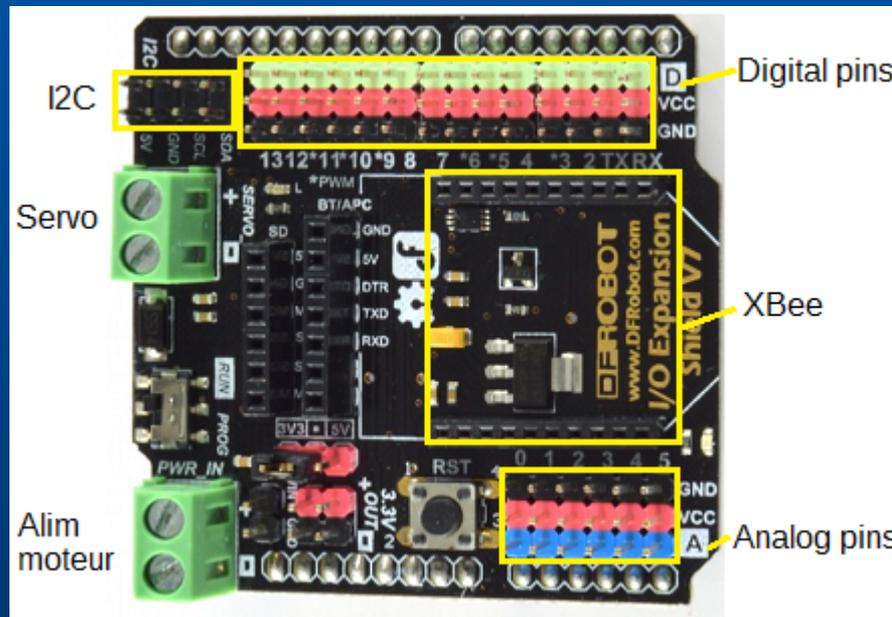
- Communication Série : Broches 0 (RX) et 1 (TX)
- 6 sorties PWM : Broches 3, 5, 6, 9, 10, et 11
résolution 8 bits : 2^8 [0 - 255]
instruction `analogWrite()`
- 6 entrées analogiques (numérotées de 0 à 5) : A0 à A5
résolution 10 bits : 2^{10} [0 - 1023]
instruction `analogRead()`



1ère STI2D

La nature de l'information

Arduino I/O Expansion Shield



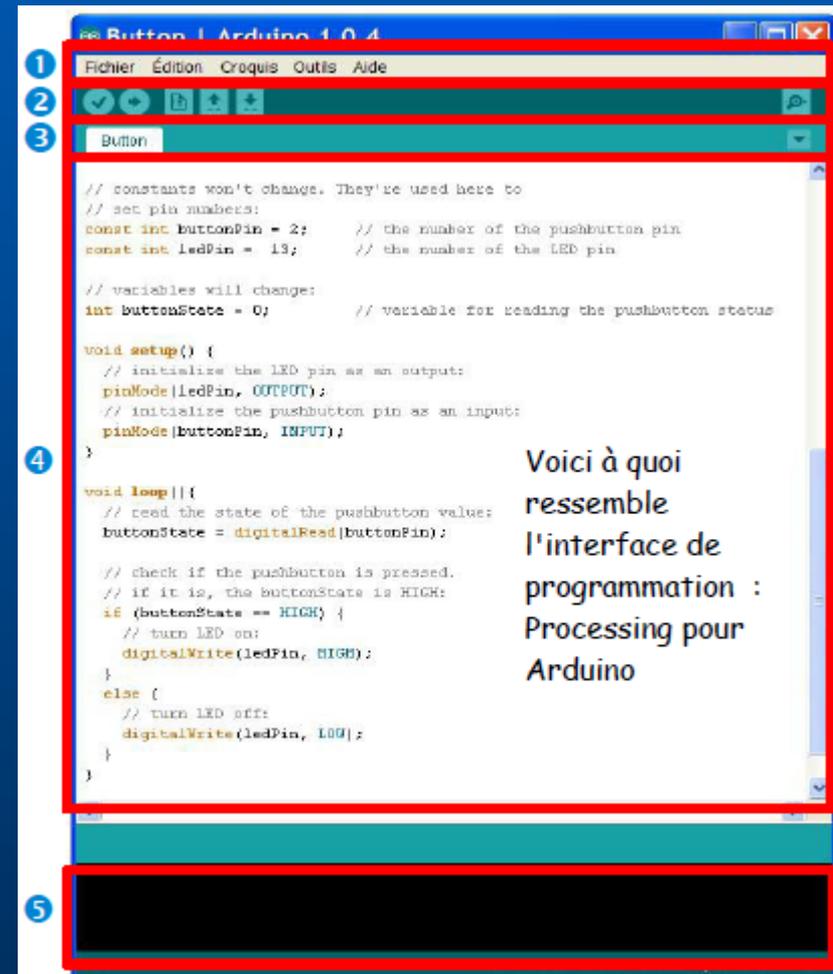
1ère STI2D

La nature de l'information



Interface Arduino

- 1) un menu
- 2) une barre d'actions
- 3) un ou plusieurs onglets correspondant aux "sketchs"
- 4) une fenêtre de programmation
- 5) une console qui affiche les informations et erreurs de compilation et de téléversement du programme



1ère STI2D

La nature de l'information



Interface de programmation

```
//Commentaires (1 ligne)  
/* zone de commentaire*/
```

- 1) la partie déclarative, variables et constantes (optionnelle).
- 2) la partie initialisation et configuration .
Déclaration des entrées/sorties.
- C'est la fonction `setup () {}`
- 3) la partie principale qui s'exécute en boucle :
- C'est la fonction `loop {}`

```
Button | Arduino 1.0.4  
Fichier Édition Croquis Outils Aide  
Button  
// constants won't change. They're used here to  
// set pin numbers:  
const int buttonPin = 2; // the number of the pushbutton pin  
const int ledPin = 13; // the number of the LED pin  
  
// variables will change:  
int buttonState = 0; // variable for reading the pushbutton status  
  
void setup() {  
  // initialize the LED pin as an output:  
  pinMode(ledPin, OUTPUT);  
  // initialize the pushbutton pin as an input:  
  pinMode(buttonPin, INPUT);  
}  
  
void loop() {  
  // read the state of the pushbutton value:  
  buttonState = digitalRead(buttonPin);  
  
  // check if the pushbutton is pressed.  
  // if it is, the buttonState is HIGH:  
  if (buttonState == HIGH) {  
    // turn LED on:  
    digitalWrite(ledPin, HIGH);  
  }  
  else {  
    // turn LED off:  
    digitalWrite(ledPin, LOW);  
  }  
}
```

1ère STI2D

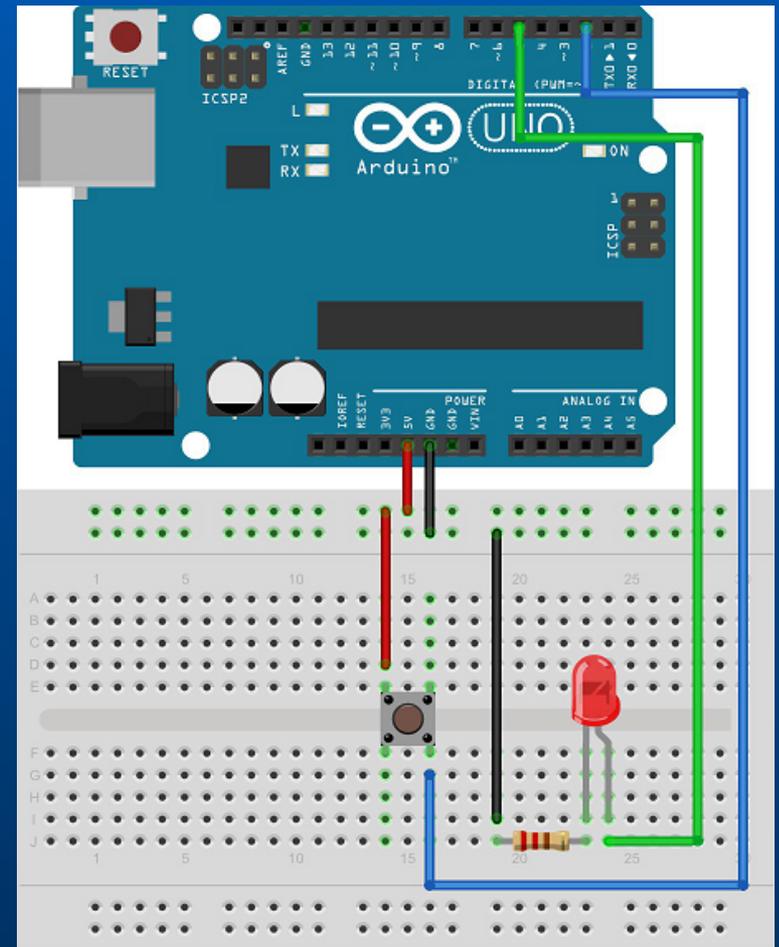
La nature de l'information



Exemple 1 : lecture numérique

// Allumage LED par bouton

```
void setup()
{
  pinMode(2, INPUT); // bouton en entrée 2
  pinMode(5, OUTPUT); // LED en sortie 5
}
void loop()
{
  if ( digitalRead(2) == HIGH ) // capteur TOR
    digitalWrite(5, HIGH); // actionneur
  else
    digitalWrite(5, LOW); // éteindre
  delay(100); // attente 100 ms
}
```



1ère STI2D

La nature de l'information



Exemple 2 : lecture analogique

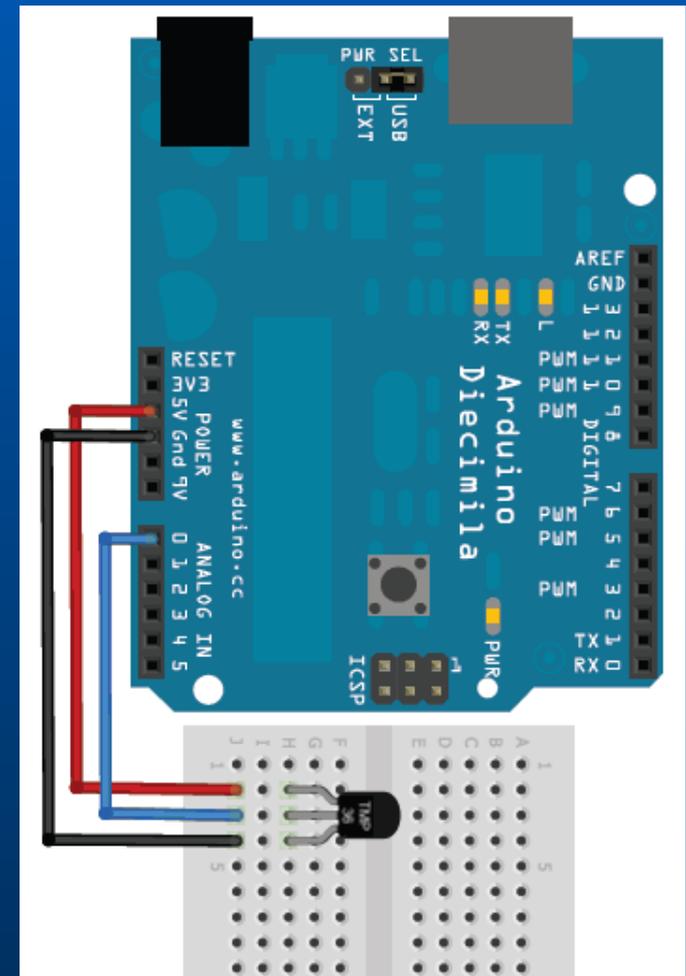
```
// lecture de température
```

```
void setup()
```

```
{  
    //initialisation vitesse liaison série à 9600 bauds  
    Serial.begin(9600);  
}
```

```
void loop()
```

```
{  
    int t = analogRead(A0);    // lecture entrée analogique  
  
    Serial.println(t);        // affichage valeur numérique  
    delay(100);              // attente 100 ms  
}
```



1ère STI2D

La nature de l'information



Exemple 3 : MLI

```
const int moteur = 3; // broche 3 en sortie
```

```
setup
```

```
{  
}
```

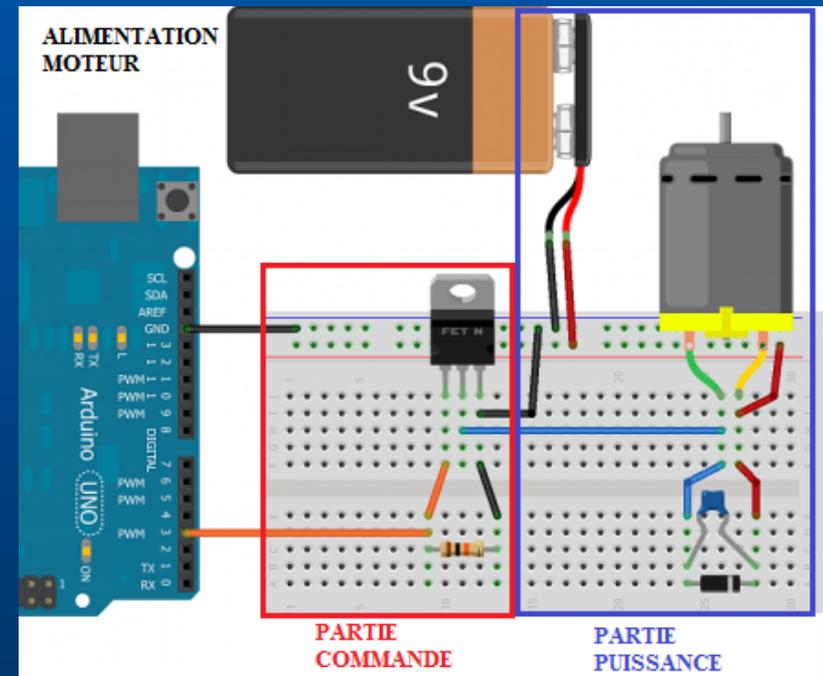
```
void loop()
```

```
{
```

```
  for (int i(0) ; i < 256 ; i++) { // accélération  
    analogWrite(moteur, i);  
    delay(50); // attente 50 ms  
  }
```

```
  for (int i(0) ; i < 256 ; i++) { // décélération  
    analogWrite(moteur, 255 - i);  
    delay(50); // attente 50 ms  
  }
```

```
}
```



1ère STI2D

La nature de l'information



Conclusions

Numérique et Logique :

```
pinMode(n° broche , mode);  
int etat = digitalRead(n° broche)  
digitalWrite(n° broche , etat);
```

```
mode= [ INPUT , OUTPUT ]
```

```
etat = [ HIGH , LOW ]
```

Analogique :

```
int valeur = analogRead(n° broche) ;  
analogWrite(n° broche, valeur) ;
```

```
valeur = [ 0 .. 1023 ]
```