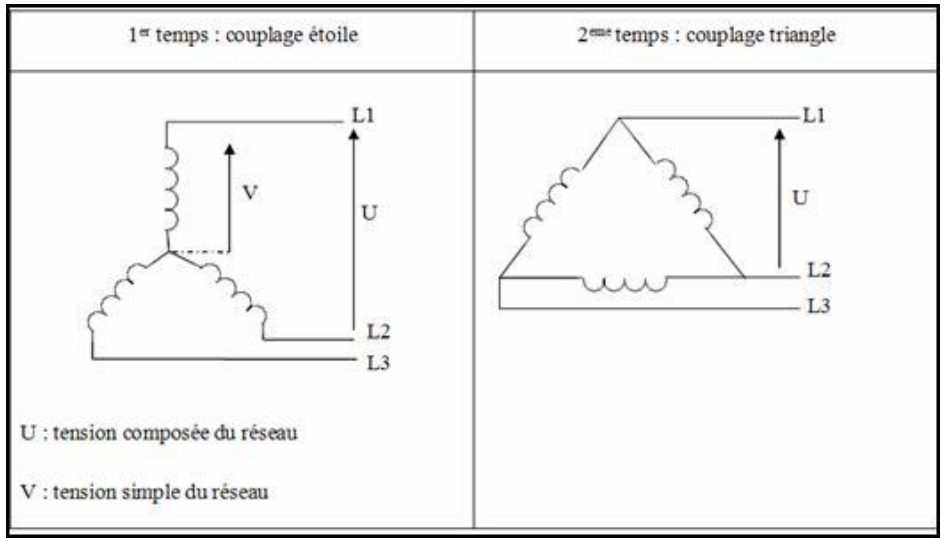
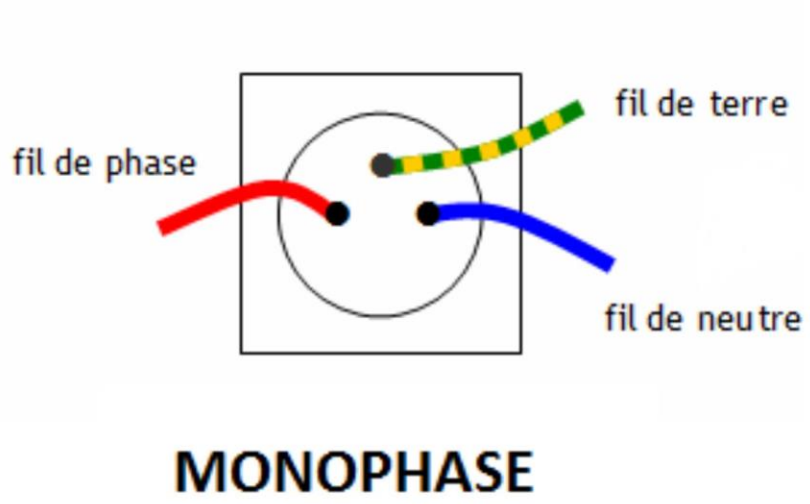
# MOTEURS

Formateur: Gauthier PINEDA

# 0/ Triphasé vs monophasé



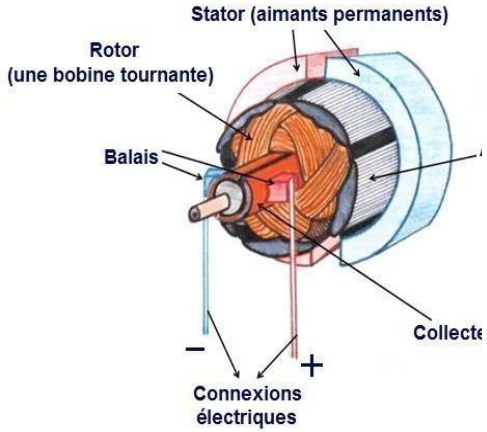

| | Monophasé | Triphasé |
|---|---|---|
| Avantages | Simple à calculer, 2-3 fils | Fortes puissances |
| Inconvénients | Faibles puissances par fil | 3-4 fils selon montage |
| Courant | Alternatif/continu | Alternatif seulement |
| Montage étoile | ---------- | Stabilité du système |
| Montage triangle | ---------- | Pmax Umax |

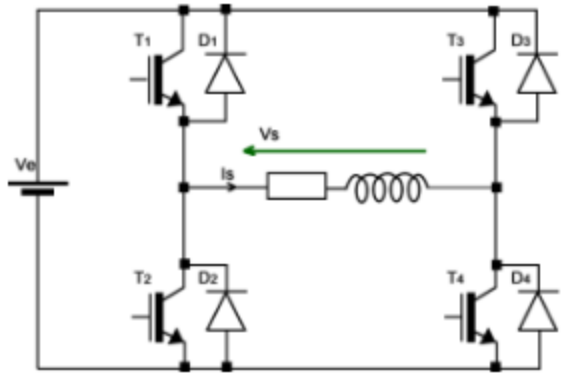
# I/Moteur à Courant Continu

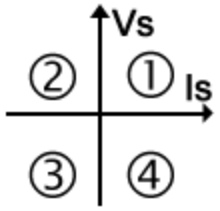
$C = K_t . I$

$V = K_c . \Omega$

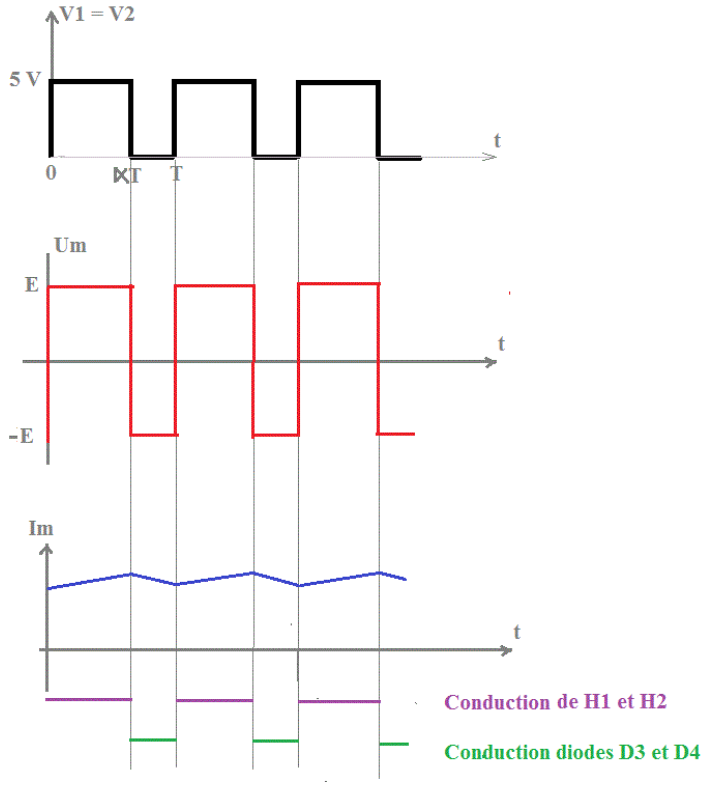
$C . \Omega = Pm$




```
#define brochePwmChoisie                5                  // On choisit d'émettre sur la broche D5
#define pourcentageRapportCycliqueChoisi  30               // avec un rapport cyclique égal à 30%

void setup() {

  // Conversion du « pourcentage » de rapport cyclique en une valeur comprise entre 0 et 255
  int rapportCycliqueEntre0et255 = map(pourcentageRapportCycliqueChoisi, 0, 100, 0, 255);

  // Génération du signal PWM
  pinMode(brochePwmChoisie, OUTPUT);                       // Définition de la broche D5 en tant que « SORTIE »
  analogWrite(brochePwmChoisie, rapportCycliqueEntre0et255);    // Génération du signal PWM, avec le rapport cyclique voulu

}

void loop() {

  // Pas de code ici, car tout se passe dans la partie setup !

}
```

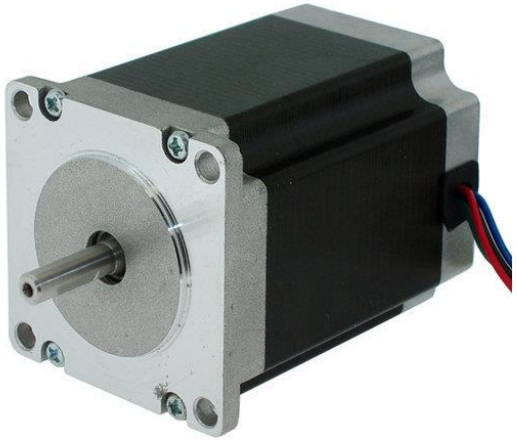


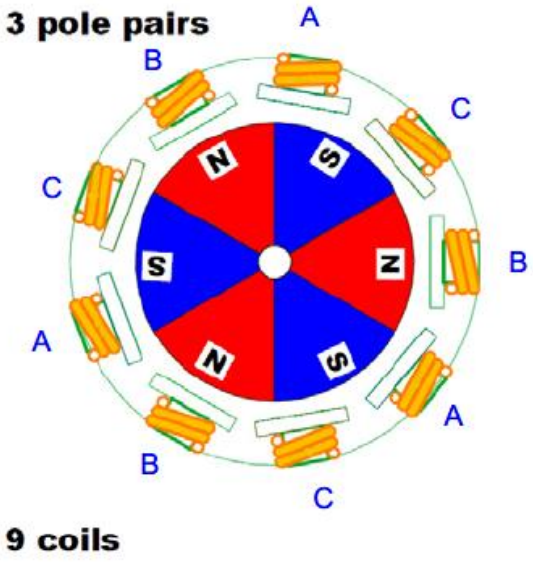
# II/Moteur Pas à Pas

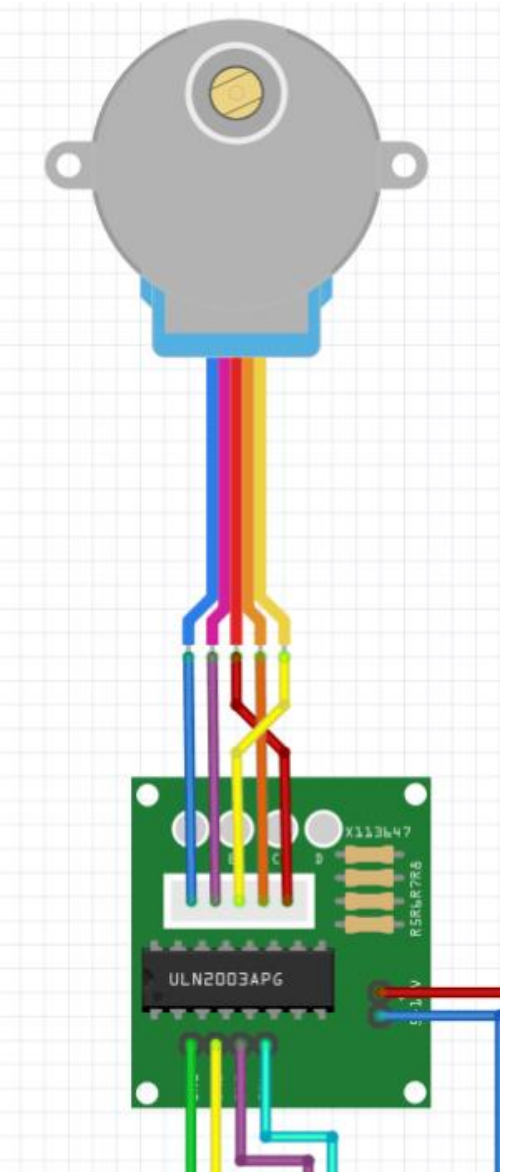
```
//we need this value t
int step_count;
void setup() {
  //seting up pins as
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);

}
```

```
void loop() {
  //using switch and based
  //as you can see we are s
  switch(step_count){
    case 0:
     digitalWrite(2, HIGH);
     digitalWrite(3, LOW);
     digitalWrite(4, LOW);
     digitalWrite(5, LOW);
    break;

    case 1:
     digitalWrite(2, LOW);
     digitalWrite(3, HIGH);
     digitalWrite(4, LOW);
     digitalWrite(5, LOW);
    break;
```

```
    case 2:
     digitalWrite(2, LOW);
     digitalWrite(3, LOW);
     digitalWrite(4, HIGH);
     digitalWrite(5, LOW);
    break;

    case 3:
     digitalWrite(2, LOW);
     digitalWrite(3, LOW);
     digitalWrite(4, LOW);
     digitalWrite(5, HIGH);
    break;
  }
  //incrementing step_count
  step_count++;
  //if it is bigger than sh
  if(step_count > 3){
    step_count = 0;
  }
```



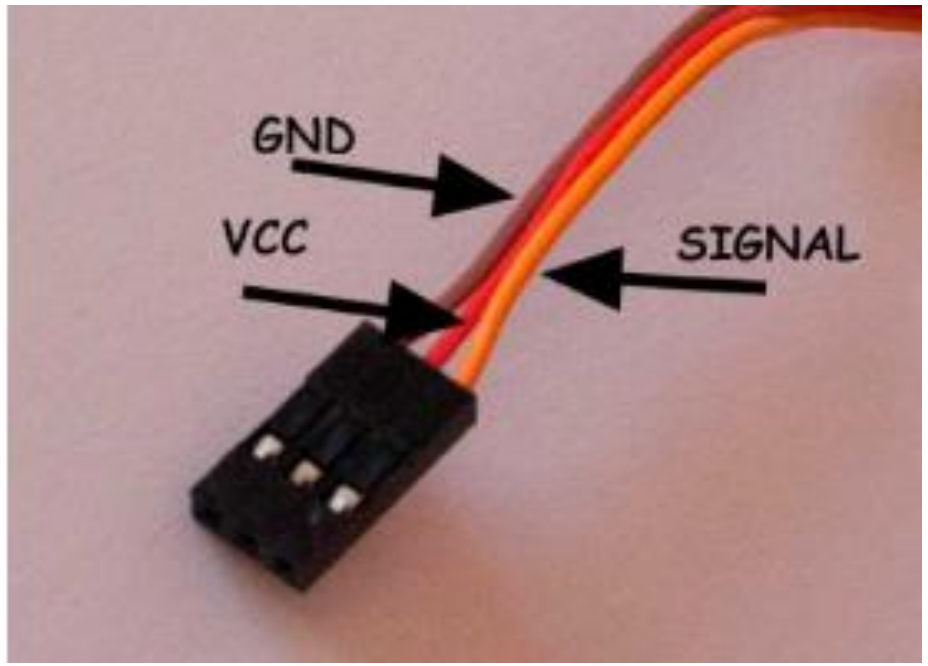


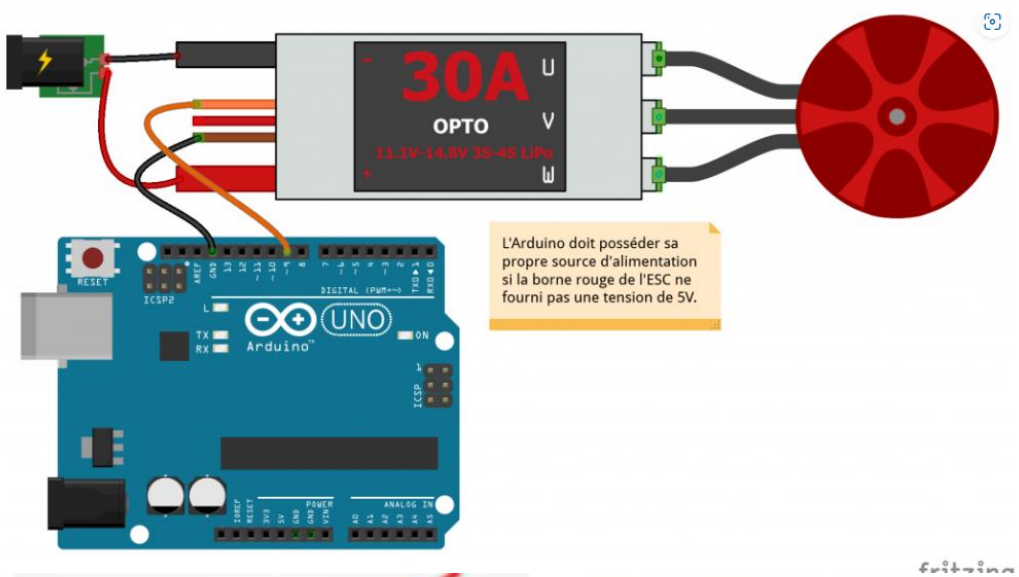
# III/ServoMoteur

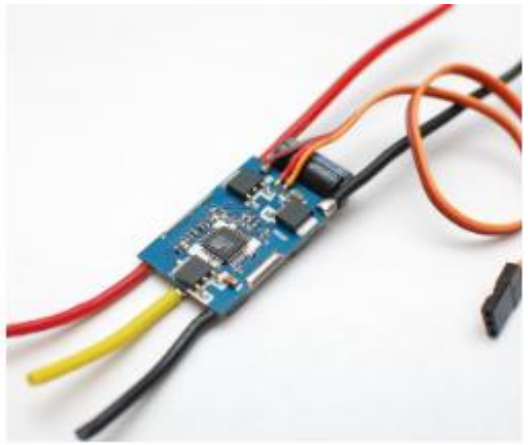

```
#include <Servo.h>
Servo monServo;

void setup() {
  monServo.attach(9);
}

void loop() {
  for (int i = 0; i <= 180; i++) {
    monServo.write(i);
    delay(15);
  }
  for (int i = 180; i >= 0; i--) {
    monServo.write(i);
    delay(15);
  }
}
```

-Limite d'angle de 0 à 180°
-Ne nécessite pas de carte de commande en plus du microcontrôleur



# IV/Brushless



```
#include <Servo.h>

Servo esc;   // Création de l'objet permettant le contrôle de l'ESC

int val = 0; //

void setup() {
   esc.attach(9); // On attache l'ESC au port numérique 9 (port PWM obligatoire)
   delay(15);
   Serial.begin(9600);

   // Initialisation de l'ESC
   //  (certains ESC ont besoin d'une "procédure d'initialisation"
   //   pour devenir opérationnels - voir notice)
   esc.write(0);
   delay(1000);
   esc.write(180);
   delay(1000);
   esc.write(0);

   // Quelques informations pour l'utilisateur
   Serial.println("Saisir un nombre entre 0 et 179");
   Serial.println("(0 = arret - 179 = vitesse maxi");
   Serial.println(" demarrage a partir de 20)");
   }

void loop() {
   if (Serial.available() > 0) {
      val = Serial.parseInt();   // lecture de la valeur passée par le port série
      Serial.println(val);
      esc.write(val);            //
      delay(15);
      }
   }
```

# Types de Moteurs utilisés

| | Moteur à Courant continu | Moteur Pas à Pas | Servomoteur | Brushless |
|---|---|---|---|---|
| Utilisation | Applications contrôlées en vitesse (ex: embase de PAMI) | Applications contrôlées en vitesse et position | Contrôle en position d'un actionneur en moins de 5V | Applications en Vitesse de Rotation |
| Commande | Courant continu / hacheur | Pont en H , double PWM, sens | Pont en H, double PWM, sens | ESC |
| | | | | |
| Avantages | Polyvalent, couple et vitesse proportionnels à Ic et Vc | Polyvalent, Précis, couple élevé | Précis | Très grande vitesse |
| Inconvénients | Commande en position impossible sans asservissement | Asservissement complexe, demande deux à quatre câbles de commande | Souvent limité à 0:180° de battement, Fragile | Couple très faible |