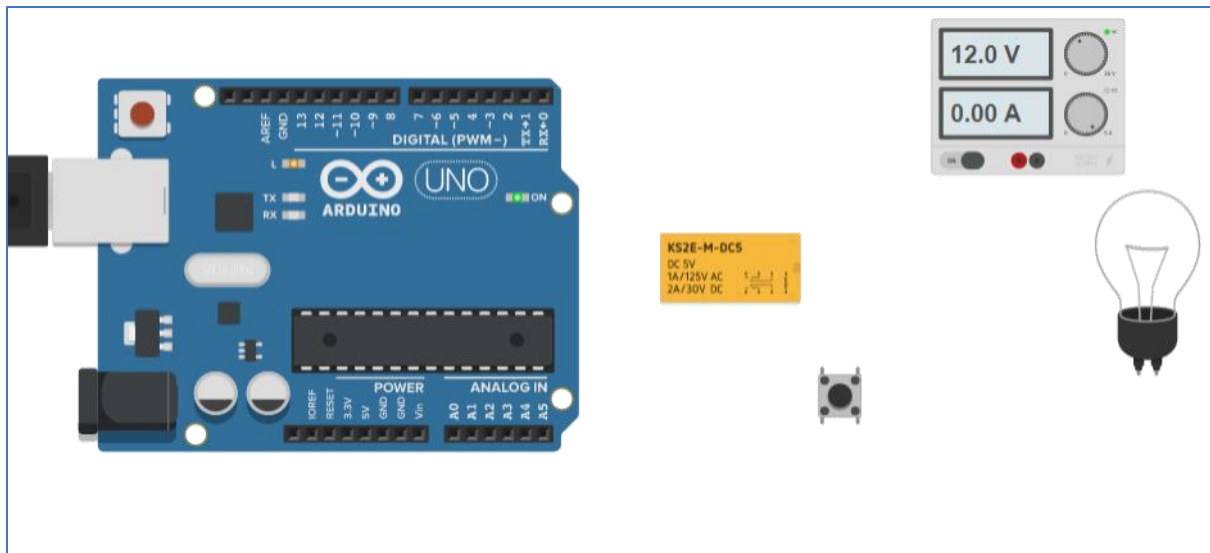
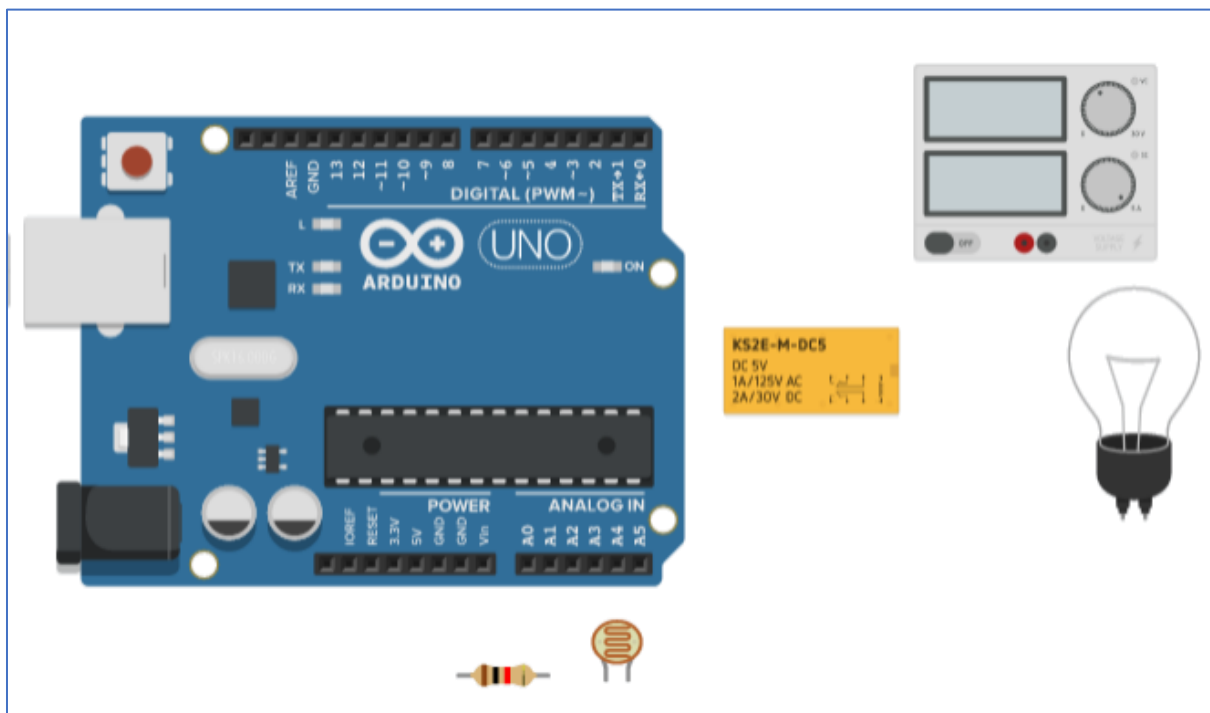


1. Relais électromécanique

Q1) Compléter le schéma électrique (respecter les couleurs de fils : rouge=alimentation, noir=masse, vert=information) pour que l'ampoule s'allume lorsque l'on appuie sur le bouton poussoir.



Q2) Compléter le schéma électrique (respecter les couleurs de fils : rouge=alimentation, noir=masse, vert=information) pour que l'ampoule s'allume lorsque la luminosité (acquisition sur broche A0) dépasse le seuil de 500. Le pilotage du relais se fera à partir de la broche 3.



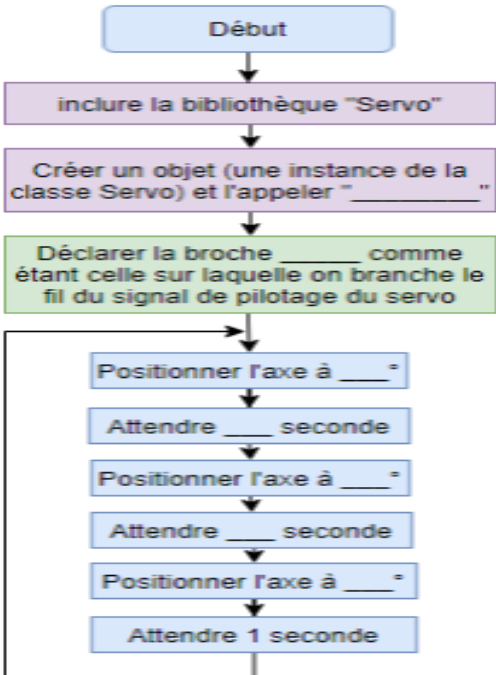
Q3) Compléter le programme ci-dessous pour que, lorsque la valeur lue sur la broche A0 dépasse le seuil de 500, la lampe s'allume.

```
void setup() {
  Serial.begin(9600);
  pinMode(_____, INPUT);
  pinMode(_____, OUTPUT);
}

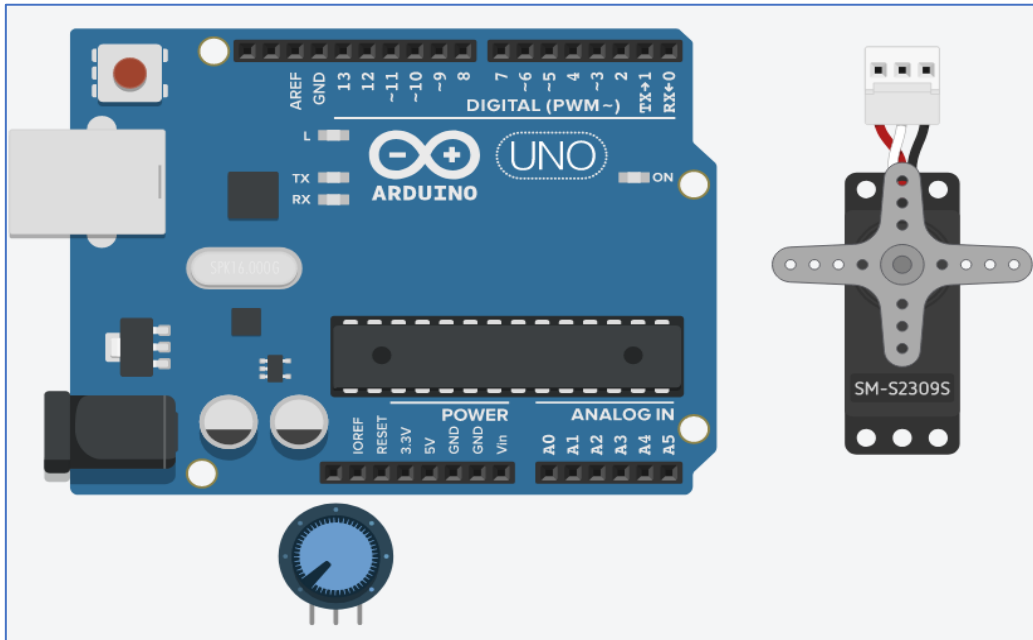
void loop() {
  Serial.println(analogRead(_____));
  if (analogRead(_____) > _____){
    digitalWrite(_____, HIGH);
  }
  else {
    digitalWrite(_____, LOW);
  }
  delay(500);
}
```

2. Servomoteur

Q4) Compléter l'algorithme et le programme pour que l'axe du servomoteur se place à 10°, puis à 20°, puis à 30° avec une attente de 2 secondes entre chaque position. Ce cycle sera exécuté en boucle.

 <pre> graph TD Start([Début]) --> Include[inclure la bibliothèque "Servo"] Include --> Create[Créer un objet (une instance de la classe Servo) et l'appeler "_____"] Create --> Declare[Déclarer la broche _____ comme étant celle sur laquelle on branche le fil du signal de pilotage du servo] Declare --> Loop subgraph Loop [Boucle] direction TB L1[Positionner l'axe à _____°] --> L2[Attendre _____ seconde] L2 --> L3[Positionner l'axe à _____°] L3 --> L4[Attendre _____ seconde] L4 --> L5[Positionner l'axe à _____°] L5 --> L6[Attendre 1 seconde] L6 --> L1 end Loop --> End([Fin]) </pre>	<pre> #include <Servo.h> Servo _____; void setup() { monServo.attach(_____); } void loop() { monServo.write(_____); delay(_____); monServo.write(_____); delay(_____); monServo.write(_____); delay(_____); } </pre>
--	--

Q5) Compléter le schéma électrique (respecter les couleurs de fils : rouge=alimentation, noir=masse, vert=information) pour que le potentiomètre renvoie une information sur la broche A0 et que le servomoteur soit piloté par la broche 3.



Q6) Compléter le programme ci-dessous pour que le servomoteur (piloté par la broche 3) pivote en fonction de l'angle du potentiomètre (broche A0).

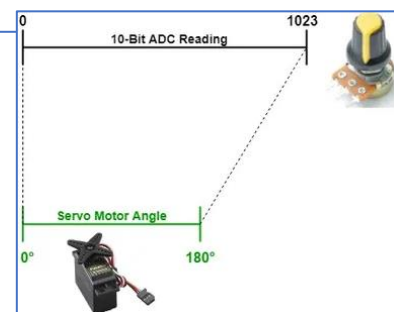
Syntaxe de la fonction map :

map (valeur, limite_basse_source, limite_haute_source, limite_basse_destination, limite_haute_destination)

```

1  #include <Servo.h>
2  Servo _____;
3  int angle=10;
4
5  void setup()
6  {
7    monServo.attach(_____);
8    pinMode(A0, INPUT);
9  }
10
11 void loop()
12 {
13   angle=map(analogRead(A0), 0, _____, 0, _____);
14   monServo.write(_____);
15   delay(500);
16 }

```



3. Moteur à courant continu

Q7) Entourer la définition des fonctions en vert et l'appel des fonctions en bleu. Compléter l'algorithme

```

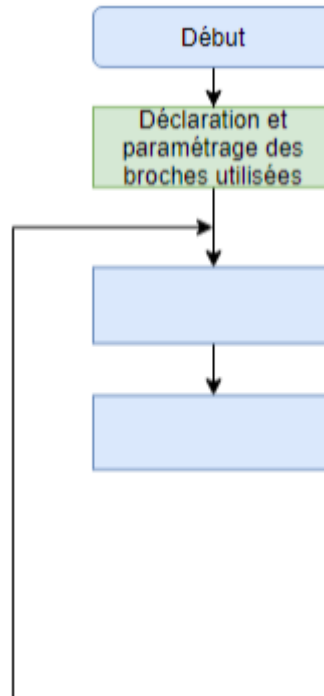
void setup(){
  pinMode(12, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(3, OUTPUT);
}
void loop(){
  Forward_speed1() ;
  delay(7000) ;
  Backward_speed2() ;
  delay(5000) ;
  Brake() ;
  delay(3000) ;
}

void Forward_speed1() {
  digitalWrite(12, HIGH);
  digitalWrite(9, LOW);
  analogWrite(3, 200);
}

void Backward_speed2() {
  digitalWrite(12, LOW);
  digitalWrite(9, LOW);
  analogWrite(3, 100);
}

void Brake() {
  digitalWrite(9, HIGH);
}

```



Q8) A quel pourcentage de sa vitesse maximale le moteur tourne-t-il en marche avant ? en marche arrière ?

Q9) Compléter le programme pour que la vitesse du moteur soit proportionnelle à la rotation du potentiomètre.

```

int vitesse;

void setup(){
  pinMode(12, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(A0, INPUT);
}

void loop(){
  vitesse=map(analogRead(_____),0,_____,0,_____);
  digitalWrite(12,_____);
  digitalWrite(9,_____);
  analogWrite(3,_____);
  delay(50);
}

```

