



CELLULE ELECTRIQUE

ARDUINO : SÉANCE 2

Presentée par : Ayoub cheggari
Ayoub.cheggari@gmail.com



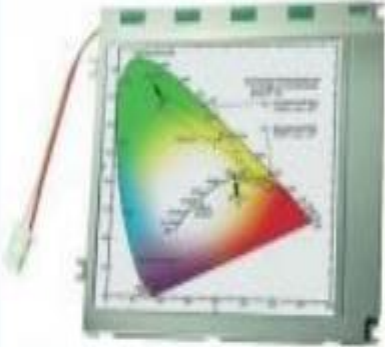
Plan :

- *L'afficheur LCD.*
- *Les sorties analogiques PWM.*
- *Les servos moteurs .*
- *Les moteurs DC.*
- *Le shield driver motor(adafruit).*
- *Le module bluetooth.*
- *Le capteur ultrason.*

L'afficheur LCD:



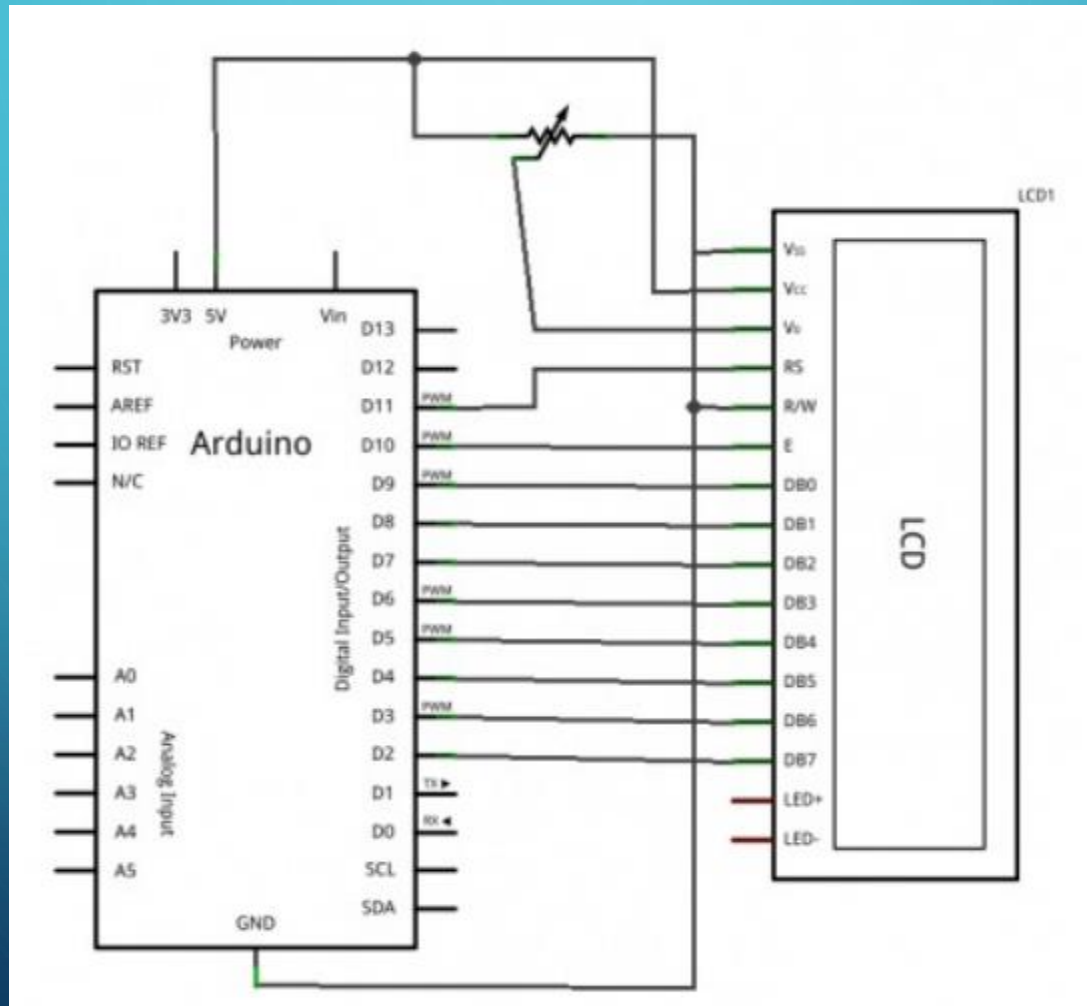
L'afficheur LCD(types):

Afficheur alphanumérique	Afficheur graphique (monochrome)	Afficheur graphique (couleur)
 <p>TOPWAY Model: LMBT62HBC</p>		

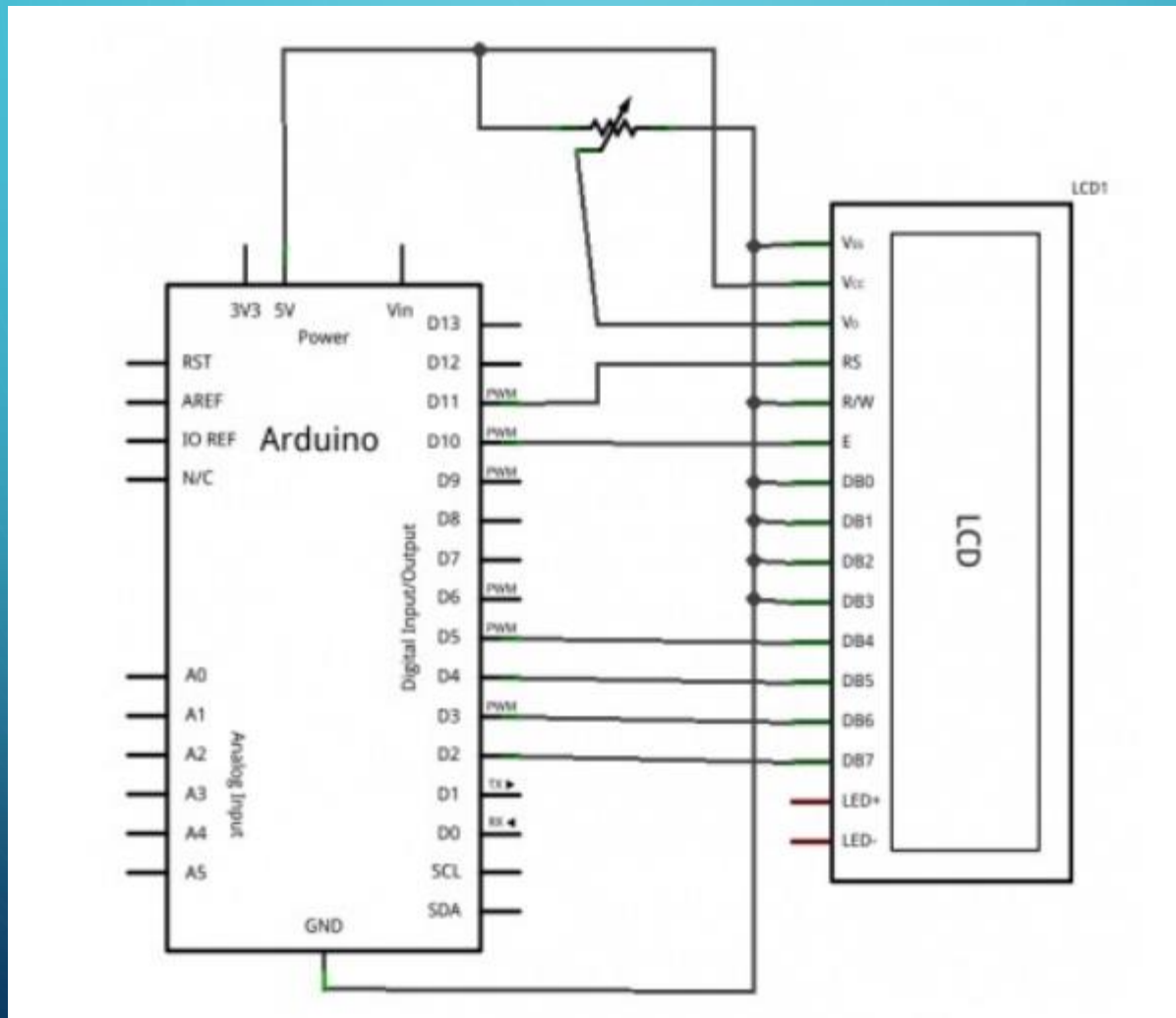
L'afficheur LCD(types de communication):

- *La communication parallèle.*
- *La communication série.*
- *La liaison I2C.*

L'afficheur LCD(brochage parallèle):



L'afficheur LCD(brochage semi_para):

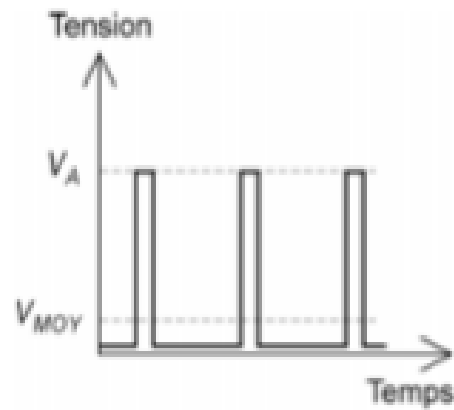


The background is a solid blue gradient. It is decorated with white circuit-like lines and nodes. These lines are primarily located along the left and right edges, with some extending towards the center. The nodes are small white circles at the intersections of the lines.

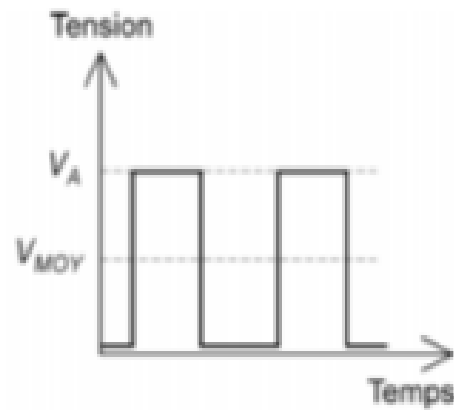
L'afficheur LCD(les fonctions):

<https://www.arduino.cc/en/Reference/LiquidCrystal>

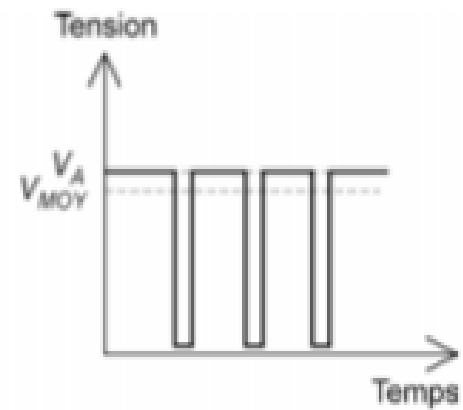
Les sorties analogiques PWM:



Rapport cyclique
proche de 0 %

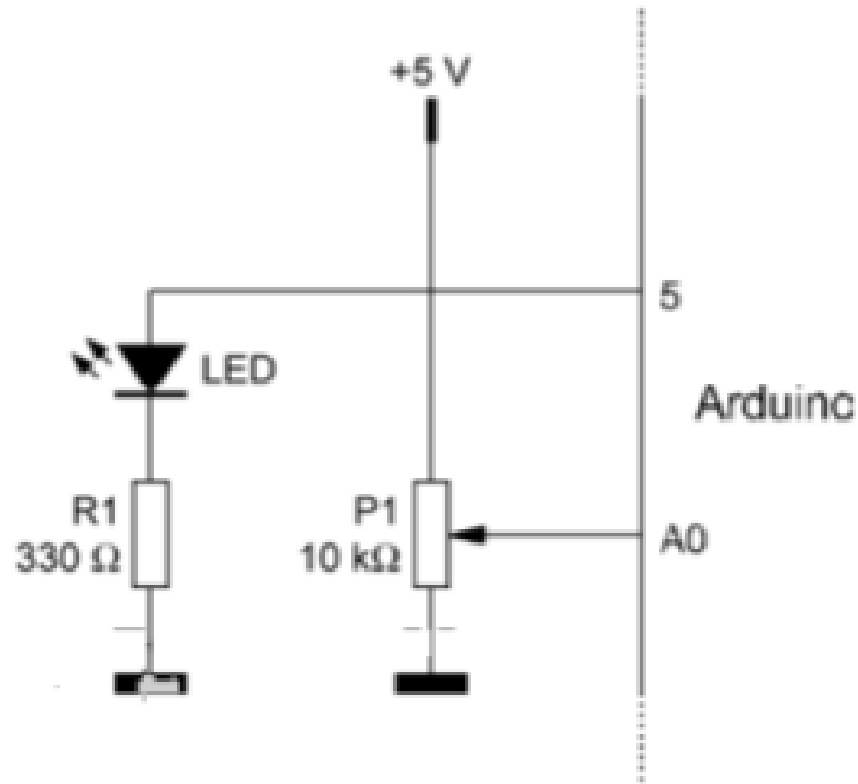


Rapport cyclique
de 50 %



Rapport cyclique
proche de 100 %

Les sorties analogiques PWM(exemple):



Les sorties analogiques PWM(exemple):

```
#define potentiometre 0
#define LED 5

void setup()
{
    pinMode(LED, OUTPUT);
}

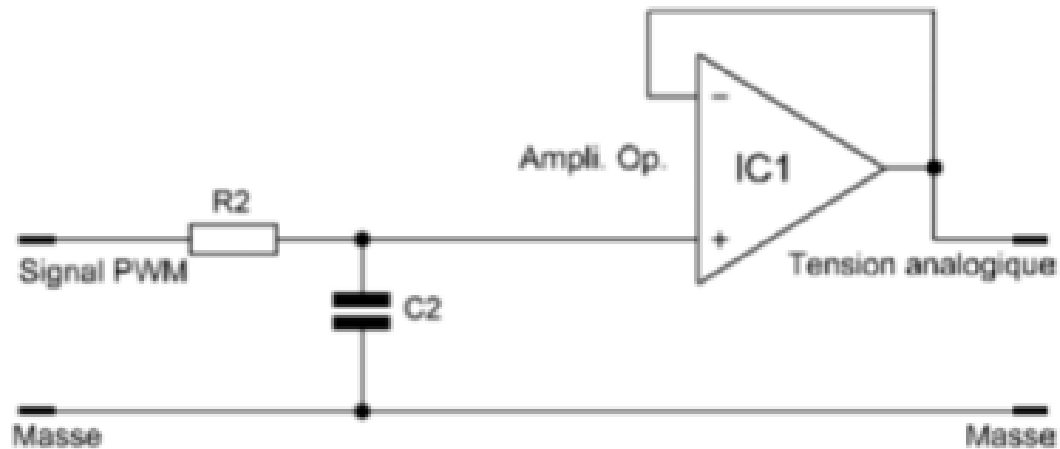
void loop()
{
    int luminosite = analogRead(potentiometre);
    // Lecture du potentiomètre
    luminosite = map(luminosite, 0, 1023, 0, 255);
    // Conversion dans la plage 0 - 255
    analogWrite(LED, luminosite);
    // Définition du signal PWM
}
```

The background is a solid blue gradient. In the corners, there are decorative white line art elements resembling electronic circuit traces or a stylized city skyline. These elements consist of vertical and horizontal lines of varying lengths, some ending in small circles, creating a geometric, circuit-like pattern.

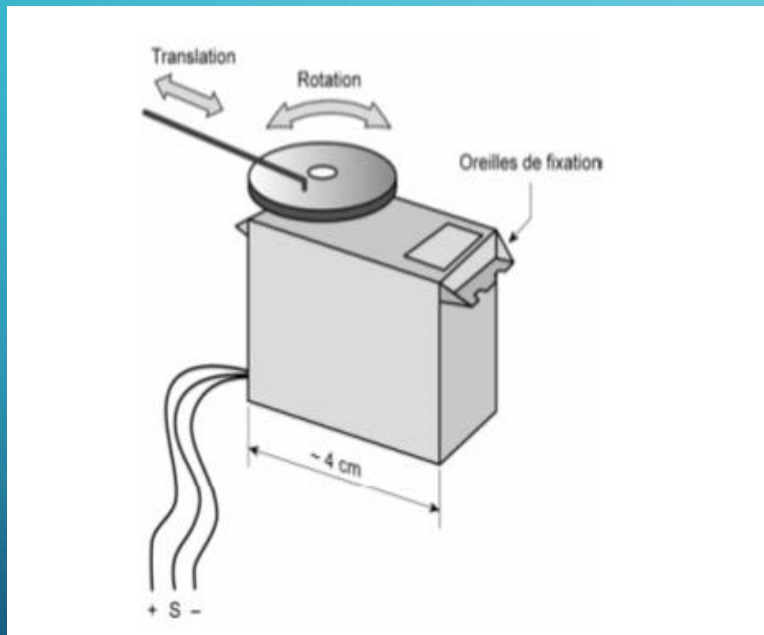
*Comment obtenir une
tension continue ?*

Les sorties analogiques PWM:

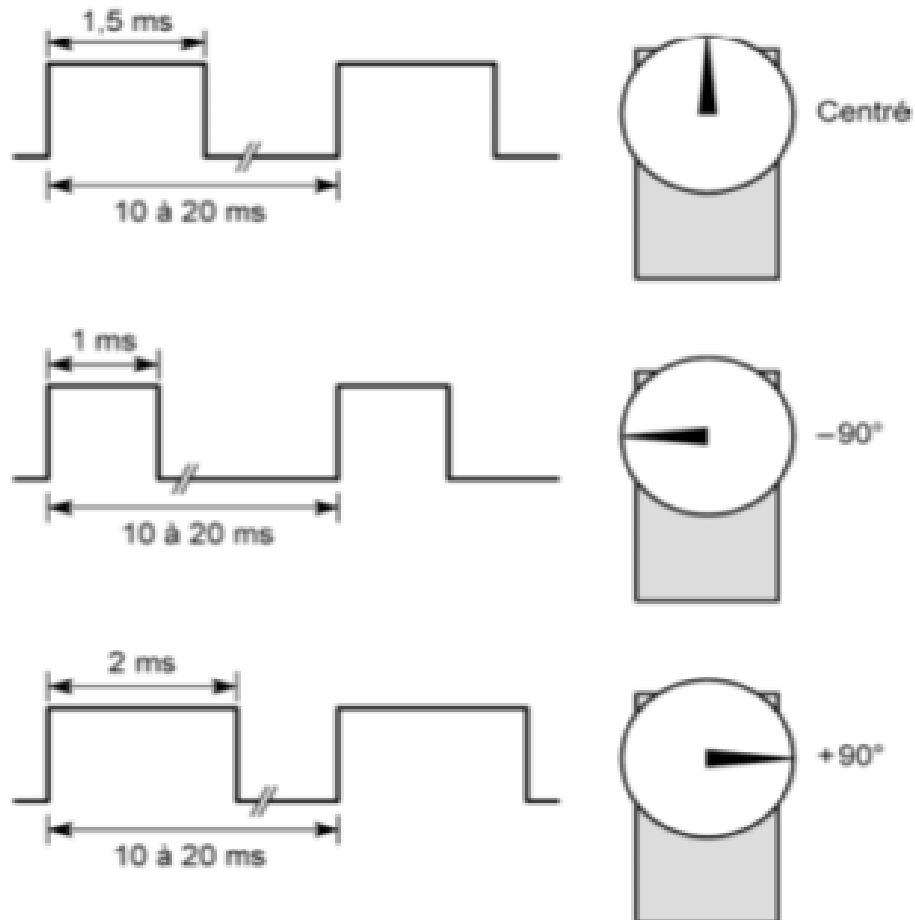
Filtre passe_bas:



Les servomoteurs:



Les servomoteurs(principe):

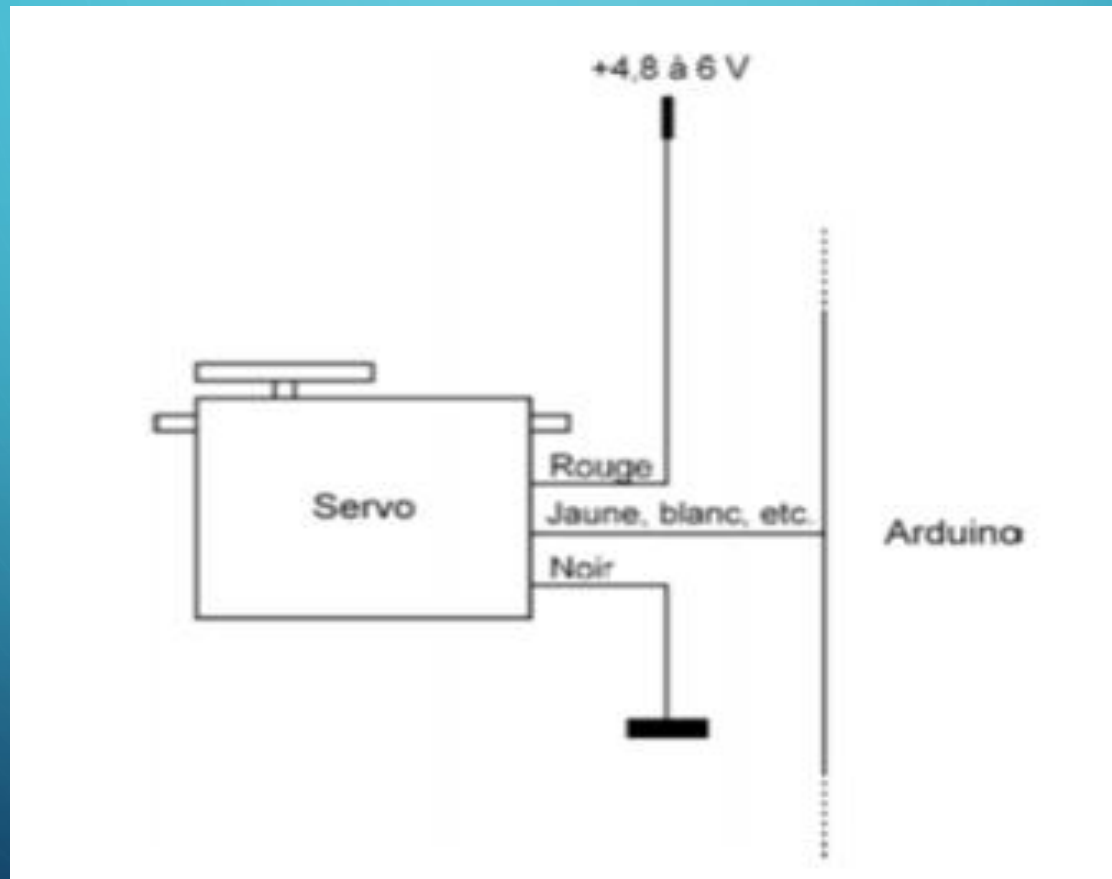


Les servomoteurs(principe):

Règles:

- *La masse doit être liée à celle de l'arduino.*
- *La tension ne doit pas être prélevée de +5v de l'arduino.*

Les servomoteurs(principe):





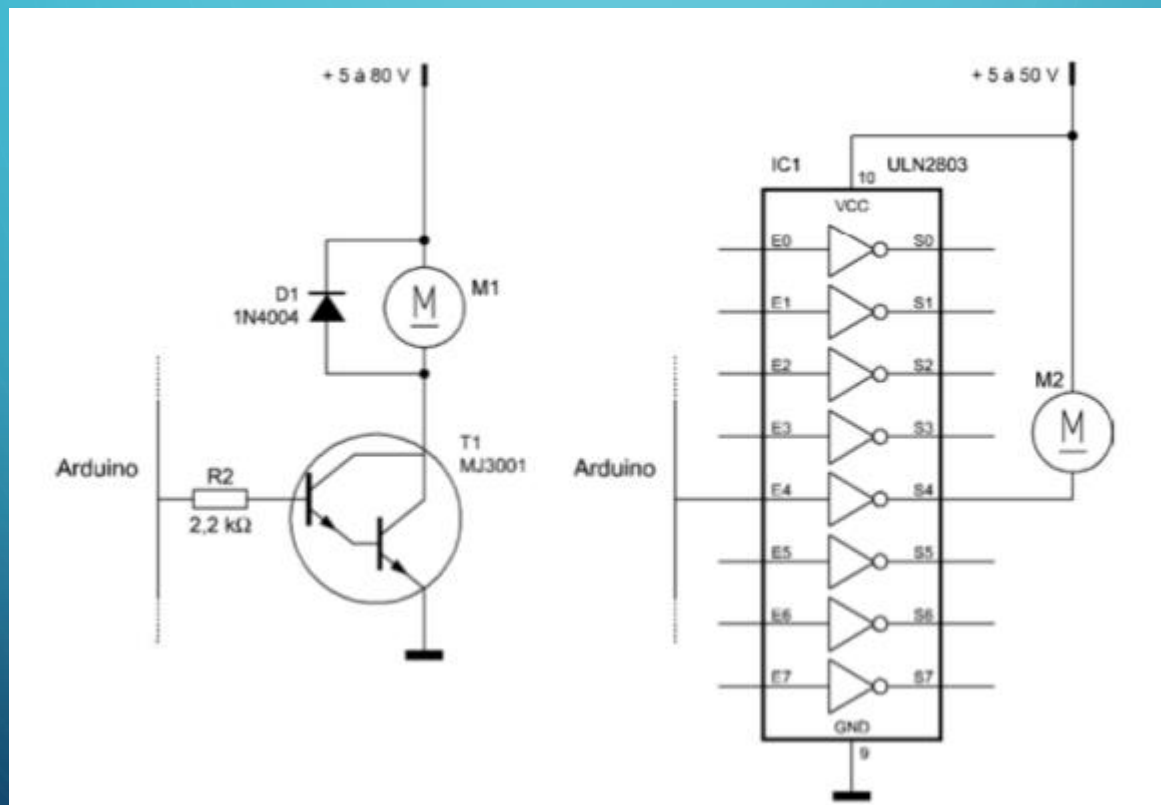
Les servomoteurs(fonctions):

<https://www.arduino.cc/en/Reference/Servo>

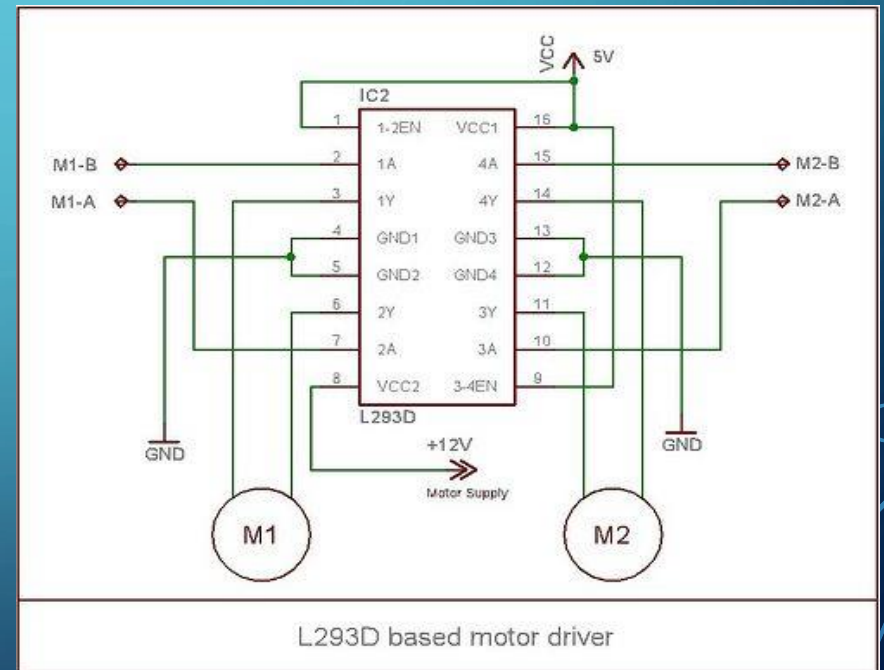
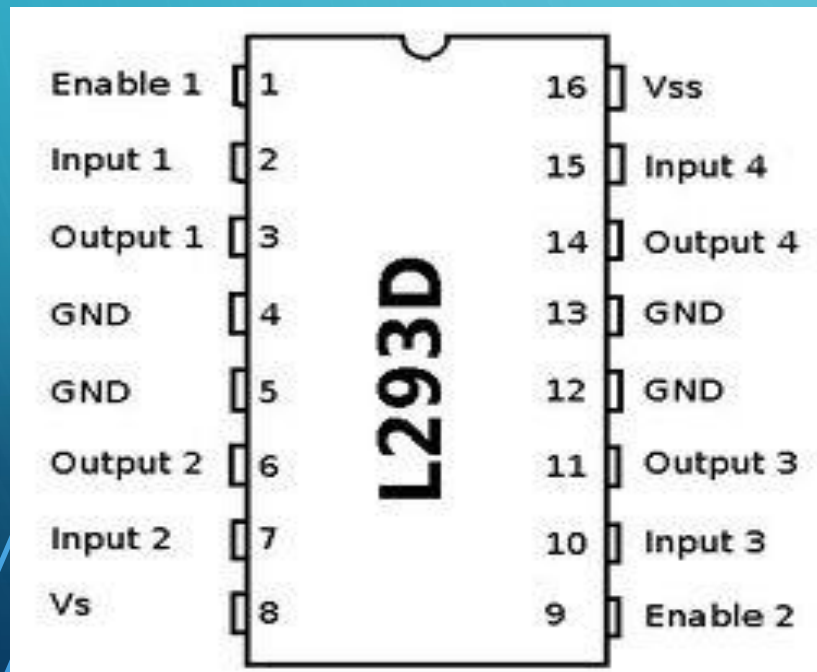


Les moteurs à courant continu:

Alimentation à un sens:



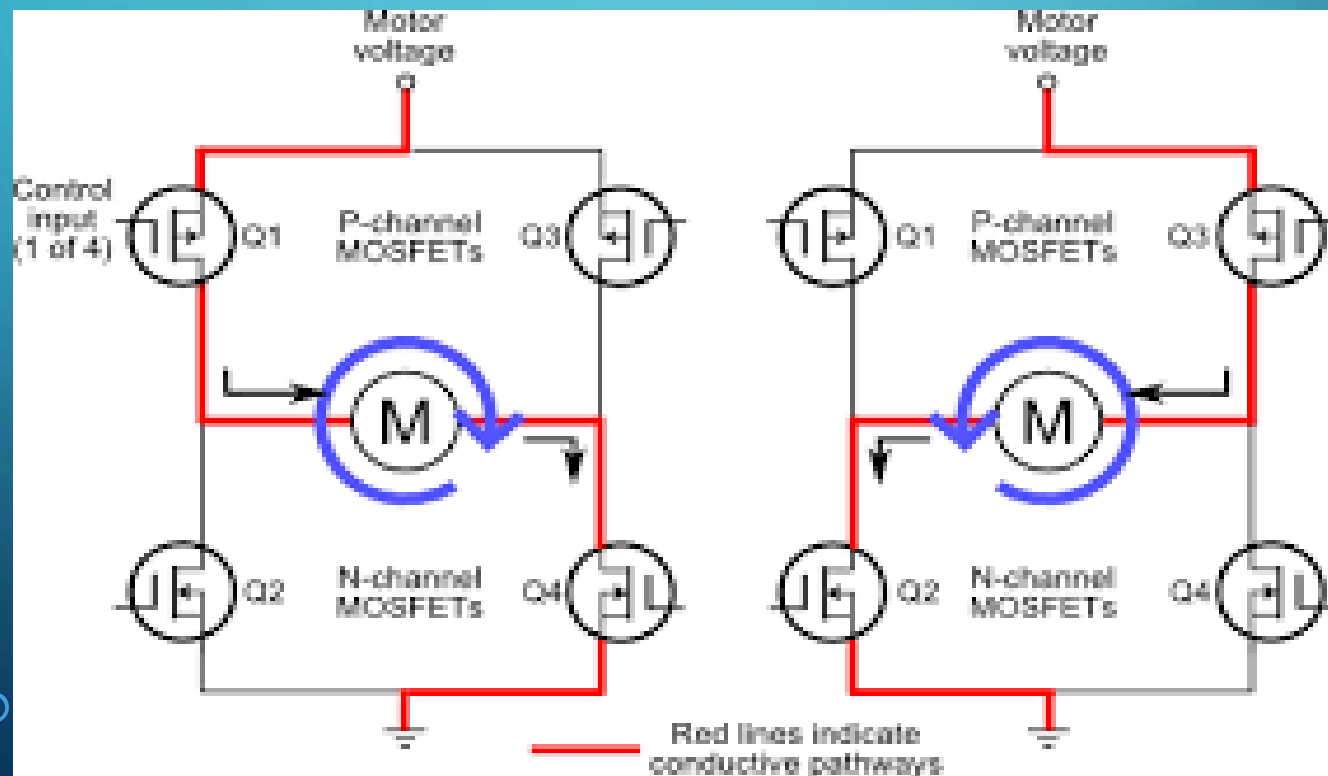
Les moteurs à courant continu(rotation dans 2 sens):



Les moteurs à courant continu(rotation dans 2 sens):

Enable 1	Input 1	Input 2	Fonction
High	Low	High	Tourne dans le sens horlogique
High	High	Low	tourne dans le sens anti-horlogique
High	Low	Low	Stop
High	High	High	Stop
Low	Non applicable	Non applicable	Stop

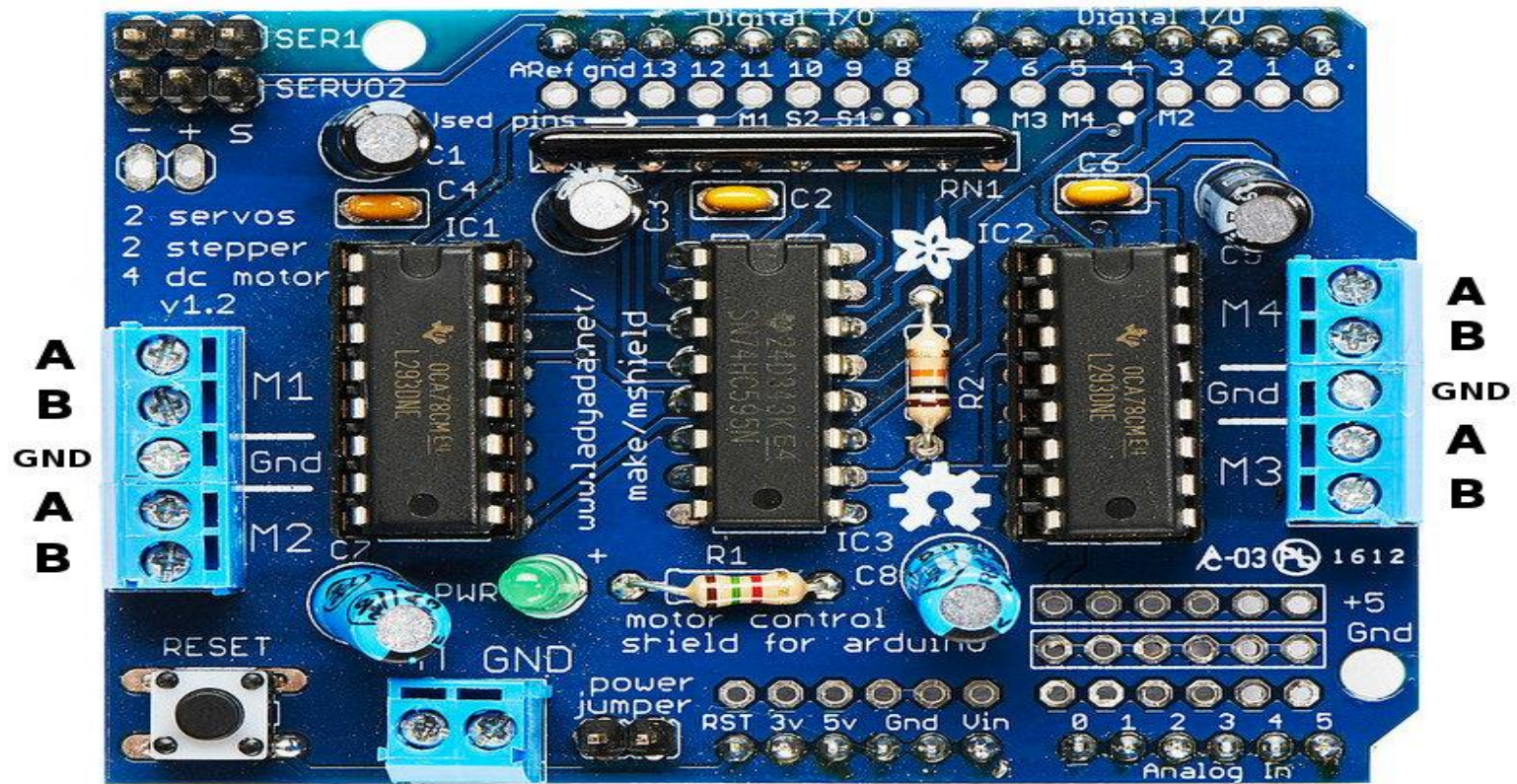
Les moteurs à courant continu(rotation dans 2 sens):



The background is a blue gradient with faint concentric circles. In the corners, there are white line-art illustrations of electronic circuit boards, featuring various lines, nodes, and circular components.

Les shields

Driver motor shield:



Driver motor shield:

- How many motors can I use with this shield?
You can use 2 DC hobby servos that run on 5V and up to 4 DC motors or 2 stepper motors (or 1 stepper and up to 2 DC motors) that run on 5-12VDC.
- What Arduinos is this shield compatible with?
It is tested to work with Duemilanove, Diecimila, Uno (all revisions), Leonardo and Mega/ADK R3 and higher.
- I'm trying to build this robot and it doesn't seem to run on a 9V battery....
You cannot power motors from a 9V battery. You must use AA batteries or a lead acid battery for motors.

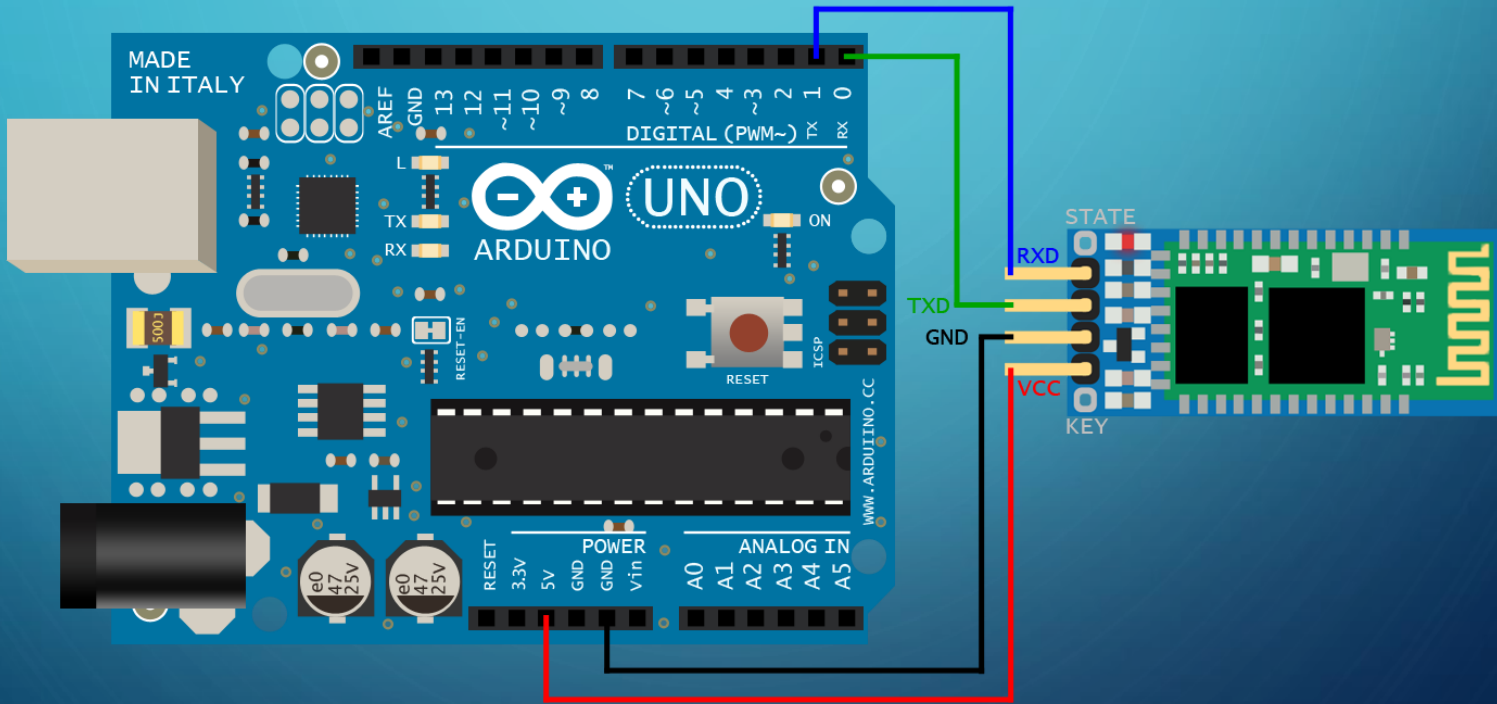
Driver motor shield(fonctions):

- `#include <AFMotor.h>`
- `AF_DCMotor motor(4, MOTOR34_64KHZ);`
- `motor.setSpeed(0);`
- `motor.run(FORWARD);`
- `motor.run(BACKWARD);`
- `motor.run(RELEASE);`

Module bluetooth:



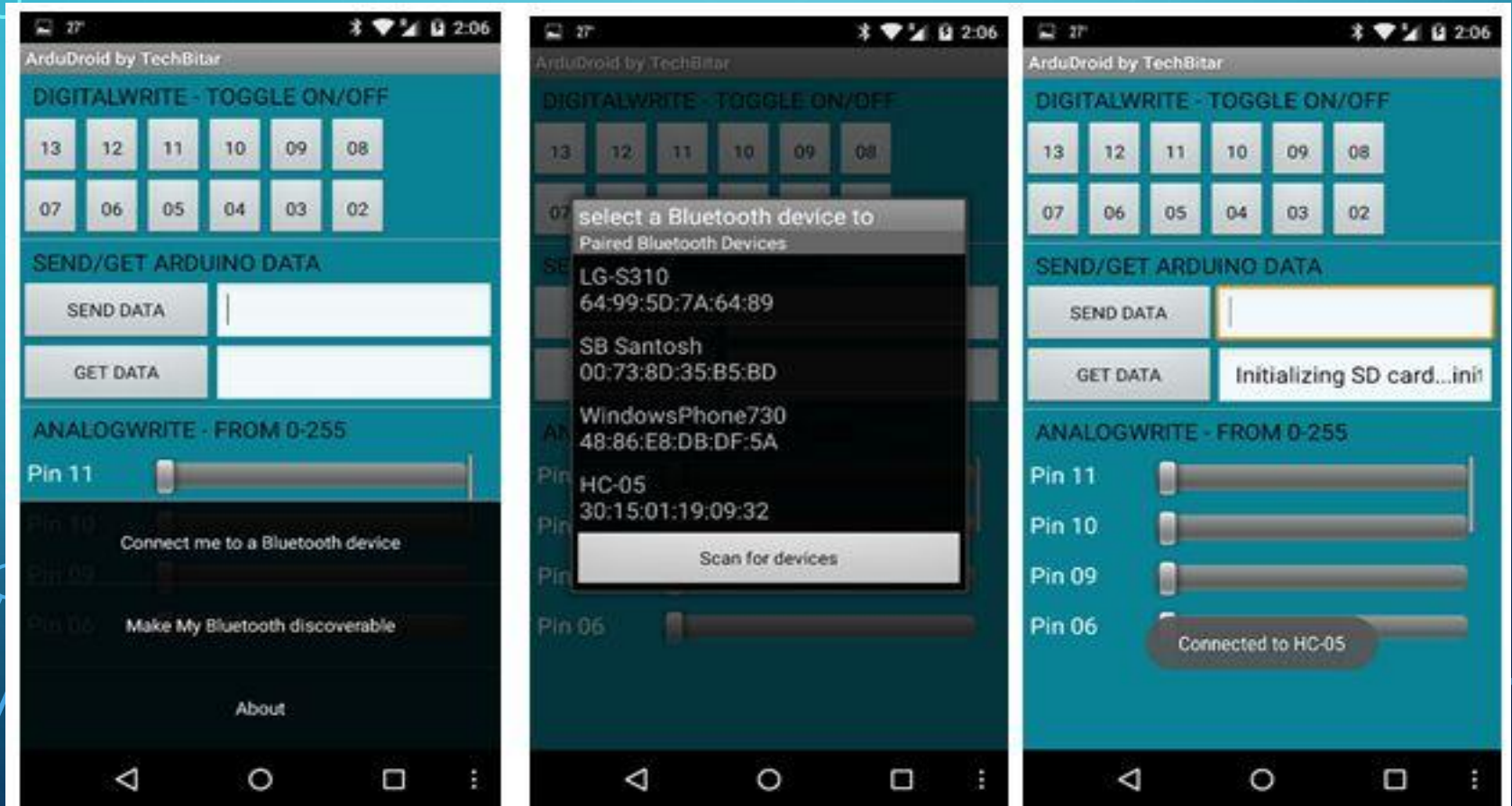
Module Bluetooth(application):



Module Bluetooth(application):

```
hc06_led
1  #include <SoftwareSerial.h>
2
3  void setup()
4  {
5      Serial.begin(9600);
6      pinMode(11, OUTPUT);
7  }
8
9  void loop()
10 {
11     if (Serial.available() > 0) {
12         char ch = Serial.read();
13         Serial.print("Received: ");
14         Serial.println(ch);
15         if (ch == 'a') {
16             digitalWrite(11, HIGH);
17         }
18         else if(ch=='b'){
19             digitalWrite(11, LOW);
20         }
21     }
22 }
```

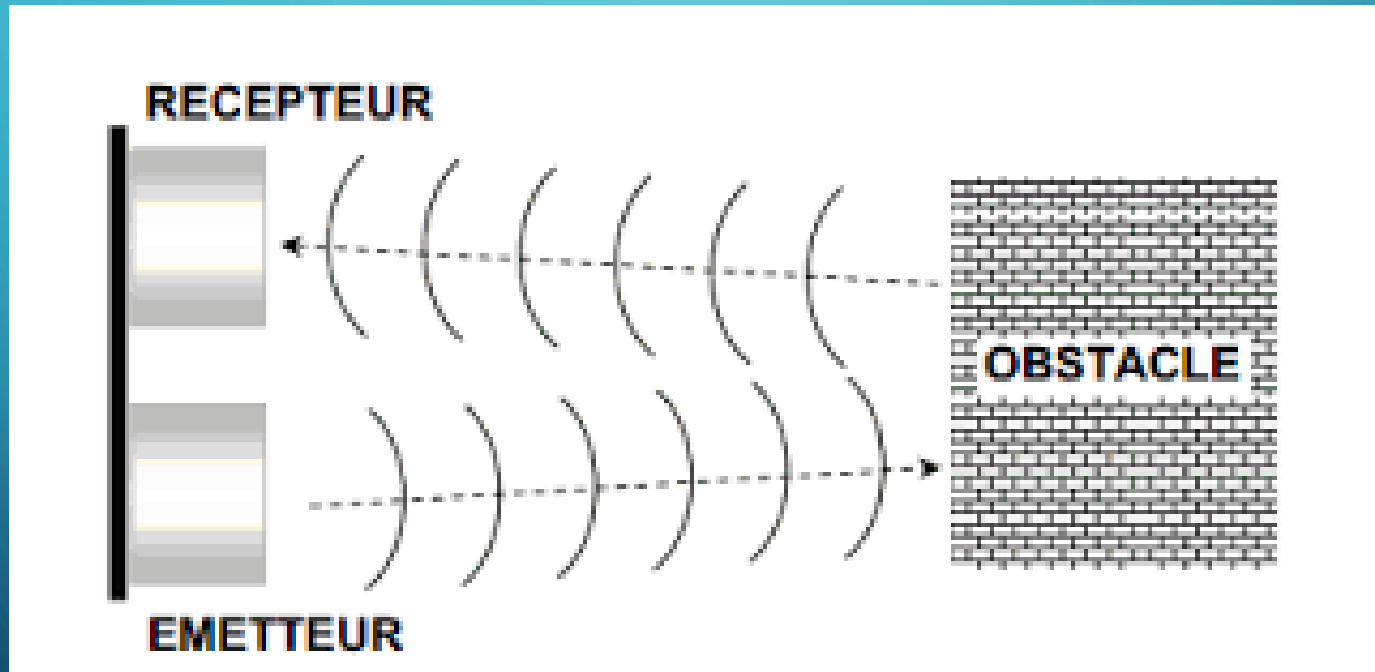
Module Bluetooth(application):



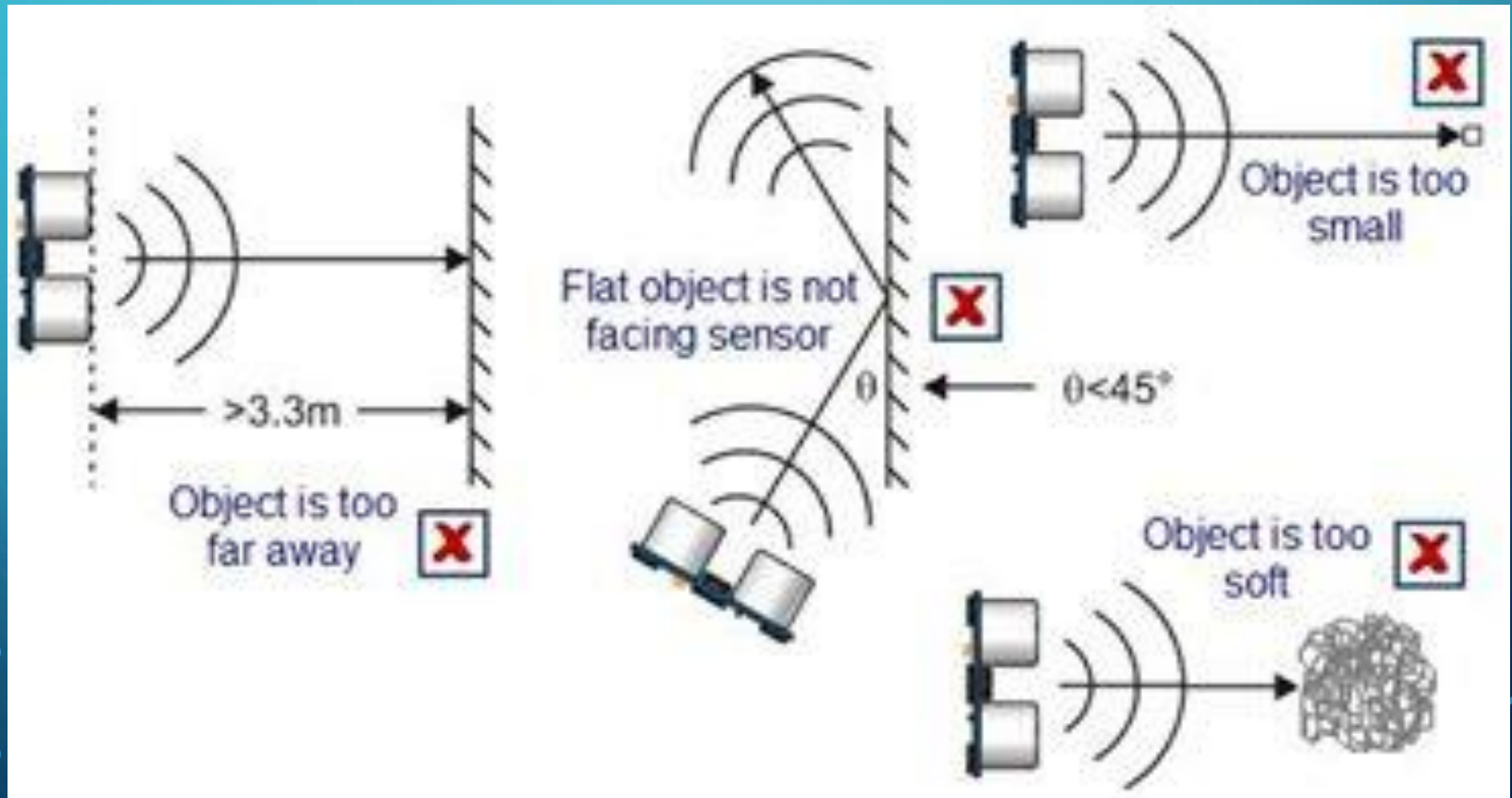
Le capteur ultrasonique:



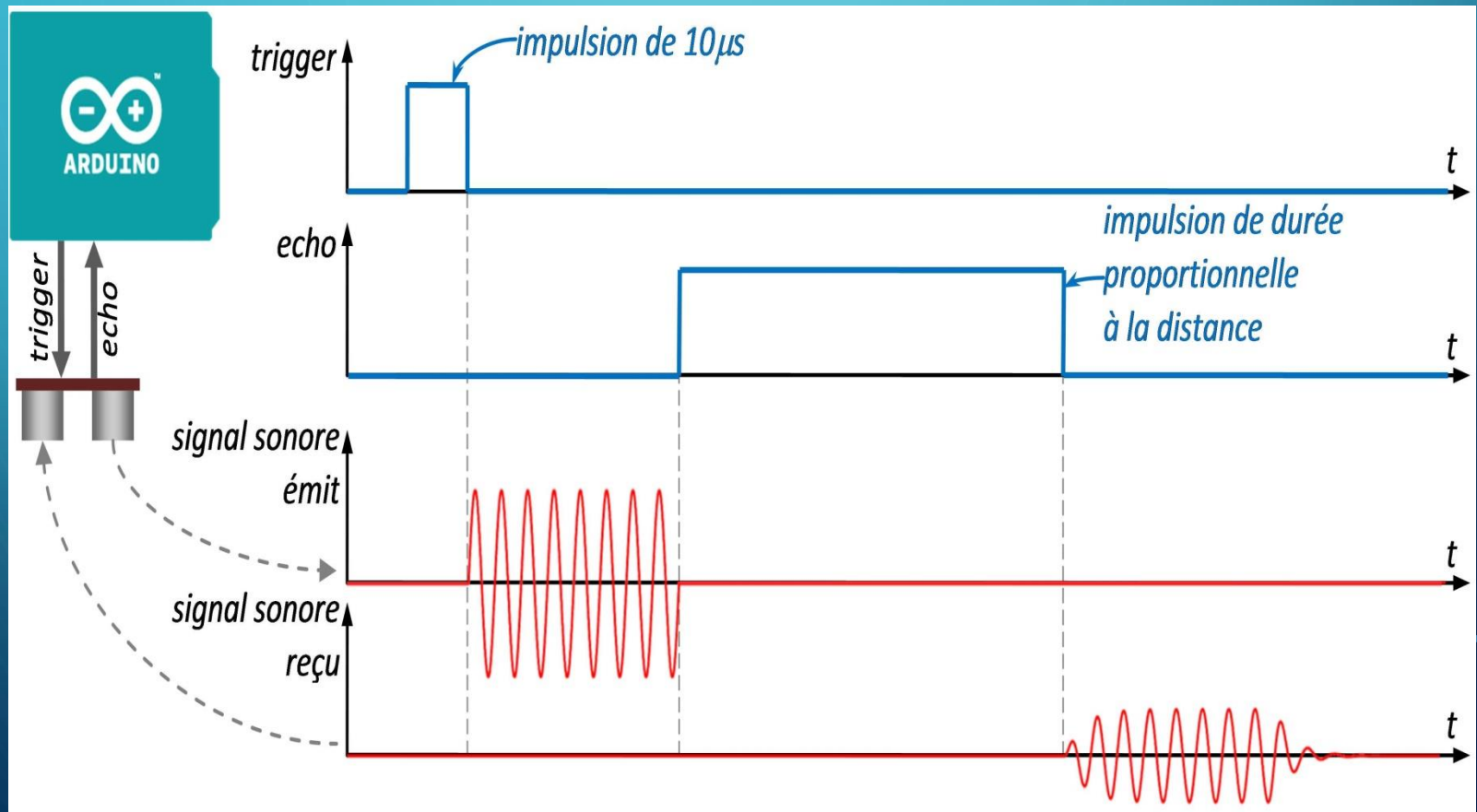
Le capteur ultrasonique(fonctionnement):



Le capteur ultrasonique(fonctionnement):



Le capteur ultrasonique(fonctionnement):



Le capteur ultrasonique(code):

```
/* Utilisation du capteur Ultrason HC-SR04 */  
  
// définition des broches utilisées  
int trig = 12;  
int echo = 11;  
long lecture_echo;  
long cm;  
  
void setup()  
{  
  pinMode(trig, OUTPUT);  
  digitalWrite(trig, LOW);  
  pinMode(echo, INPUT);  
  Serial.begin(9600);  
}  
  
void loop()  
{  
  digitalWrite(trig, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trig, LOW);  
  lecture_echo = pulseIn(echo, HIGH);  
  cm = lecture_echo / 58;  
  Serial.print("Distancem : ");  
  Serial.println(cm);  
  delay(1000);  
}
```

Le capteur ultrasonique(code):

$$V(m/s) = D(m)/T(s)$$

$$D(m) = V(m/s) \times T(s)$$

$$D(m) = 340(m/s) \times T(s)$$

$$D(m) = 340(m/s) \times TAR(s) / 2$$

$$D(cm) = 0.0340(cm/s) \times TAR(s) / 2$$

$$D(cm) = TAR(s) / 58.8(s/cm)$$

*Merci pour votre
attention!!!*

Des questions ???