

Table des matières

Introduction	3
1. Module joystick analogique avec bouton-poussoir	4
2. Module à relais	6
3 et 4. Modules à micro	7
5. Module suiveur de ligne	9
6. Module détecteur d'obstacle	10
7. Module détecteur de flamme	11
8. Module à effet Hall.....	13
9. Module tactile	14
10. Capteur de température digital.....	15
11. Module à buzzer actif.....	17
12. Module à buzzer passif.....	18
13. Module à LED RGB 5 mm.....	19
14. Module à Led RGB CMS.....	22
15 et 16. Modules à Led bicolore 5 mm et 3 mm.....	25
17. Module à interrupteur ILS	27
18. Module miniature à ILS	28
19. Capteur de pulsations cardiaques	29
20. Module à Led 7 couleurs	30
21. Module laser.....	31
22. Module à bouton-poussoir	32
23. Module détecteur de chocs.....	33
24. Module à encodeur rotatif	34
25. Module capteur d'inclinaison avec Led rouge (x2).....	36
26. Module capteur d'inclinaison.....	37
27. Module capteur de vibrations	38
28. Module capteur de luminosité (LDR).....	39
29. Module capteur de température et d'humidité (DHT11).....	40
30. Capteur à effet Hall à sortie analogique	42
31. Capteur à effet Hall à sortie numérique	43
32. Capteur de température DS18B20	44
33. Capteur de température analogique	46
34. Module à émetteur IR	47
35. Module à récepteur IR.....	48
36. Module à capteur fourche.....	50
37. Module détecteur de chocs et de vibrations.....	51

Introduction.

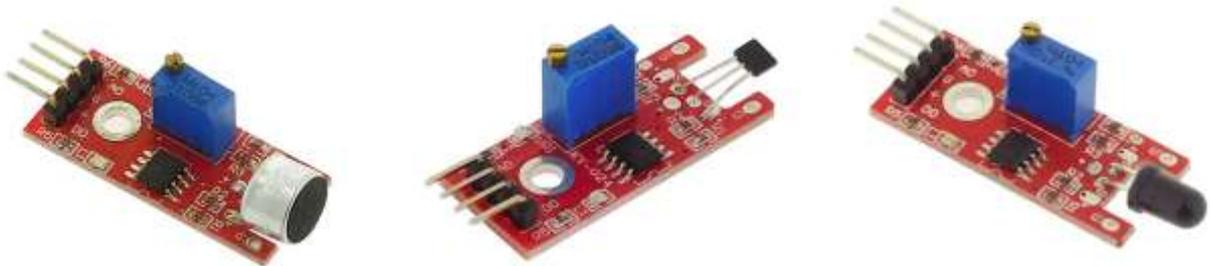
Cher client, merci d'avoir acheté ce kit de capteurs et de modules à usage didactique.

Les caractéristiques, le câblage et les exemples de codes sont indiqués dans ce guide.

Certains exemples de ce kit nécessitent du matériel supplémentaire non inclus :

- Pour les modules à Leds, il est impératif d'ajouter des résistances de limitation permettant d'éviter leur destruction. (100, 150 et 180 Ω conseillés).
- Afin de facilement raccorder ces résistances aux modules et à la carte microcontrôleur, nous conseillons également l'utilisation d'une [plaque de montage rapide](#) avec [cordons de connexion M/M](#).

Le module récepteur IR peut être utilisé avec une télécommande IR. L'exemple inclus dans ce guide utilise la télécommande [ADA389](#). La télécommande [GT017](#) est également compatible.



Certains modules et capteurs de ce kit tel que le capteur à effet Hall, les micros, le détecteur de flamme disposent d'un potentiomètre permettant de régler le seuil de déclenchement du signal digital.

1. Module joystick analogique avec bouton-poussoir.

Ce module joystick comporte deux sorties analogiques pour les axes X et Y et une sortie digitale pour le bouton poussoir Z.

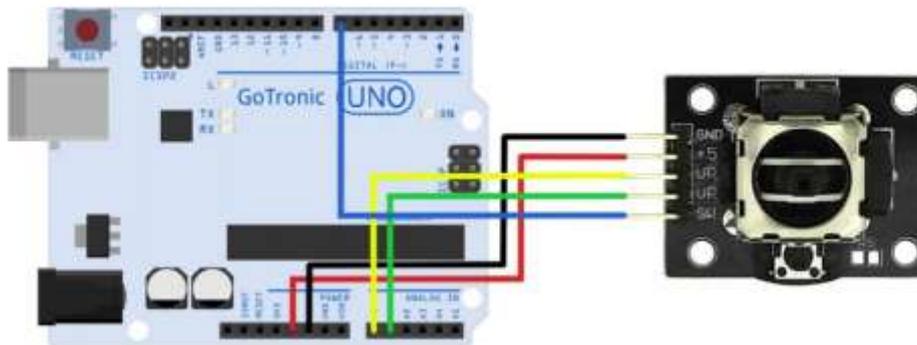


Caractéristiques :

- Alimentation : 5 Vcc
- Sorties :
 - 2 x sorties analogiques pour les axes X et Y
 - 1 x sortie digitale pour le bouton-poussoir Z
- Tension de sortie : entre 0 et 5 Vcc suivant la position (2,5 V en position milieu)
- Dimensions : 37 x 25 x 32 mm

Câblage :

Carte Uno	Module joystick
GND	GND
+ 5 Vcc	+ 5 Vcc
A0	VRx
A1	VRy
7	SW



Exemple de programme :

L'exemple de code suivant, à copier dans l'IDE Arduino®, permet d'afficher la tension analogique des axes X et Y. Cette tension varie en fonction de la position du joystick.

```
int axeX = A0; // signal de l'axe X sur entrée A0
int axeY = A1; // signal de l'axe Y sur entrée A1
int BP7 = 7; // bouton-poussoir en broche 7

float X, Y; // valeur des axes X et Y en Volt
bool bouton; // valeur du bouton (poussé ou non)

void setup ()
{
  pinMode (axeX, INPUT); // définition de A0 comme une entrée
  pinMode (axeY, INPUT); // définition de A1 comme une entrée
  pinMode (BP7, INPUT); // définition de la broche 7 comme une entrée
  digitalWrite(BP7, HIGH); // Activation de la résistance de Pull-Up

  Serial.begin (9600);
}

void loop ()
{
  X = analogRead (axeX) * (5.0 / 1023.0);
  Y = analogRead (axeY) * (5.0 / 1023.0);
  bouton = digitalRead (BP7);

  Serial.print ("Axe X:");
  Serial.print (X, 4);
  Serial.print ("V, ");

  Serial.print ("Axe Y:");
  Serial.print (Y, 4);
  Serial.print ("V, ");

  Serial.print ("Bouton:");

  if (bouton==1)
  {
    Serial.println (" Aucune pression sur le bouton poussoir ");
  }
  else
  {
    Serial.println (" Bouton-poussoir actif ");
  }
  delay (500);
}
```

2. Module relais.

Ce module commute un relais à partir d'un signal digital. Le relais permet la commande d'appareils nécessitant une puissance importante (moteur, solénoïde, etc.).

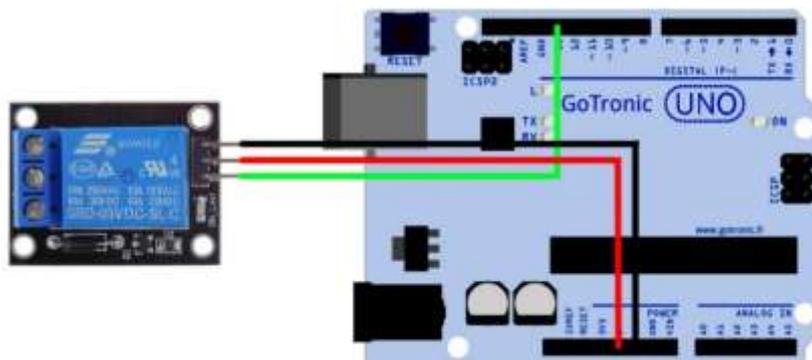


Caractéristiques :

- Alimentation : 5 Vcc
- Sortie : 1 contact RT 30 Vcc/10 A
- Connecteurs : 3 broches (Vcc, GND et Signal)
- Dimensions : 34 x 27 mm

Câblage :

Carte Uno	Module Relais
GND	-
+ 5 Vcc	+ (broche centrale)
13	S



Exemple de programme :

L'exemple de code suivant, à copier dans l'IDE Arduino®, fait commuter le relais et allumer la LED 13 intégrée à la carte Uno toutes les secondes.

```
int led = 13; // Définition de la broche 13 en led

void setup()
{
    pinMode(led, OUTPUT); // Définition de la broche 13 en sortie
}
void loop()
{
    digitalWrite(led, HIGH); // Passe la sortie 13 à l'état haut
    delay(1000); // Temporisation d'une seconde
    digitalWrite(led, LOW); // Passe la sortie 13 à l'état bas
    delay(1000); // Temporisation d'une seconde
}
```

3 et 4. Modules à micro à électret.

Ces capteurs sonores didactiques sont constitués d'un micro et d'un amplificateur.

Ce capteur délivre un signal analogique et un signal digital en fonction de l'intensité sonore reçue.



Caractéristiques :

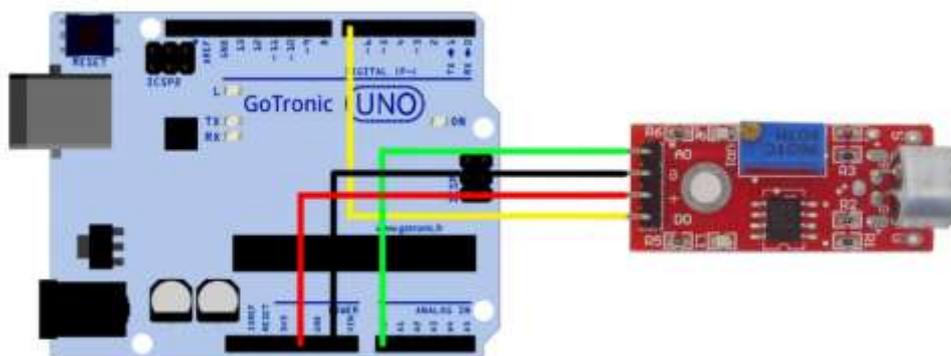
- Alimentation : 3,3 ou 5 Vcc
- Connecteurs : 3 broches (Vcc, GND et D0 et A0)
- Dimensions : 36 x 15 x 13 mm



Sens horaire: augmentation de la sensibilité
Sens anti-horaire: diminution de la sensibilité

Câblage :

Carte Uno	Modules Micro
7	S
A0	A0
5 Vcc	+
GND	-



Exemple de programme :

L'exemple de code suivant, à copier dans l'IDE Arduino®, allume la Led 13 intégrée à la carte Uno lorsque le capteur détecte un son assez fort. La sensibilité est réglable via le potentiomètre du module.

```
int Led = 13 ;
int buttonpin = 7;
int val = 0;
void setup ()
{
  pinMode (Led, OUTPUT) ;
  pinMode (buttonpin, INPUT) ;
}
void loop ()
{
  val = digitalRead(buttonpin);
  if (val == HIGH)
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
}
```

Second exemple de programme :

Ce second programme permet la lecture de la sortie analogique du module (A0). Cette lecture se fait dans le moniteur série de l'IDE Arduino®.

```
int sensorPin = A0;
int sensorValue = analogRead(A0);

void setup ()
{
  Serial.begin (9600);
}
void loop ()
{
  sensorValue = analogRead (sensorPin);
  delay (500);
  Serial.println (sensorValue, DEC);
}
```

5. Module suiveur de ligne.

Module suiveur de ligne basé sur un réflecteur optique et un amplificateur. Le seuil de déclenchement du signal digital est réglable via un potentiomètre ajustable.

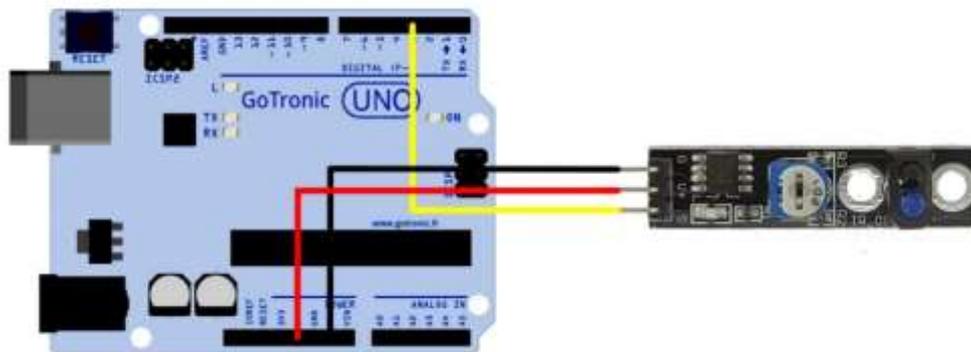


Caractéristiques :

- Alimentation : 3,3 à 5 Vcc
- Consommation : 20 mA sous 5 V
- Sortie :
 - état bas : ligne noire
 - état haut : ligne blanche
- Dimensions : 42 x 11 x 12 mm

Câblage :

Carte Uno	Module suiveur de ligne
3	5
5 Vcc	+ (broche du milieu)
GND	-



Exemple de programme :

L'exemple de code suivant, à copier dans l'IDE Arduino®, fait allumer la Led 13 intégrée à la carte Uno lorsque le module détecte une ligne noire.

```
int Led = 13;
int buttonpin = 3;
int val;
void setup()
{ pinMode(Led, OUTPUT);
  pinMode(buttonpin, INPUT);
}
void loop()
{
  val = digitalRead(buttonpin);
  if (val == HIGH)
  {
    digitalWrite(Led, LOW);
  }
  else {
    digitalWrite(Led, HIGH);
  }
}
```

6. Module détecteur d'obstacle.

Ce module équipé d'un émetteur et d'un récepteur infrarouge (IR) est basé sur un amplificateur NE555 et permet la détection d'obstacles sur une portée de 2 à 40 cm (réglable par deux potentiomètres).

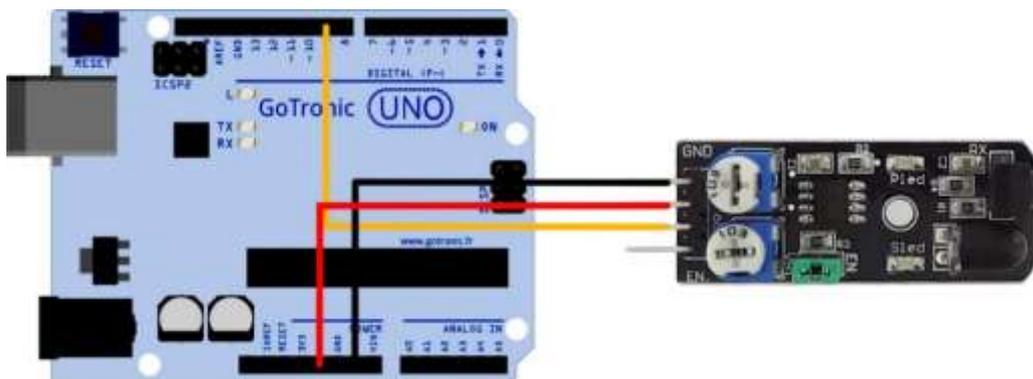


Caractéristiques :

- Alimentation : 3,3 à 5 Vcc
- Consommation : environ 20 mA
- Détection : 2 à 40 cm (réglage par deux potentiomètres)
- Angle de détection : 35°
- Sortie : 4 broches (Gnd, +, S et EN)
- Sortie digitale : état bas lors d'une détection
- Leds d'indication de détection
- Dimensions : 45 x 17 x 13 mm

Câblage :

Carte Uno	Détecteur d'obstacle
9	DAT
5 Vcc	+ (broche du milieu)
GND	-



Exemple de programme :

L'exemple de code suivant, à copier dans l'IDE Arduino®, permet d'afficher un « 1 » lorsqu'aucun obstacle n'est détecté et un « 0 » en cas de détection.

```
int count;
void setup () {
  Serial.begin (9600);
  pinMode (9, INPUT); // Sélection de la broche de sortie
}
void loop () {
  Serial.print ("Sensor: ");
  Serial.println (digitalRead(9)); // Affiche l'état de la sortie du
  capteur
  delay (500);
}
```

7. Module détecteur de flamme.

Module didactique basé sur un récepteur IR permettant la détection d'une flamme ou d'autres sources lumineuses de la même plage.

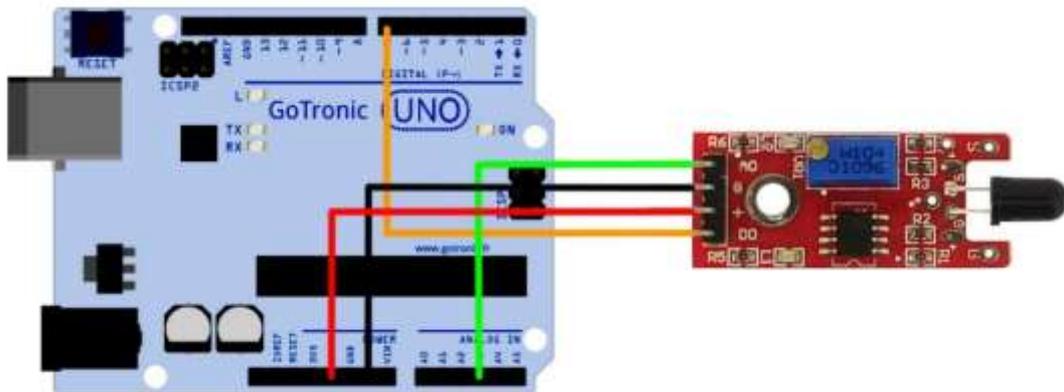


Caractéristiques :

- Alimentation : 5 Vcc
- Plage de mesure : 760 à 1100 nm
- Sorties : A0, Gnd, Vcc et D0
- Température de service : -40 °C à +85 °C
- Humidité de service : 30 à 90 % RH
- Dimensions : 42 x 16 x 15 mm

Câblage :

Carte Uno	Détecteur de flamme
7	D0
A3	A0
5 Vcc	+
GND	-



Exemple de programme :

L'exemple de code suivant, à copier dans l'IDE Arduino®, utilise les deux sorties du capteur (A0 et D0). La valeur de la sortie analogique du capteur est affichée dans le moniteur série de l'IDE Arduino® (CTRL+MAJ+M).

La broche digitale D0 permet d'allumer la Led 13 intégrée à la carte Uno dès que le capteur franchi un seuil. Ce seuil est ajustable via le potentiomètre intégré au module.

```
int Led = 13 ; // Déclaration de la 13 sur la broche 13
int buttonpin = 3; // Déclaration de l'entrée digitale sur 3
int analog = A3; // Déclaration de l'entrée analogique sur A3
int val ; // Déclaration d'une variable numérique
float sensor; // Lecture de la valeur analogique
void setup ()
{
  pinMode (Led, OUTPUT) ; // Définition de Led en tant que sortie
  pinMode (buttonpin, INPUT) ; // Définition de buttonpin en tant qu'entrée
  pinMode (analog, INPUT) ; // Définition de analog en tant qu'entrée
  Serial.begin(9600);
}
void loop ()
{
  sensor = analogRead(analog);
  Serial.println(sensor); // Affiche la température via le moniteur série
  val = digitalRead (buttonpin) ;
  if (val == HIGH) // Quand le capteur délivre un signal, la Led 13 s'allume
  {
    digitalWrite (Led, HIGH);
  }
  else
  {
    digitalWrite (Led, LOW);
  }
  delay(1000);
}
```

8. Module à effet Hall.

Ce module didactique à effet Hall permet la détection d'un champ magnétique à proximité du détecteur (un aimant par exemple).

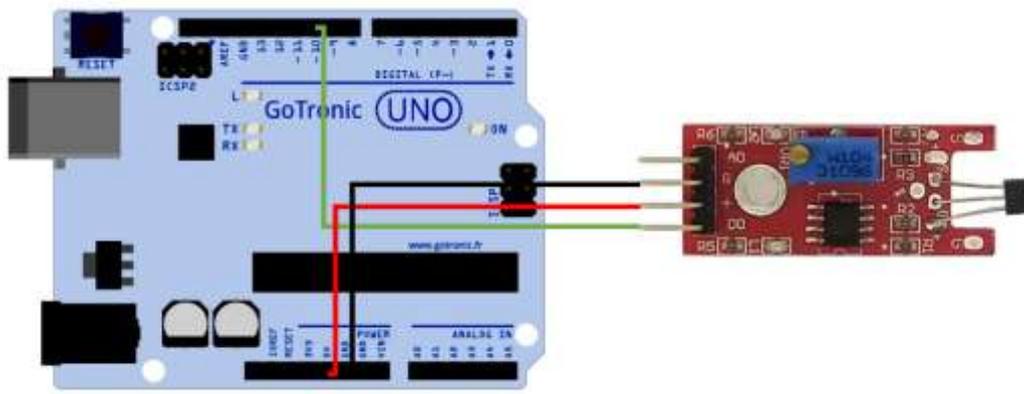


Caractéristiques :

- Alimentation : 5 Vcc
- Connecteurs : 3 broches (Vcc, GND et A0 et D0)
- Dimensions : 25 x 15 mm

Câblage :

Carte Uno	Module à effet Hall
10	10
5 Vcc	+
GND	-



Exemple de programme :

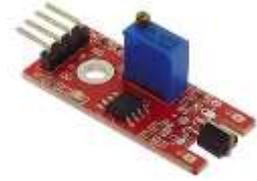
```
int capteur = 10;
int etat = 0;
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
  pinMode(capteur, INPUT);
}
void loop(){

  etat = digitalRead(capteur);
  if (etat == LOW) {
    digitalWrite(led, LOW);
  }
  else {
    digitalWrite(led, HIGH);
  }
}
```

9. Module tactile.

La sortie de ce capteur tactile change d'état lorsque l'on touche le capteur.

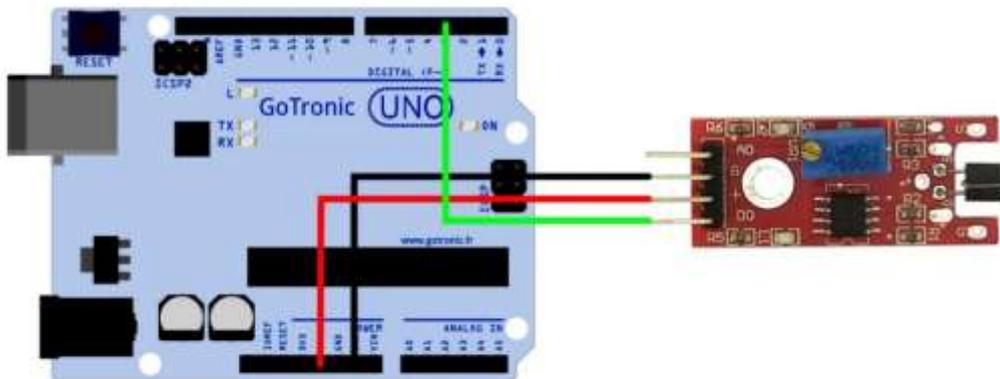


Caractéristiques :

- Alimentation : 5 Vcc
- Connecteurs : 3 broches (Vcc, GND et A0 et D0)
- Dimensions : 25 x 15 mm

Câblage :

Carte Uno	Module tactile
3	D0
5 Vcc	+
GND	-



Exemple de programme :

L'exemple de code suivant, à copier dans l'IDE Arduino®, permet d'allumer la Led 13 intégrée à la carte Uno lorsque vous touchez le capteur.

```
int capteur = 3;
int etat = 0;
int led = 13;

void setup() {
  pinMode(led, OUTPUT);
  pinMode(capteur, INPUT);
}
void loop(){
  etat = digitalRead(capteur);
  if (etat == LOW) {
    digitalWrite(led, LOW);
  }
  else {
    digitalWrite(led, HIGH);
  }
}
```

10. Capteur de température digital.

Ce capteur de température didactique basé sur une thermistance délivre un signal analogique et digital en fonction de la température mesurée.



Raccordement sur une entrée analogique ou une entrée digitale d'une carte Arduino® ou compatible.

Remarque : la formule suivante permet de calculer la température (pour la sortie analogique) :

$$1/T = A + B \ln(R) + C (\ln(R))^2$$

T est la t° en degrés Kelvin

R est la résistance en ohms

ln = logarithme népérien

A=0,001129148

B=0,000234125

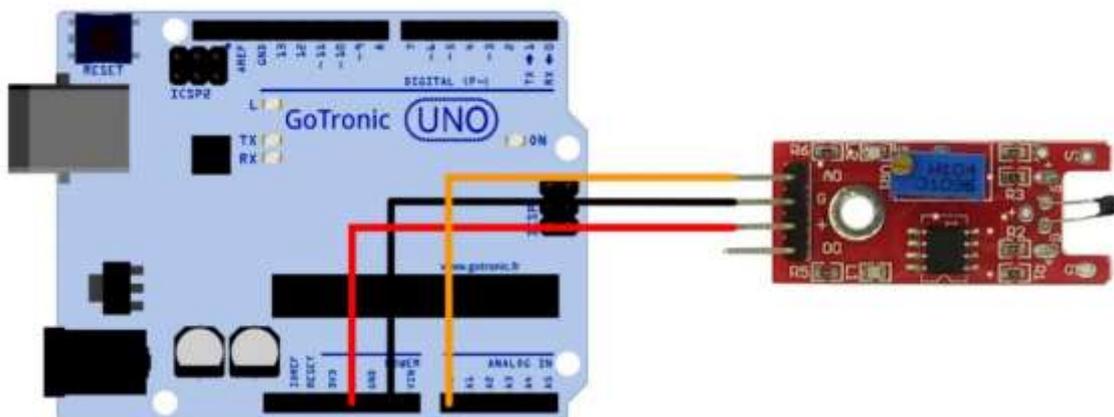
C=0,0000000876741

Caractéristiques :

- Alimentation: 5 Vcc
- Plage de mesure: -55 à +125 °C
- Précision: 0,5 °C
- Thermistance: MF52 (B = 3950)
- Sorties: A0, Gnd, Vcc et D0
- Dimensions: 39 x 16 x 15 mm

Câblage :

Carte Uno	Capteur de température
A0	A0
5 Vcc	+ (broche du milieu)
GND	-



Exemple de programme :

```
#include <math.h>

double Thermister(int RawADC) {
double Temp;
Temp = log(((10240000/RawADC) - 10000));
Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp ))* Temp );
Temp = 273.15 - Temp;
return Temp;
}

void setup() {
Serial.begin(9600);
}

void loop()
{ Serial.print(Thermister(analogRead(0)));
  Serial.println("c");
  delay(1000);
}
```

11. Module à buzzer actif.

Module à buzzer actif compatible Arduino®. Ne nécessite pas d'oscillateur externe.

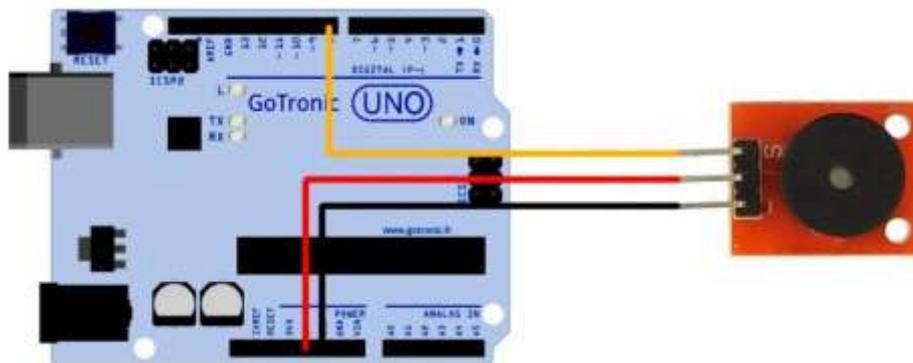
Caractéristiques :

- Alimentation : 3,3 et 5 Vcc
- Dimensions : 25 x 15 x 12 mm



Câblage :

Carte Uno	Module à buzzer
8	S
5 Vcc	+ (broche du milieu)
GND	-



Exemple de programme :

```
int speakerPin = 8; // Définition du buzzer sur la broche 8

void setup () {
  pinMode (speakerPin, OUTPUT); // Définition de la broche 8 en tant que sortie
}

void loop () {
  analogWrite (speakerPin, 255); // Signal haut sur broche 8
  delay (50); // pendant 50 ms
  analogWrite (speakerPin, 0); // Signal bas sur broche 8
  delay (10); // pendant 10 ms
}
```

12. Module à buzzer passif.

Module à buzzer passif compatible Arduino®. Ce module est similaire au précédent en apparence, il doit être cependant piloté par un signal carré et non par un signal continu classique.

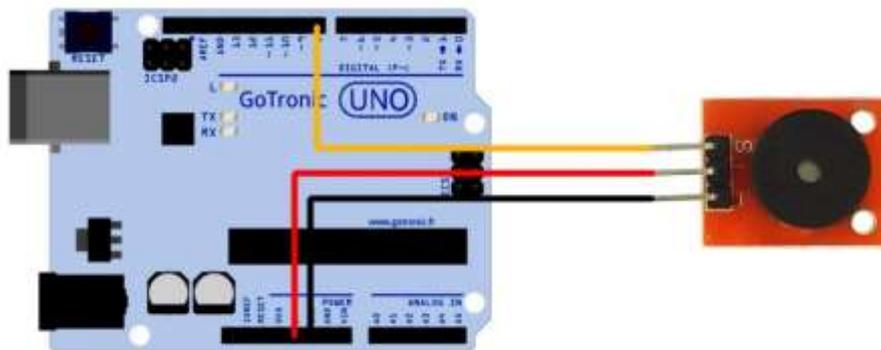


Branchement :

- Alimentation : 3,3 et 5 Vcc
- Dimensions : 25 x 15 x 12 mm

Câblage :

Carte Uno	Module à buzzer
8	S
5 Vcc	+ (broche du milieu)
GND	-



Exemple de programme :

```
int buzzer = 8 ;
void setup ()
{
  pinMode (buzzer, OUTPUT) ;
}
void loop ()
{
  unsigned char i, j;
  while (1)
  {
    for (i = 0; i <80; i++)
    {
      digitalWrite (buzzer, HIGH);
      delay (1) ;// Delay 1ms
      digitalWrite (buzzer, LOW);
      delay (1) ;// delay ms
    }
    for (i = 0; i <100; i++)
    {
      digitalWrite (buzzer, HIGH);
      delay (2) ;// delay 2ms
      digitalWrite (buzzer, LOW);
      delay (2) ;// delay 2ms
    }
  }
}
```

13. Module à LED RGB 5 mm.

Module à led RGB piloté à partir de 3 sorties digitales d'une carte Arduino® ou compatible. Chaque couleur peut être commandée par un signal PWM pour faire varier l'intensité lumineuse.



Remarque : nécessite l'ajout d'une résistance de limitation par couleur en fonction de la tension d'alimentation.

Caractéristiques :

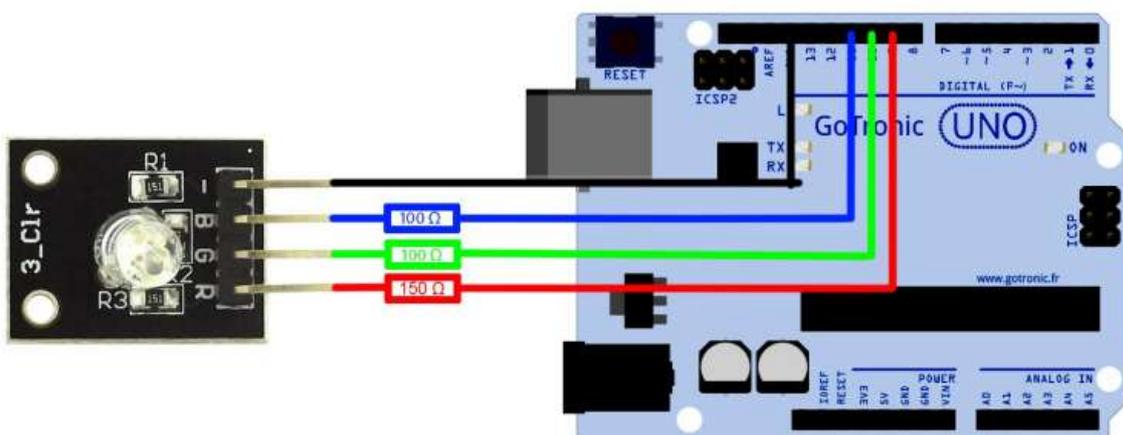
- Alimentation : 5 Vcc
- Sorties : R, G, B et Gnd
- Dimensions : 19 x 16 x 18 mm

Câblage :

Il est nécessaire de placer une résistance de limitation pour chaque couleur :

- 100 Ω pour la broche G (couleur verte).
- 150 Ω pour la broche R (couleur rouge).
- 100 Ω pour la broche B (couleur bleue).

Carte Uno	Module à Led RGB
9	R
10	G
11	B
GND	-



L'exemple de code suivant (à copier dans l'IDE Arduino®) va faire effectuer au module led RGB un cycle de couleurs différentes en activant les sorties digitales.

```
const int ledrouge=9;
const int ledverte=10;
const int ledbleue=11;

void setup () {
  pinMode(ledrouge, OUTPUT); // définition de la broche ledrouge en sortie
  pinMode(ledverte, OUTPUT); // définition de la broche ledverte en sortie
  pinMode(ledbleue, OUTPUT); // définition de la broche ledbleue en sortie
}

void loop () {
  // Exemple de couleurs

  // couleur Verte
  digitalWrite(ledrouge, LOW);
  digitalWrite(ledverte, HIGH);
  digitalWrite(ledbleue, LOW);
  delay(500);

  // couleur Rouge
  digitalWrite(ledrouge, HIGH);
  digitalWrite(ledverte, LOW);
  digitalWrite(ledbleue, LOW);
  delay(500);

  // couleur Blanc
  digitalWrite(ledrouge, HIGH);
  digitalWrite(ledverte, HIGH);
  digitalWrite(ledbleue, HIGH);
  delay(500);

  // couleur Magenta
  digitalWrite(ledrouge, HIGH);
  digitalWrite(ledverte, LOW);
  digitalWrite(ledbleue, HIGH);
  delay(500);

  // couleur Jaune
  digitalWrite(ledrouge, HIGH);
  digitalWrite(ledverte, HIGH);
  digitalWrite(ledbleue, LOW);
  delay(500);

  // OFF
  digitalWrite(ledrouge, LOW);
  digitalWrite(ledverte, LOW);
  digitalWrite(ledbleue, LOW);
  delay(2000);
}
```

Le signal PWM (Pulse width modulation) permet de gérer simplement le niveau de chaque couleur et de créer facilement des mélanges de couleurs. Chaque Led peut être réglée entre 0 et 255.

```
int ledrouge = 9;
int ledverte = 10;
int ledbleue = 11;

void setup()
{
  pinMode(ledrouge, OUTPUT); // définition de ledrouge en sortie
  pinMode(ledverte, OUTPUT); // définition de ledverte en sortie
  pinMode(ledbleue, OUTPUT); // définition de ledbleue en sortie
}

void loop()
{
  couleur (255, 0, 0);      //rouge
  delay(1000);
  couleur (0, 255, 0);     //verte
  delay(1000);
  couleur (0, 0, 255);     //bleu
  delay(1000);
  couleur (255, 255, 0);   //jaune
  delay(1000);
  couleur (80, 0, 50);     //violet
  delay(1000);
}

void couleur(int red, int green, int blue)
{
  analogWrite(ledrouge, red);
  analogWrite(ledverte, green);
  analogWrite(ledbleue, blue);
}
```

14. Module à Led RGB CMS.

Module à led RGB permettant d'obtenir une large palette de couleurs à partir de 3 sorties PWM d'une carte Arduino® ou compatible (nécessite une résistance de limitation par Led).



Caractéristiques :

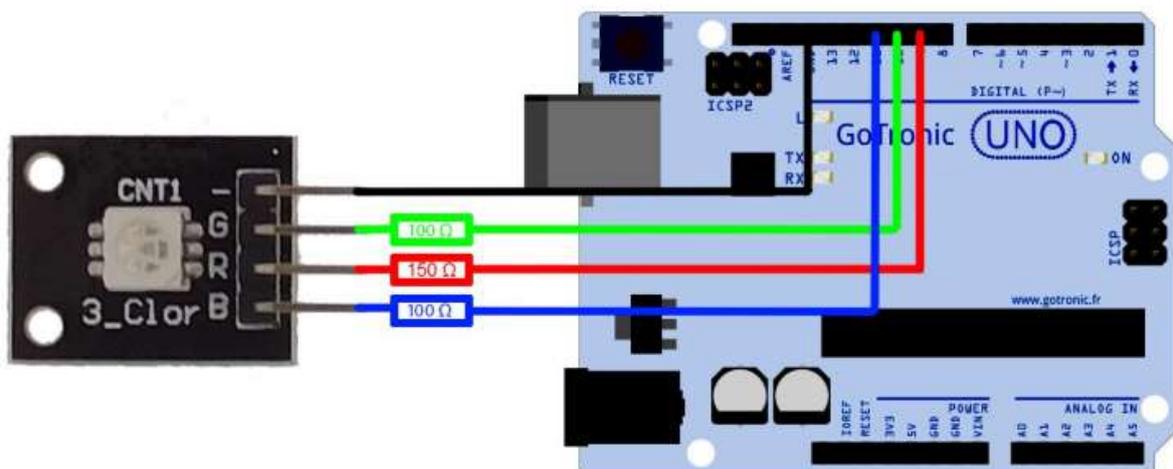
- Alimentation : 3,5 et 5 Vcc
- Sorties : GND, vert, rouge et bleu
- Dimensions : 20 x 16 x 7mm

Câblage :

Il est nécessaire de placer une résistance de limitation pour chaque couleur :

- 100 Ω pour la broche G (couleur verte).
- 150 Ω pour la broche R (couleur rouge).
- 100 Ω pour la broche B (couleur bleue).

Carte Uno	Module à Led RGB
9	R
10	G
11	B
GND	-



L'exemple de code suivant, à copier dans l'IDE Arduino®, va faire effectuer au module led RGB un cycle de couleurs.

```
const int ledrouge=9;
const int ledverte=10;
const int ledbleue=11;

void setup () {
  pinMode(ledrouge, OUTPUT); // définition de la broche ledrouge en sortie
  pinMode(ledverte, OUTPUT); // définition de la broche ledverte en sortie
  pinMode(ledbleue, OUTPUT); // définition de la broche ledbleue en sortie
}

void loop () {
  // Exemple de couleurs

  // couleur Verte
  digitalWrite(ledrouge, LOW);
  digitalWrite(ledverte, HIGH);
  digitalWrite(ledbleue, LOW);
  delay(500);

  // couleur Rouge
  digitalWrite(ledrouge, HIGH);
  digitalWrite(ledverte, LOW);
  digitalWrite(ledbleue, LOW);
  delay(500);

  // couleur Blanc
  digitalWrite(ledrouge, HIGH);
  digitalWrite(ledverte, HIGH);
  digitalWrite(ledbleue, HIGH);
  delay(500);

  // couleur Magenta
  digitalWrite(ledrouge, HIGH);
  digitalWrite(ledverte, LOW);
  digitalWrite(ledbleue, HIGH);
  delay(500);

  // couleur Jaune
  digitalWrite(ledrouge, HIGH);
  digitalWrite(ledverte, HIGH);
  digitalWrite(ledbleue, LOW);
  delay(500);

  // OFF
  digitalWrite(ledrouge, LOW);
  digitalWrite(ledverte, LOW);
  digitalWrite(ledbleue, LOW);
  delay(2000);
}
```

Le PWM (Pulse width modulation) permet de gérer simplement le niveau de chaque couleur et de créer facilement des mélanges de couleurs. Chaque led peut être réglée entre 0 et 255.

```
int ledrouge = 9;
int ledverte = 10;
int ledbleue = 11;

void setup()
{
  pinMode(ledrouge, OUTPUT); // définition de ledrouge en sortie
  pinMode(ledverte, OUTPUT); // définition de ledverte en sortie
  pinMode(ledbleue, OUTPUT); // définition de ledbleue en sortie
}

void loop()
{
  couleur (255, 0, 0);      //rouge
  delay(1000);
  couleur (0, 255, 0);     //verte
  delay(1000);
  couleur (0, 0, 255);     //bleu
  delay(1000);
  couleur (255, 255, 0);   //jaune
  delay(1000);
  couleur (80, 0, 50);     //violet
  delay(1000);
}

void couleur(int red, int green, int blue)
{
  analogWrite(ledrouge, red);
  analogWrite(ledverte, green);
  analogWrite(ledbleue, blue);
}
```

15 et 16. Modules à Led bicolore 5 mm et 3 mm.

Ces deux modules à Led bicolore verte et rouge de 5 mm/3 mm nécessitent l'ajout de 2 résistances sous peine de détruire la Led.



Afin de faciliter le câblage, une plaque de montage rapide de type [SD04N](#) avec cordons de connexions M/M [BBJ4](#) sont recommandés.

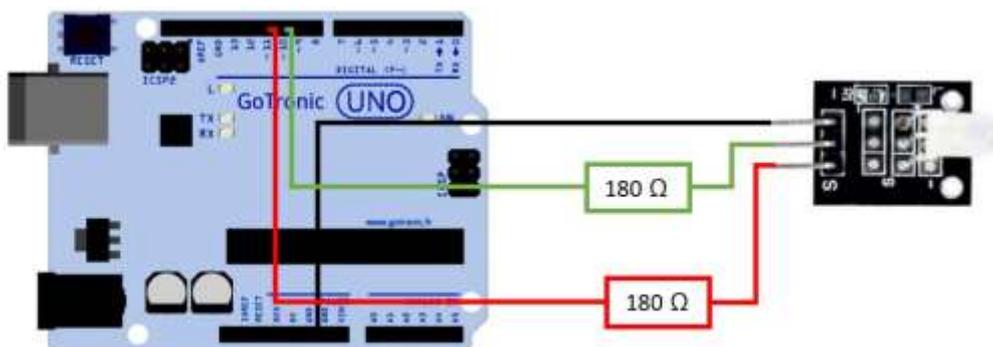
Caractéristiques :

- Couleurs : vert et rouge
- Diamètre : 3 et 5 mm
- Tension de seuil :
 - Vert : 2,3 à 2,6 Vcc
 - Rouge : 1,9 à 2,2 Vcc
- Courant : 20 mA

Remarque : dans notre exemple et malgré la différence de tension de seuil des deux couleurs, nous utiliserons des résistances de 180 Ω afin de limiter le courant.

Câblage :

Carte Uno	Module à Led
11	R
10	G
GND	-



Exemple de programme :

```
int ledrouge = 11;
int ledverte = 10;
int val;

void setup () {
  pinMode (ledrouge, OUTPUT);
  pinMode (ledverte, OUTPUT);
}
void loop ()
{
  for (val = 255; val > 0; val --)
  {
    analogWrite (11, val);
    analogWrite (10, 255 - val);
    delay (15);
  }
  for (val = 0; val < 255; val ++ )
  {
    analogWrite (11, val);
    analogWrite (10, 255 - val);
    delay (15);
  }
}
```

17. Module à interrupteur ILS.

Module à interrupteur ILS dont la sortie commute lorsqu'un aimant passe à proximité.

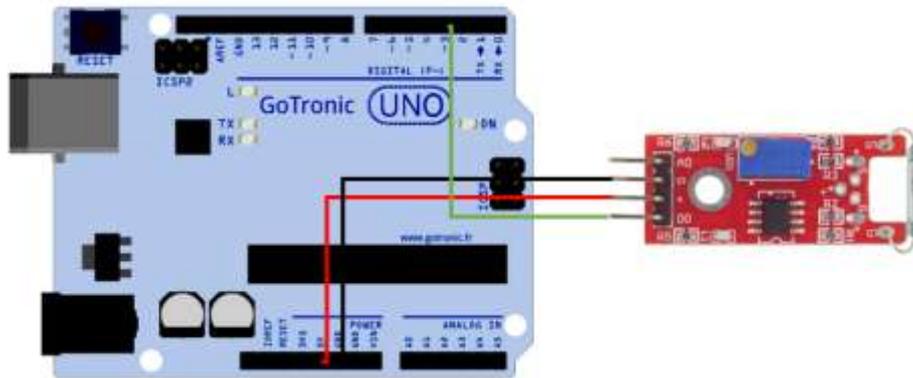


Caractéristiques :

- Alimentation : 5 Vcc
- Signal : digital et analogique
- Dimensions : 45 x 15 mm

Câblage :

Carte Uno	Module à ILS
5 V	+
GND	G
3	D0



Exemple de programme :

```
int Led = 13 ; // Utilisation de la Led 13 intégrée
int buttonpin = 3; // Définition de la broche 3 pour le capteur
int val ; // Définition d'une variable numérique
void setup ()
{
  pinMode (Led, OUTPUT) ; // Définition de Led en tant que sortie
  pinMode (buttonpin, INPUT) ; // Définition de buttonpin en tant qu'entrée
}
void loop () {
  val = digitalRead (buttonpin);
  if (val == HIGH) //
  {
    digitalWrite (Led, LOW);
  }
  else
  {
    digitalWrite (Led, HIGH);
  }
}
```

18. Module miniature à ILS.

Module miniature à interrupteur ILS dont la sortie commute lorsqu'un aimant passe à proximité.

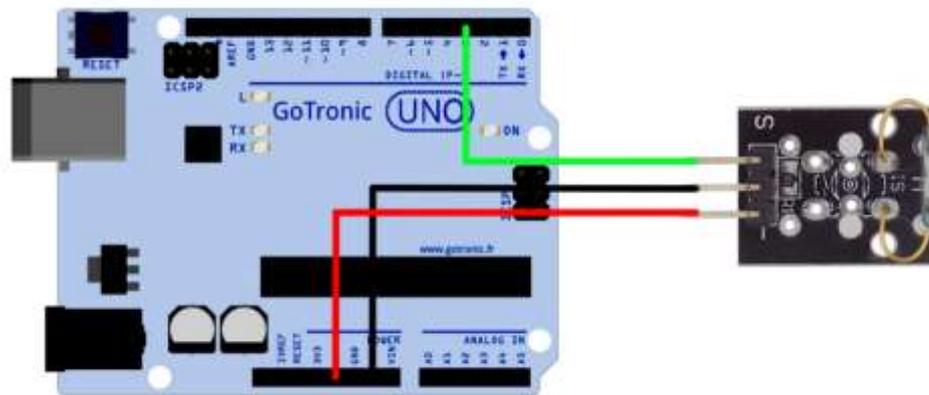


Caractéristiques :

- Alimentation : 5 Vcc
- Signal : digital
- Dimensions : 25 x 15 mm
- Poids : 2 g

Câblage :

Carte Uno	Module miniature à ILS
5 V	+
GND	G
3	S



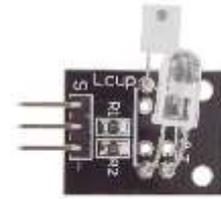
Exemple de programme :

L'exemple de code suivant, à copier dans l'IDE Arduino®, fait allumer la Led 13 intégrée à la carte Uno lorsque l'interrupteur ILS change d'état suite au passage de l'aimant.

```
int Led = 13 ; // Utilisation de la Led 13 intégrée
int buttonpin = 3; // Définition de la broche 3 pour le capteur
int val ; // Définition d'une variable numérique
void setup ()
{
  pinMode (Led, OUTPUT) ; // Définition de Led en tant que sortie
  pinMode (buttonpin, INPUT) ; // Définition de buttonpin en tant qu'entrée
}
void loop () {
  val = digitalRead (buttonpin) ;
  if (val == HIGH) //
  {
    digitalWrite (Led, LOW);
  }
  else
  {
    digitalWrite (Led, HIGH);
  }
}
```

19. Capteur de pulsations cardiaques.

Ce module permet la détection de pulsations cardiaques grâce à une Led IR et un phototransistor. Il suffit de placer son doigt entre les deux diodes et de visualiser le changement de la valeur analogique via le moniteur série.



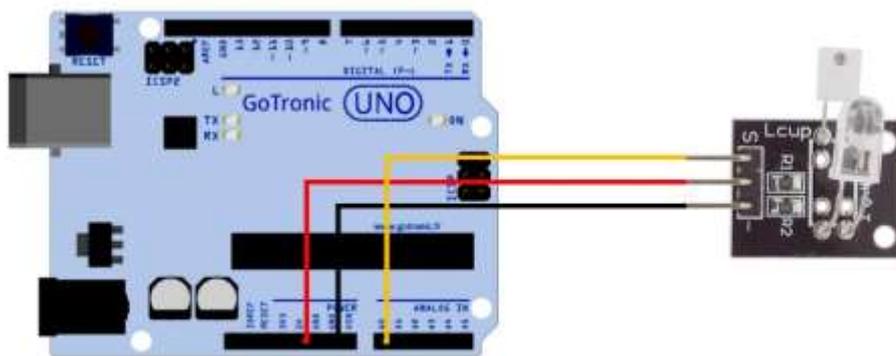
Caractéristiques :

- Alimentation : 5 Vcc
- Signal : digital
- Dimensions : 25 x 15 mm

Attention : module didactique, ne pas utiliser dans des applications médicales ou de sécurités.

Câblage :

Carte Uno	Capteur de pulsations
5 V	+
GND	G
A0	S



Exemple de programme :

```
int sensorPin = 0;
double alpha = 0.75;
int period = 100;
double change = 0.0;
double minval = 0.0;
void setup ()
{
  Serial.begin (9600);
}
void loop ()
{
  static double oldValue = 0;
  static double oldChange = 0;
  int rawValue = analogRead (sensorPin);
  double value = alpha * oldValue + (1 - alpha) * rawValue;
  Serial.print (rawValue);
  Serial.print (",");
  Serial.println (value);
  oldValue = value;
  delay (period);
}
```

20. Module à Led 7 couleurs.

Ce module à Led permet de générer 7 couleurs de façon aléatoire.

Ce module nécessite une résistance de 180 Ω à placer entre la sortie digitale de la carte Uno et la broche Signal du module à Led.

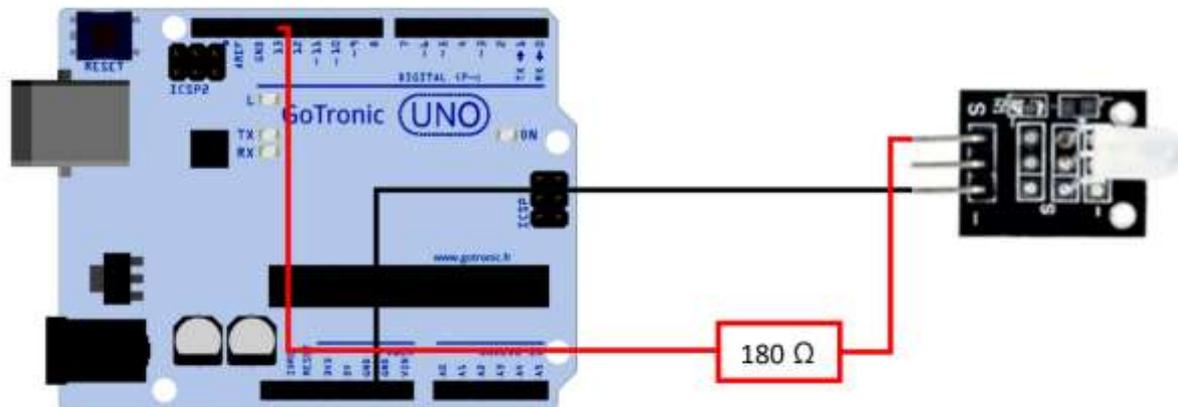


Caractéristiques :

- Tension de seuil de la Led : 4,5 Vcc
- Diamètre : 5 mm

Câblage :

Carte Uno	Module à Led
GND	G
13	S



Exemple de programme :

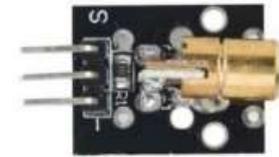
```
void setup() {
  pinMode(13, OUTPUT);
}
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
}
```

21. Module laser.

Module laser rouge 1 mW se raccordant à une sortie digitale d'un microcontrôleur Arduino® ou compatible.

Caractéristiques :

- Alimentation : 5 Vcc
- Longueur d'onde : 650 nm
- Couleur : rouge
- Laser de classe 3B
- Dimensions : 27 x 15 mm

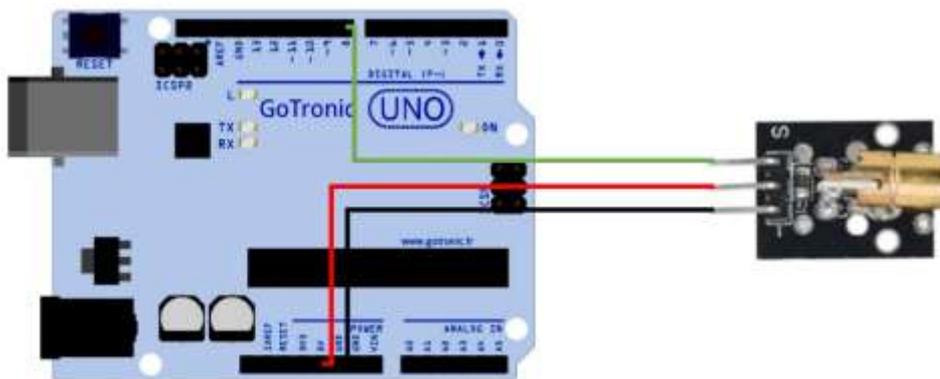


Ne pas diriger le rayon vers les yeux.



Câblage :

Carte Uno	Module laser
5 V	+
GND	G
8	S



Exemple de programme :

```
int laser = 8;
void setup ()
{
  pinMode (laser, OUTPUT);
}
void loop () {
  digitalWrite (laser, HIGH);
  delay (1000);
  digitalWrite (laser, LOW);
  delay (1000);
}
```

22. Module à bouton-poussoir.

Ce module est un simple bouton-poussoir permettant de faire commuter une entrée digitale d'une carte Arduino® ou compatible.

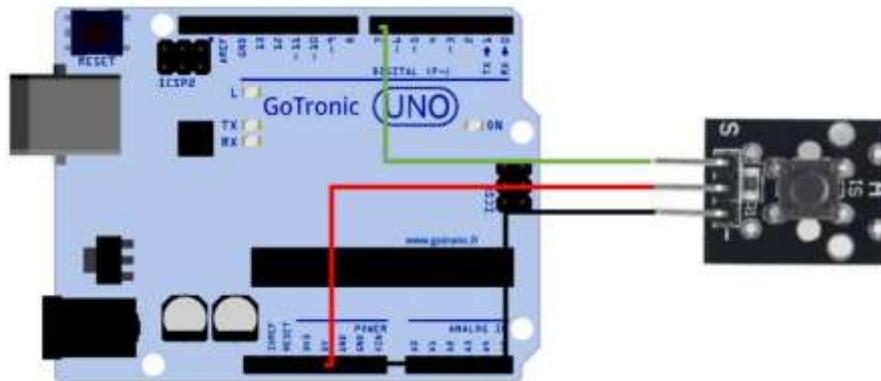


Caractéristiques :

- Alimentation : 5 Vcc
- Sortie : digitale
- Dimensions : 19 x 16 x 10 mm

Câblage :

Carte Uno	Bouton-poussoir
5 V	+
GND	G
7	S



Exemple de programme:

```
int Led = 13;
int buttonpin = 7;
int val;
void setup ()
{
  pinMode (Led, OUTPUT);
  pinMode (buttonpin, INPUT);
}
void loop ()
{
  val = digitalRead (buttonpin);
  if (val == HIGH)
  {
    digitalWrite (Led, LOW);
  }
  else
  {
    digitalWrite (Led, HIGH);
  }
}
```

23. Module détecteur de chocs.

Ce module est un capteur de vibration dont la sortie passe de l'état bas au repos à l'état haut (impulsions) lors d'une vibration ou d'un choc suffisamment fort.

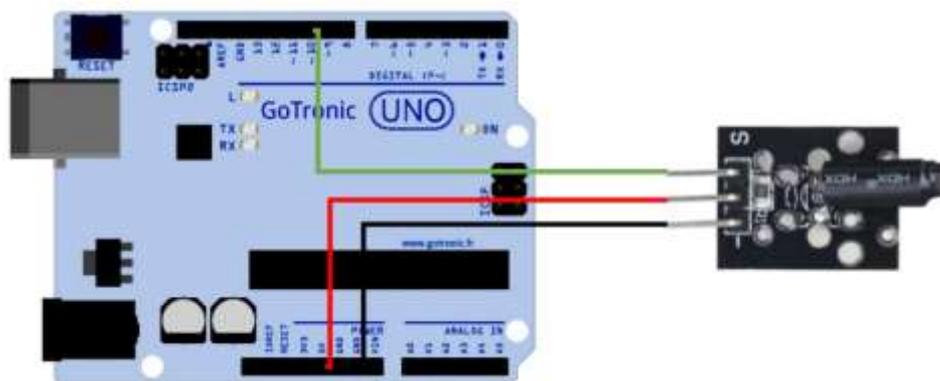


Caractéristiques :

- Alimentation : 5 Vcc
- Sortie : digitale
- Dimensions : 23 x 15 x 8 mm

Câblage :

Carte Uno	Détecteur de chocs
5 V	+
GND	G
10	S



Exemple de programme :

```
int led = 13 ; // Declaration de la broche de sortie LED
int capteur = 10; // Declaration de la broche d'entree du capteur du choc
int val; // Variable temporaire de stockage des informations
void setup ()
{
  pinMode (led, OUTPUT) ; // Initialisation de la broche de sortie
  pinMode (capteur, INPUT) ; // Initialisation de la broche du capteur
  digitalWrite(capteur, HIGH); // Activation de la resistance de Pullup
}
void loop (){
  val = digitalRead (capteur) ; // Lecture de la valeur du signal
  if (val == HIGH) // Si un signal est detecte, la LED s'allume
  {
    digitalWrite (led, LOW);
  }
  else
  {
    digitalWrite (led, HIGH);
  }
}
```

24. Module à encodeur rotatif.

Module encodeur rotatif à but didactique permettant la génération d'impulsions électriques en fonction de la rotation. Ce module se raccorde sur deux entrées digitales d'une carte type Arduino® ou compatible.

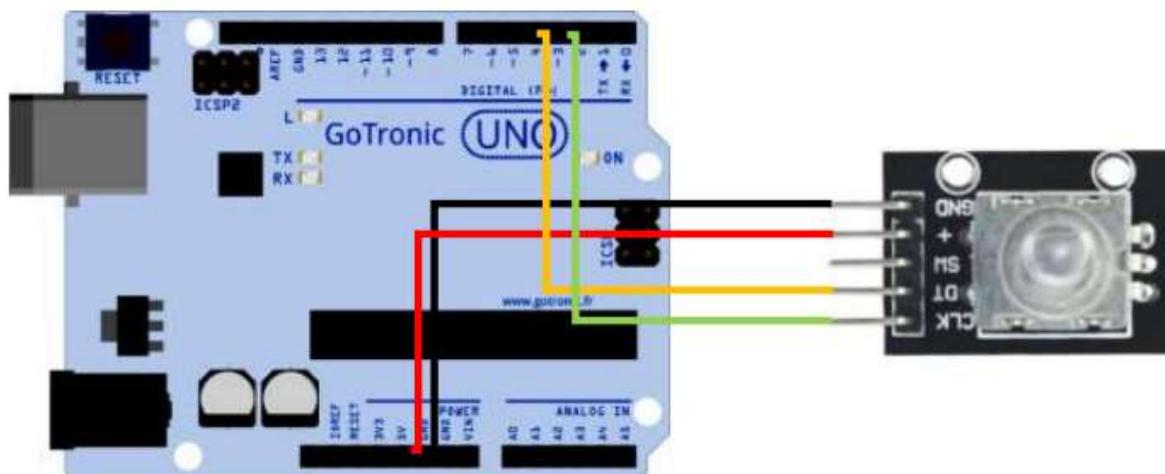


Caractéristiques :

- Alimentation : 5 Vcc
- Sorties :
 - CLK et DT : sorties encodeur
 - SW : bouton-poussoir
 - GND et + : alimentation 5 Vcc et masse
- Nombre d'impulsions : 20
- Dimensions : 32 x 20 x 30 mm

Câblage :

Carte Uno	Module encodeur
5 V	+
GND	G
3	CLK
4	DT



Remarque : Si vous souhaitez utiliser le bouton-poussoir, il suffit de le raccorder sur une entrée digitale du microcontrôleur avec une résistance de tirage de 10 k

Exemple de programme :

```
int A = 3; // Connecté à CLK
int B = 4; // Connecté à DT
int encoderPosCount = 0;
int ALast;
int aVal;
boolean bcW;
void setup() {
  pinMode (A, INPUT);
  pinMode (B, INPUT);
  ALast = digitalRead(A);
  Serial.begin (9600);
}
void loop() {
  aVal = digitalRead(A);

  if (aVal != ALast){ // L'axe est en rotation
    // si l'axe est en rotation, le sens doit être déterminé
    // La broche DT (B) doit être lue.

    if (digitalRead(B) != aVal) { // Signale que A change d'état avant B - la
// rotation est donc horaire
encoderPosCount ++;
bcW = true;

} else { // Sinon, c'est que B change avant A, la rotation est antihoraire
bcW = false;
encoderPosCount--;
}
Serial.print ("Rotation: ");
if (bcW){
Serial.println ("Sens horaire");
}else{
Serial.println("Sens anti-horaire");
}
Serial.print("Position de l'encodeur: ");
Serial.println(encoderPosCount);

}
ALast = aVal;
}
```

25. Module capteur d'inclinaison avec Led rouge (x2).

Ce module comporte un capteur d'inclinaison avec une Led rouge. Il communique avec un microcontrôleur type Arduino® ou compatible via deux broches digitales.

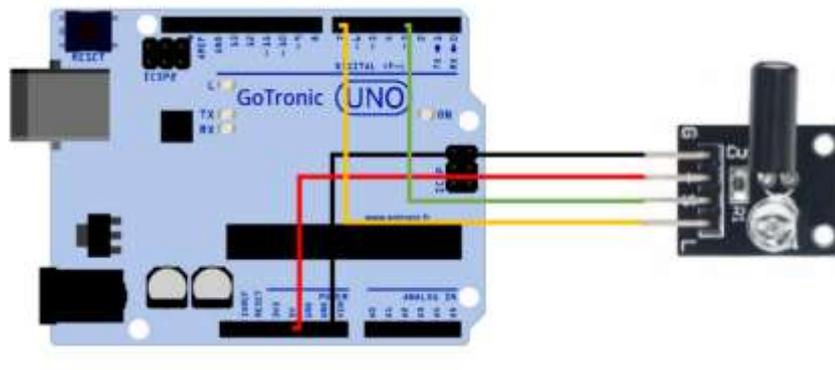


Caractéristiques :

- Alimentation : 5 Vcc
- Sortie : digitale

Câblage :

Carte Uno	Capteur d'inclinaison
5 V	+
GND	G
3	S (signal)
7	L (led)



Exemple de programme :

Ce programme permet de faire allumer la Led intégrée au module lors de son inclinaison.

```
int Led = 7;
int Capteur = 3;
int val;
void setup ()
{
  pinMode (Led, OUTPUT);
  pinMode (Capteur, INPUT);
  digitalWrite(Capteur, HIGH);
}
void loop ()
{
  val = digitalRead (Capteur);
  if (val == HIGH)
  {
    digitalWrite (Led, LOW);
  }
  else
  {
    digitalWrite (Led, HIGH);
  }
}
```

26. Module capteur d'inclinaison.

Ce module comporte un capteur d'inclinaison et communique avec un microcontrôleur type Arduino® ou compatible via une broche digitale.

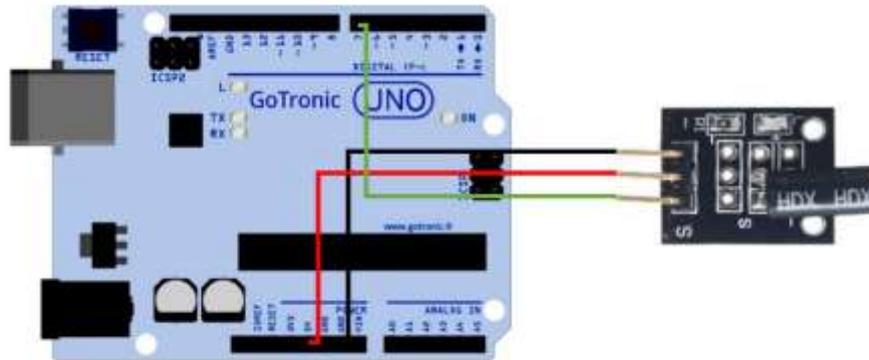


Caractéristiques :

- Alimentation : 5 Vcc
- Sortie : digitale

Câblage :

Carte Uno	Capteur d'inclinaison
5 V	+
GND	G
7	S



Exemple de programme :

Ce programme permet de faire allumer la Led intégrée à la carte Uno lors de son inclinaison.

```
int Led = 13;
int Capteur = 7;
int val;
void setup ()
{
  pinMode (Led, OUTPUT);
  pinMode (Capteur, INPUT);
  digitalWrite (Capteur, HIGH);
}
void loop ()
{
  val = digitalRead (Capteur);
  if (val == HIGH)
  {
    digitalWrite (Led, LOW);
  }
  else
  {
    digitalWrite (Led, HIGH);
  }
}
```

27. Module capteur de vibrations.

Ce module comporte un capteur de vibrations et communique avec un microcontrôleur type Arduino® ou compatible via une broche digitale.

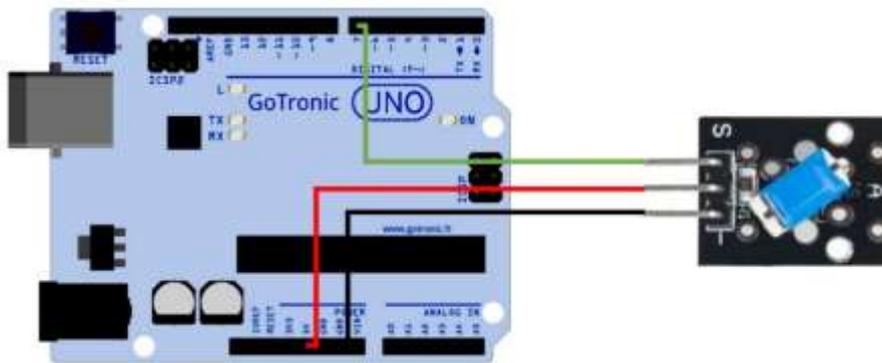


Caractéristiques :

- Alimentation : 5 Vcc
- Sortie : digitale
- Dimensions : 19 x 15 x 8 mm

Câblage :

Carte Uno	Capteur de vibrations
5 V	+
GND	G
7	S



Exemple de programme :

Ce programme permet de faire allumer la Led intégrée à la carte Uno lors de vibrations.

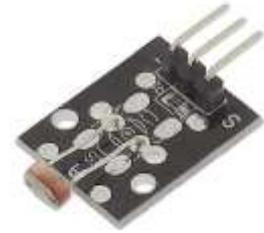
```

int Led = 13;
int Capteur = 7;
int val;
void setup ()
{
  pinMode (Led, OUTPUT);
  pinMode (Capteur, INPUT);
  digitalWrite (Capteur, HIGH);
}
void loop ()
{
  val = digitalRead (Capteur);
  if (val == HIGH)
  {
    digitalWrite (Led, LOW);
  }
  else
  {
    digitalWrite (Led, HIGH);
  }
}

```

28. Module capteur de luminosité (LDR).

Module à usage didactique basé sur une LDR. La résistance varie en fonction de la quantité de lumière reçue. Le module se raccorde sur une entrée analogique d'une carte Arduino® ou compatible.

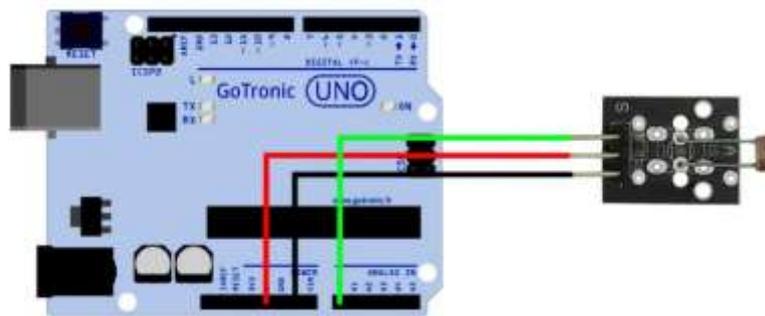


Caractéristiques :

- Alimentation : 5 Vcc
- Sorties : Signal, Vcc, GND
- Valeur de la résistance :
 - > 8 K Ω dans l'obscurité
 - < 200 Ω en extérieur

Câblage :

Carte Uno	Module LDR
5 V	+
GND	G
A0	S



Exemple de programme :

L'exemple de code suivant permet d'afficher la valeur de la résistance de la LDR dans le moniteur série et fait clignoter la led de la carte UNO en fonction de la luminosité.

```
int sensorPin = A0; // Définition de la broche d'entrée pour le signal
int ledPin = 13; // Définition de la broche de la Led 13
int sensorValue = 0; // Définition d'une variable

void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  sensorValue = analogRead(sensorPin);
  digitalWrite(ledPin, HIGH);
  delay(sensorValue);
  digitalWrite(ledPin, LOW);
  delay(sensorValue);
  Serial.println(sensorValue, DEC);
}
```

29. Module capteur de température et d'humidité (DHT11).

Ce capteur de température et d'humidité basé sur le circuit DHT11 communique avec un microcontrôleur type Arduino® ou compatible via une sortie digitale.



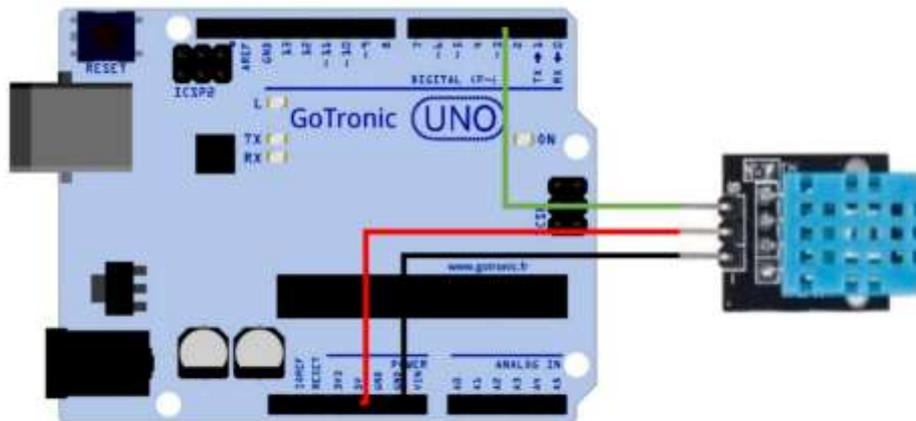
Remarque : ne fonctionne pas en dessous de 0°C.

Caractéristiques :

- Alimentation : +5 Vcc
- Plage de mesure :
 - température : 0 à 50 °C
 - humidité : 20 à 90 % RH
- Précision :
 - température : ± 2 °C
 - humidité : ± 5 % RH
- Sorties : S, Vcc, Gnd

Câblage :

Carte Uno	Capteur de température DHT11
5 V	+
GND	G
2	S



30. Capteur à effet Hall à sortie analogique.

Module à effet Hall délivrant une tension analogique en fonction du champ magnétique se situant à proximité (un aimant par exemple).

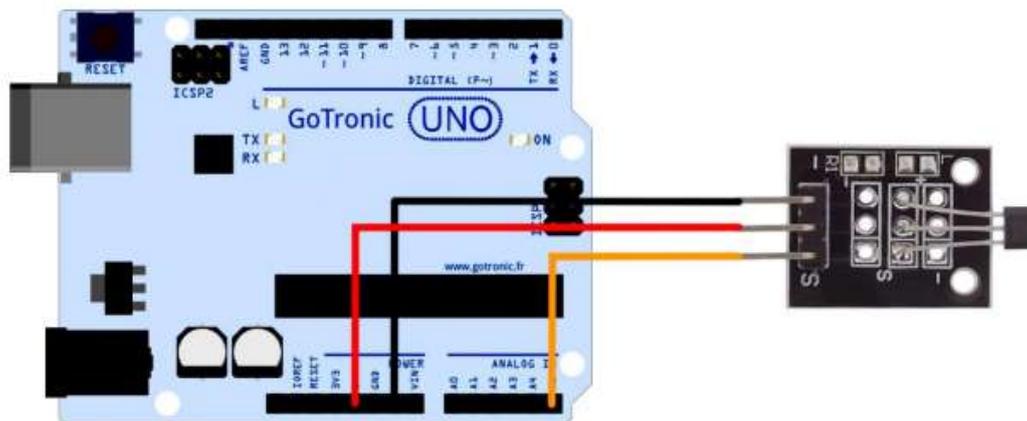


Caractéristiques :

- Alimentation : +5 Vcc
- Signal de sortie : analogique
- Dimensions : 30 x 21 mm

Câblage :

Carte Uno	Capteur à effet Hall
5 V	+
GND	G
A5	S



Exemple de programme :

L'exemple de code suivant fait clignoter la LED de la carte UNO et indique l'état analogique (de 0 à 1023) de la sortie du capteur en fonction du champ magnétique à proximité.

```
int sensorPin = A5; // Définition de la broche d'entrée
int ledPin = 13; // Définition de la Led intégrée en broche 13
int sensorValue = 0; // Variable permettant de stocker les données du
//capteur

void setup () {
  pinMode (ledPin, OUTPUT);
  Serial.begin (9600);
}

void loop () {
  sensorValue = analogRead (sensorPin);
  digitalWrite (ledPin, HIGH);
  delay (sensorValue);
  digitalWrite (ledPin, LOW);
  delay (sensorValue);
  Serial.println (sensorValue, DEC);
}
```

31. Capteur à effet Hall à sortie numérique.

Module didactique basé sur un capteur à effet hall A3144 permettant de capter un champ magnétique (lorsque l'on approche un aimant par exemple).



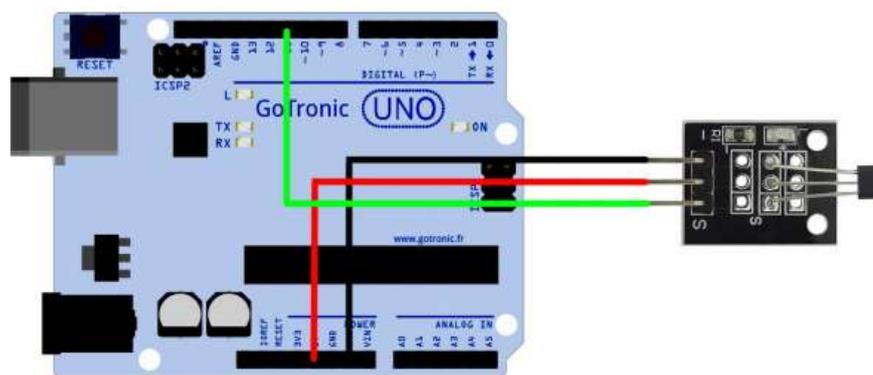
Il se différencie du capteur analogique par une résistance et une led intégrées au module.

Caractéristiques :

- Alimentation : 5 Vcc
- Sortie : digitale
- Dimensions : 25 x 15mm

Câblage :

Carte Uno	Capteur à effet Hall
5 V	+
GND	G
11	S



Exemple de programme :

L'exemple de code suivant, à copier dans l'IDE Arduino®, fait allumer la Led 13 intégrée à la carte Uno lorsque l'aimant est proche du capteur à effet Hall.

```
int Led = 13 ; // Déclaration de la Led en broche 13
int capteur = 11 ; // Déclaration du capteur en broche 10
int val ; // Déclaration d'une variable numérique
void setup ()
{
  pinMode (Led, OUTPUT) ; // Définition de la broche Led en tant que sortie
  pinMode (capteur, INPUT) ; // Définition du capteur en tant qu'entrée.
}
void loop ()
{
  val = digitalRead (capteur) ;
  if (val == LOW) { // Si le capteur passe à l'état bas
    digitalWrite (Led, HIGH) ; // // La Led s'allume
  }
  else {
    digitalWrite (Led, LOW) ; // Sinon la Led est éteinte
  }
}
```

32. Capteur de température DS18B20.

Ce capteur de température, basé sur le DS18B20, permet de mesurer une température ambiante de -55°C à $+125^{\circ}\text{C}$.

Ce module se raccorde sur une entrée digitale d'une carte compatible Arduino®.

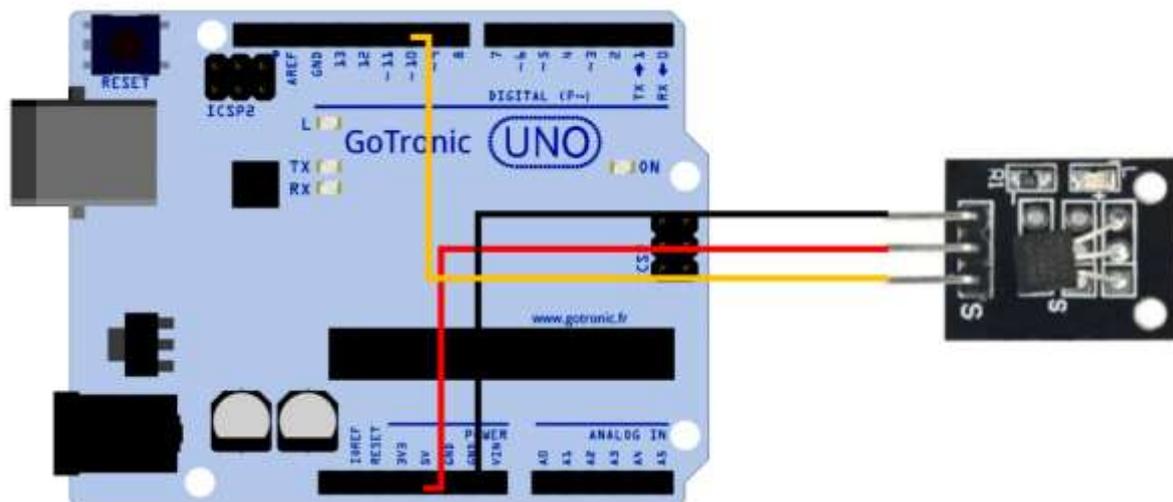


Caractéristiques :

- Alimentation : 5 Vcc
- Plage de mesure : -55 à $+125^{\circ}\text{C}$
- Précision : $0,5^{\circ}\text{C}$
- Dimensions : $23 \times 21 \times 10$ mm

Câblage :

Carte Uno	Capteur de température
5 V	+
GND	G
10	S



Exemple de programme :

L'utilisation de ce module nécessite l'ajout de deux bibliothèques à l'IDE Arduino® :

- OneWire.h
- DallasTemperature.h

Ces deux bibliothèques se téléchargent via le gestionnaire de bibliothèque.

L'exemple de code ci-dessous est installé automatiquement avec la bibliothèque DallasTemperature.

Vous pouvez le retrouver directement dans les exemples de l'IDE sous le nom « Simple » (situé dans exemples -> DallasTemperature).

Ce code permet d'afficher la température dans le moniteur série de l'IDE Arduino®.

```
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 10

OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire);

void setup(void)
{
  Serial.begin(9600);
  Serial.println("Dallas Temperature IC Control Library Demo");
  sensors.begin();
}

void loop(void)
{
  Serial.print("Requesting temperatures...");
  sensors.requestTemperatures(); // Send the command to get temperatures
  Serial.println("DONE");
  Serial.print("Temperature for the device 1 (index 0) is: ");
  Serial.println(sensors.getTempCByIndex(0));
}
```

33. Capteur de température analogique.

Ce capteur de température didactique basé sur une thermistance délivre un signal analogique en fonction de la température mesurée.

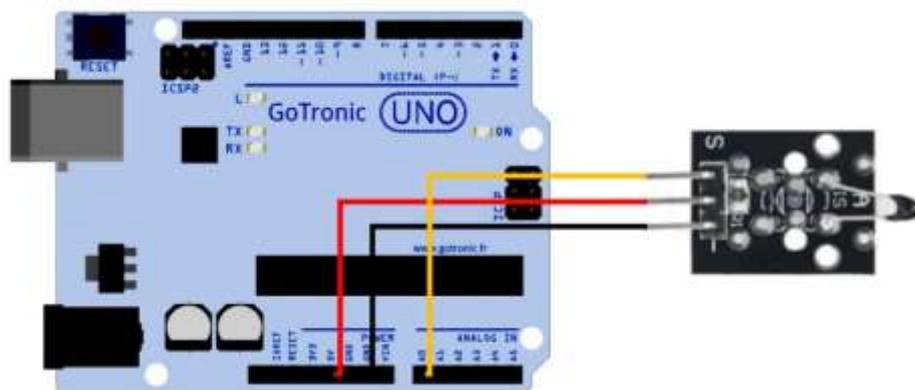
Caractéristiques :

- Alimentation : 5 Vcc
- Plage de mesure : -55 à +125 °C
- Précision : 0,5 °C
- Thermistance : MF52 (B = 3950)
- Sorties : S, Vcc, Gnd
- Dimensions : 19 x 16 x 7 mm



Câblage :

Carte Uno	Capteur de température
5 V	+
GND	G
A0	S



Exemple de programme :

```
#include <math.h>

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  double val = analogRead(0);
  double fencya = (val / 1023) * 5;
  // r/100=fencya/(3.3-fencya)
  //double r=(5-fencya)/fencya*10000;
  double r = fencya / (5 - fencya) * 10000;
  Serial.println( 1/(log(r/10000)/3950+1/(25+273.15))- 273.15);
  delay(1000);
}
```

34. Module à émetteur IR.

Module émetteur IR didactique pouvant être utilisé pour transmettre des données ou des commandes infrarouges jusqu'à 10 mètres.

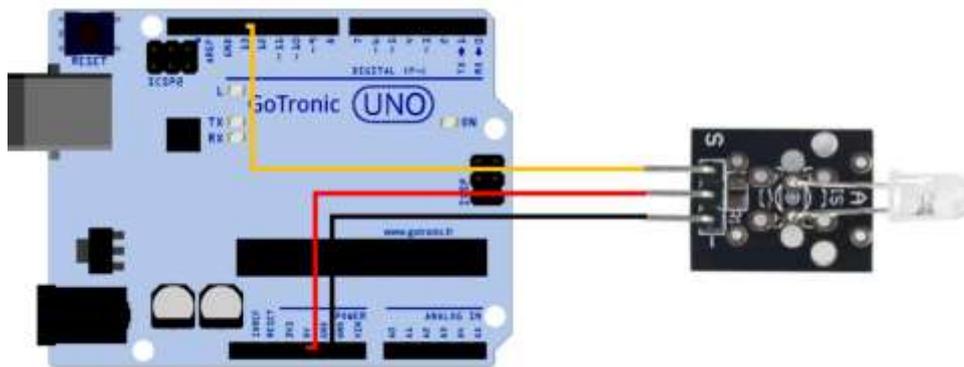


Caractéristiques :

- Alimentation : 5 Vcc
- Portée : 10 mètres maxi
- Connecteurs : 3 broches (Vcc, GND et Signal)
- Dimensions : 28 x 21 x 10 mm

Câblage :

Carte Uno	Emetteur IR
5 V	+
GND	G
13	S



Exemple de programme :

Cet exemple permet de faire allumer de manière répétée l'émetteur IR pendant une seconde puis de l'éteindre pendant une seconde. La visualisation de l'état de la led ne peut se faire qu'avec par exemple le module caméra d'un smartphone.

```
void setup() {
  pinMode(13, OUTPUT); // Définition de la broche 13 en sortie
}
void loop() {
  digitalWrite(13, HIGH); // Met la broche 13 à l'état haut
  delay(1000);
  digitalWrite(13, LOW); // Met la broche 13 à l'état bas
  delay(1000);
}
```

35. Module à récepteur IR.

Ce module didactique digital basé sur un récepteur IR permet la réception de signaux infrarouges (par exemple d'une télécommande).



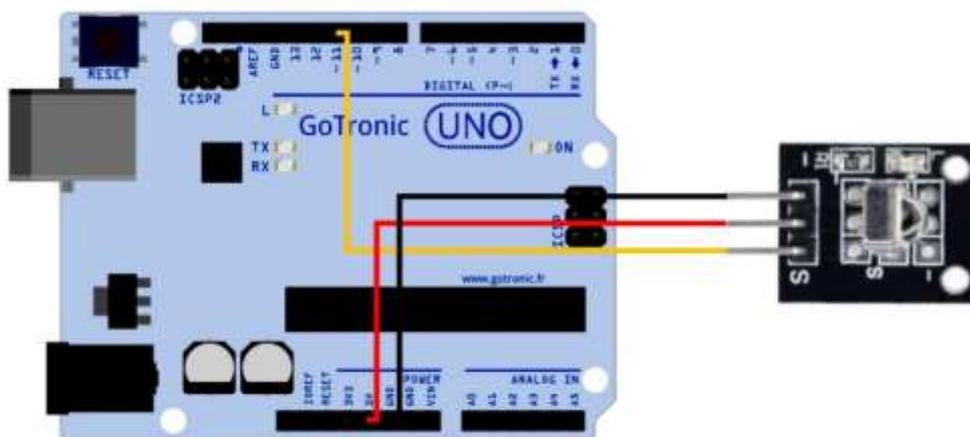
Caractéristiques :

- Alimentation : 5 Vcc
- Connecteurs : 3 broches (Vcc, GND et Signal)
- Sortie : digitale
- Dimensions : 28 x 21 x 12 mm

Remarque : ce module nécessite l'utilisation d'un émetteur IR. Voir le module précédent ou encore une télécommande IR de type [ADA389](#).

Câblage :

Carte Uno	Récepteur IR
5 V	+
GND	G
13	S



Installation de la bibliothèque pour la télécommande :

La bibliothèque IRremote.h est nécessaire lors d'une utilisation avec une télécommande IR de type ADA389. Cette bibliothèque est à installer manuellement.

Exemple : avec la télécommande ADA389 présenté plus haut, la bibliothèque se télécharge à cette adresse : <https://github.com/adafruit/Adafruit-NEC-remote-control-library>

Cette bibliothèque s'installe ensuite dans le dossier situé dans le dossier Documents/Arduino/Library

Exemple de programme :

```
#include <IRremote.h> // Déclaration de la librairie IR
int RECV_PIN = 11; // Définition de la broche d'entrée
IRrecv irrecv (RECV_PIN);
decode_results results;

void setup()
{
  Serial.begin (9600);
  irrecv.enableIRIn ();
}

void loop()
{
  if (irrecv.decode (& results)) {
    Serial.println (results.value, HEX);
    irrecv.resume ();
  }
}
```

36. Module à capteur fourche.

Module capteur optique à fourche dont la sortie digitale commute lorsque le faisceau lumineux est coupé. Applications : capteur de vitesse, etc.

Ce module se raccorde sur une entrée digitale d'un module à microcontrôleur (Arduino® par exemple).

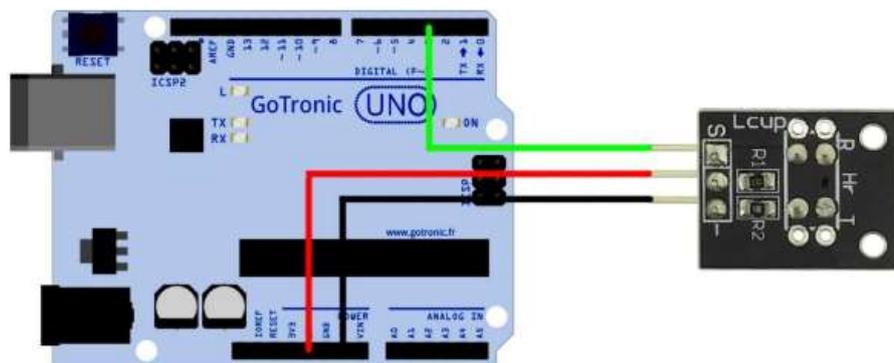


Caractéristiques :

- Alimentation : 5 Vcc
- Brochage : Vcc, GND et Signal
- Ouverture : 2mm
- Dimensions : 19 x 16 x 8 mm

Câblage :

Carte Uno	Capteur à fourche
5 V	+
GND	G
3	S



Exemple de programme :

```
int Led = 13; // Déclaration de la Led sur la broche 13
int capteur = 3; // Déclaration du capteur sur la broche 3
int val; // Déclaration d'une variable numérique
void setup() {
  pinMode(Led, OUTPUT); //Déclaration de Led en tant que sortie
  pinMode(capteur, INPUT); // Déclaration de capteur en tant qu'entrée
}
void loop() {
  val = digitalRead(capteur); // Lecture de l'état du capteur
  if (val == HIGH) {
    digitalWrite(Led, HIGH); // Allume la led 13
  }
  else {
    digitalWrite(Led, LOW); // Eteint la led 13
  }
}
```

37. Module détecteur de chocs et de vibrations.

Ce module didactique est basé sur un capteur de chocs à ressort. La sortie digitale commute lors d'un choc suffisamment important.



Caractéristiques :

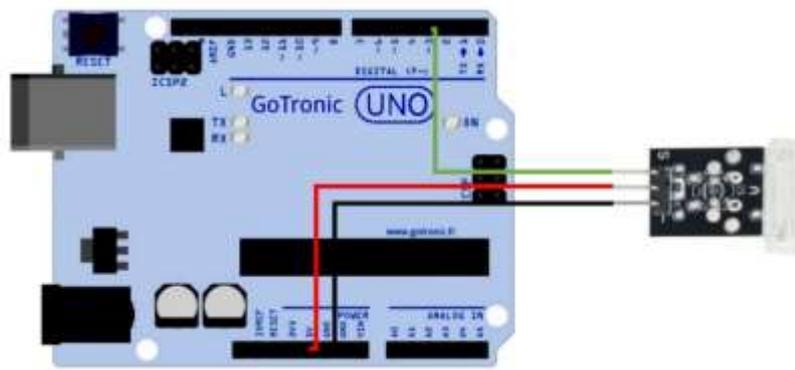
Alimentation : 5 Vcc

Sorties : Signal, Vcc, GND

Dimensions : 28 x 21 x 10

Câblage :

Carte Uno	Module détecteur de choc
5 V	+
GND	G
3	S



Exemple de programme :

```
int Led = 13; // Déclaration de la Led sur la broche 13
int capteur = 3; // Déclaration du capteur sur la broche 3
int val; // Déclaration d'une variable numérique
void setup()
{
  pinMode(Led, OUTPUT); //Déclaration de Led en tant que sortie
  pinMode(capteur, INPUT); // Déclaration de capteur en tant qu'entrée
}
void loop()
{
  val = digitalRead(capteur); // Lecture de l'état du capteur
  if (val == LOW)
  {
    digitalWrite(Led, HIGH);
  }
  else
  {
    digitalWrite(Led, LOW);
  }
}
```



Si vous rencontrez des problèmes, merci de nous contacter par courriel à :

sav@gotronic.fr