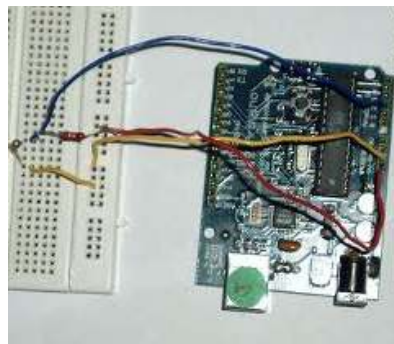
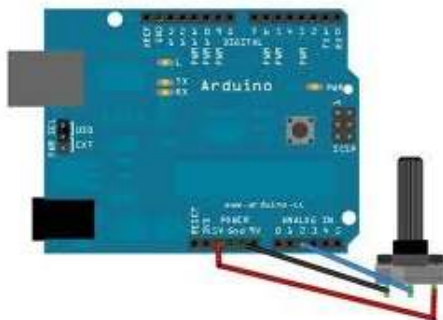




Arduino en technologie



Quatrième / Troisième





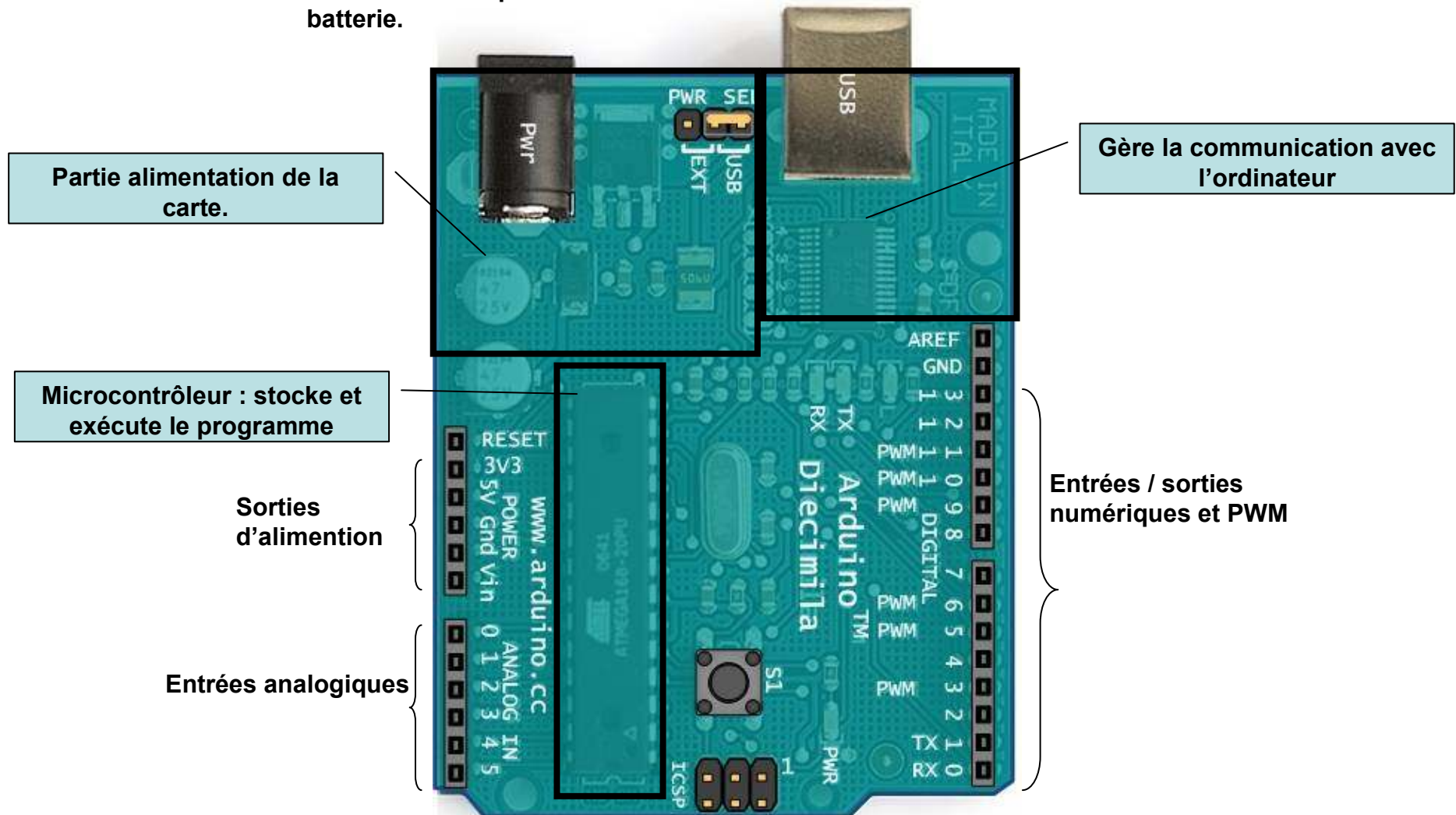
Arduino, c'est quoi ?

- Une carte d'interface programmable capable de piloter des capteurs et des actionneurs afin de simuler ou créer des systèmes automatisés.
- Un logiciel de programmation qui permet de programmer la carte en fonction du comportement désiré.
- Le tout, logiciel comme matériel, est sous licence libre. Une grande communauté d'amateurs et de passionnés contribuent à développer des applications et à les partager.

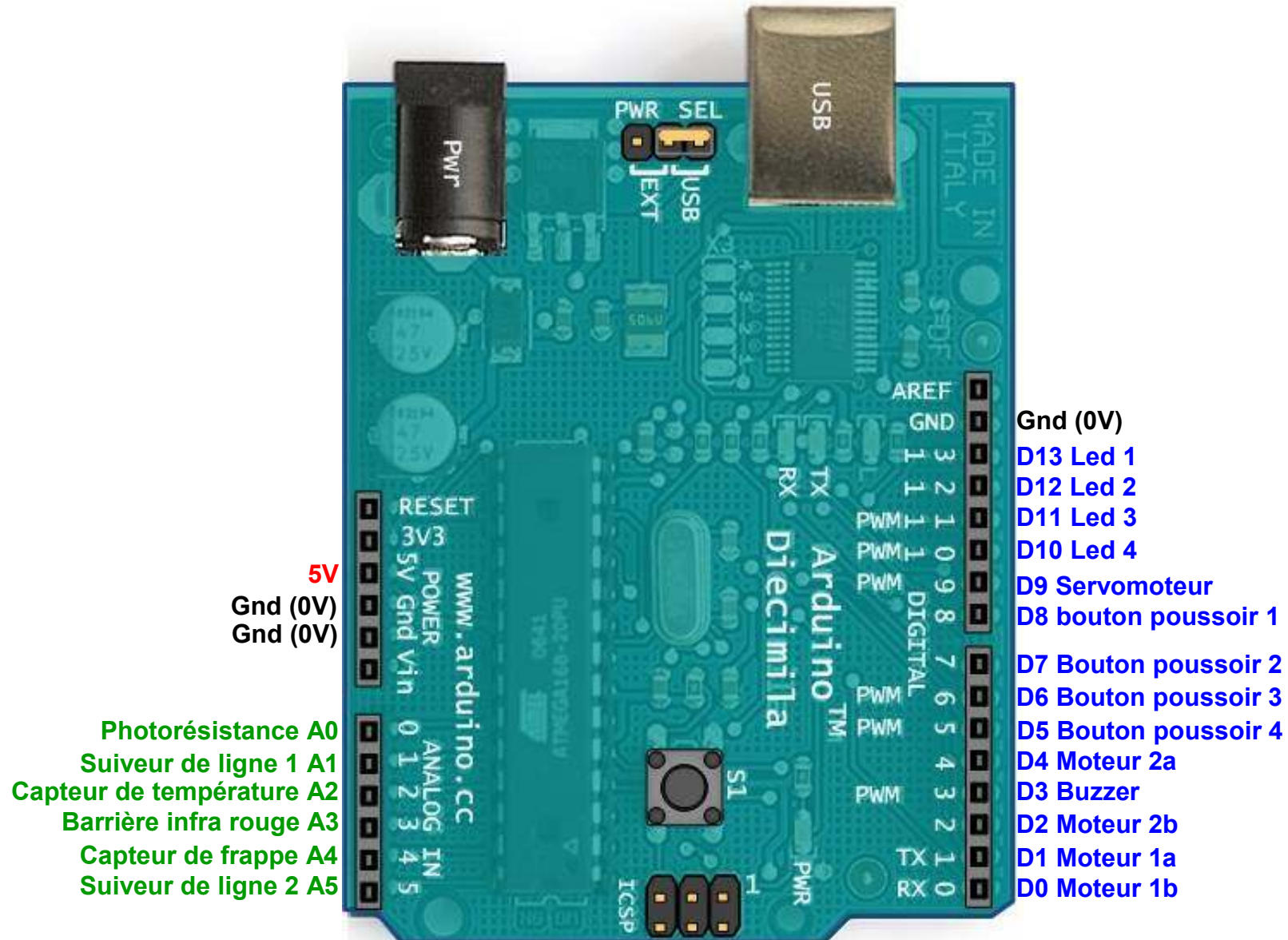
Présentation de la carte

Connecteur jack, permet de rendre autonome la carte en y connectant une pile ou une batterie.

Port USB, permet de communiquer avec la carte et de l'alimenter en 5V.



Où connecter les modules ?





Programmer avec le logiciel arduino

Le programme s'écrit à l'aide des fonctions décrites dans les pages suivantes. Chaque module électronique utilisé est connecté à la carte. Le programme est envoyé dans la carte qui gèrera les modules comme cela a été programmé.

Les fonctions s'écrivent entre les deux accolades de void loop() du programme de base.

```
void loop()
{
  lirecapteur(ir);
  if(barriere<900)
  {allumer(led1);}
  else
  {eteindre(led1);}
}
```

On programme ici

Charger un programme dans la carte

- Etape 1 : lancer le logiciel.



- Etape 2 : ouvrir et modifier le programme.

- Etape 3 : vérifier le programme.



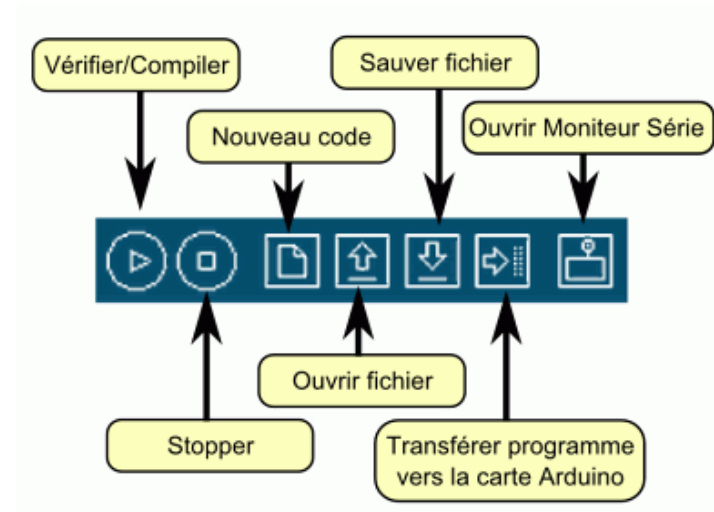
- Etape 4 : connecter la carte à l'ordinateur avec le cordon USB.

- Etape 5 : transférer le programme vers la carte.



Si un message d'erreur apparaît : faire Outil/Port série/Com ... et choisir le bon port USB.

- Etape 6 : vérifier le fonctionnement.

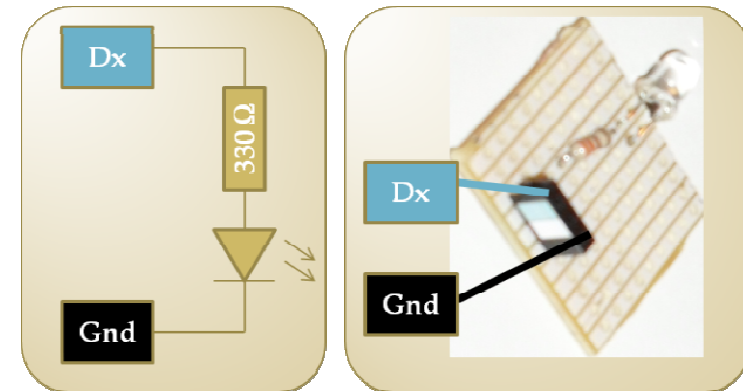


Module LED ou DEL

*La LED ou DEL est un actionneur.
Lorsqu'elle est parcourue par un courant électrique, elle émet de la lumière.*

Nom à utiliser dans le programme	Connecté à la broche
led1	D13
led2	D12
ledvar1	D11
ledvar2	D10

Fonctions	Commentaires
<code>allumer(<i>nom</i>);</code>	Allume les LED <i>led1</i> ou <i>led2</i>
<code>eteindre(<i>nom</i>);</code>	Eteint les LED <i>led1</i> ou <i>led2</i>
<code>varier(<i>nom</i>);</code>	Fait varier la luminosité des LED connectées à <i>ledvar1</i> et <i>ledvar2</i> .
<code>pendant(<i>durée</i>);</code>	<i>durée</i> est une valeur en millisecondes. 1s = 1000 ms



Exemple : faire clignoter une LED connectée à D13

```
void loop()
{
  allumer(led1);
  pendant(1000);
  eteindre(led1);
  pendant(1000);
}
```

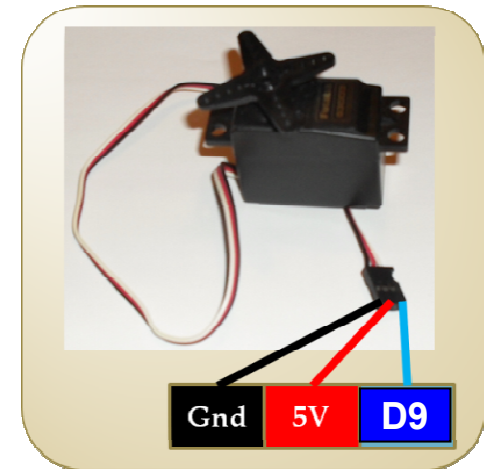
Servomoteur

Un servomoteur est un moteur dont on peut commander la position angulaire, de 0° à 180° avec précision.

Le fil jaune ou blanc se connecte à D9.

Fils	Connecté à la broche
Fil noir	Gnd
Fil rouge	5V
Fil jaune ou blanc	D9

Fonctions	Commentaires
angleservo(<i>angle</i>);	<i>angle</i> est compris entre 0 et 180. Il correspond à l'angle auquel le servomoteur se déplacera.



Exemple : faire tourner le servomoteur de 0° à 45° par pas de 5°

```
void loop()
{
  angleservo(0);
  for(int i=0; i<=45; i+=5)
  {
    angleservo(i);
    pendant(500) ;
  }
}
```

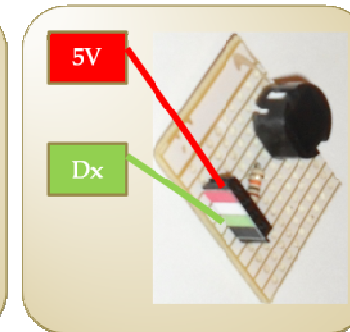
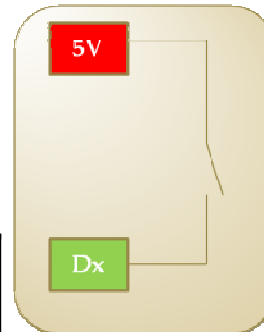

Bouton poussoir, interrupteur et micro rupteur

Le bouton poussoir est un interrupteur qui ferme le circuit lorsqu'on appuie dessus et l'ouvre lorsqu'on relâche.

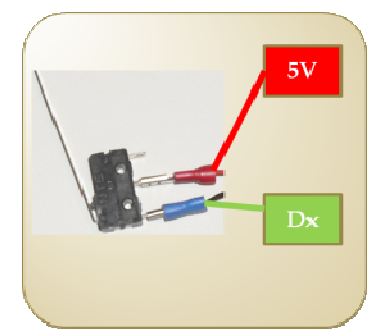
Le micro rupteur est un bouton poussoir auquel est ajoutée une languette pour détecter un contact.

Nom à utiliser dans le programme	Connecté à la broche
bp1	D8
bp2	D7
bp3	D6
bp4	D5

Fonctions	Commentaires
<code>lire(nom);</code>	Permet de lire la valeur d'un bouton poussoir.
<pre> if(etatnom==HIGH) { ACTION DESIREE } else { ACTION DESIREE } </pre>	<p>Déclenche une action si le bouton poussoir est appuyé. Sinon, déclenche une autre action désirée.</p> <p>HIGH : il est appuyé LOW : il est relâché</p>



Bouton poussoir



Micro rupteur

Exemple : fait tourner le moteur dans un sens si on appuie sur le bouton poussoir connecté à D8

```

void loop()
{
lire(bp1);
if (etatbp1==HIGH)
{ avancer(); }
else
{ arret(); }
}

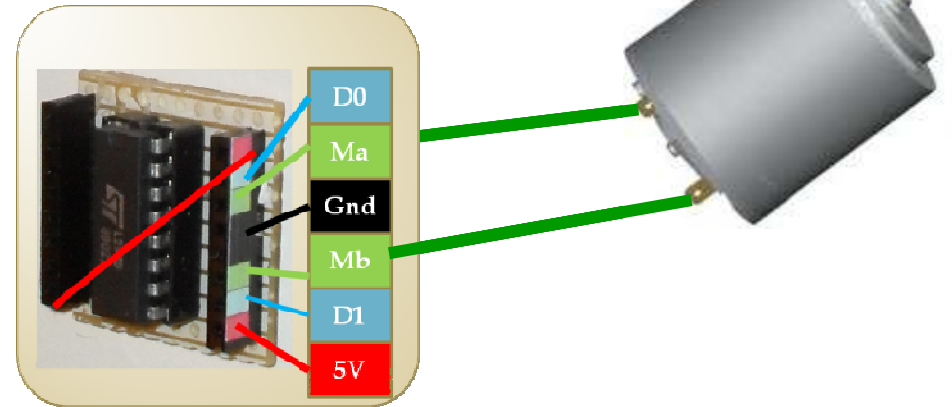
```

Moteur à courant continu

Le module de commande de moteurs à courant continue est une interface de puissance qui permet de piloter deux moteurs. On peut les faire tourner dans les deux sens et les arrêter.

Broches des moteurs	Connecté à la broche
Moteur 1 a	D1
Moteur 1 b	D0
Moteur 2 a	D4
Moteur 2 b	D2

Fonctions	Commentaires
avancer();	Fait tourner les moteurs dans un sens
reculer();	Fait tourner les moteurs dans l'autre sens
arret();	Arrête les moteurs
pendant(<i>durée</i>);	<i>durée</i> est une valeur en millisecondes. 1s = 1000 ms

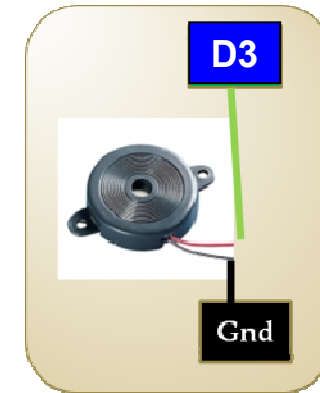
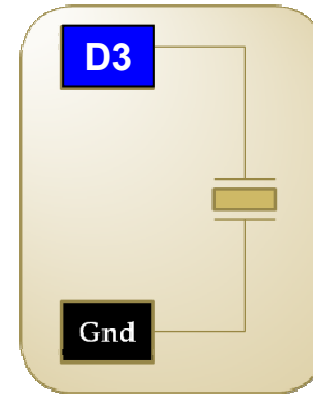


Exemple : si le bouton poussoir est appuyé, les moteurs

```
void loop()
{
  lire(bp1);
  if (etatbp1==HIGH)
  {
    reculer(); pendant(2000);
    arret();
  }
  else
  {
    avancer(); pendant(2000);
    arret();
  }
}
```

Buzzer

Le buzzer vibre en émettant un son à la fréquence du signal électrique qui le parcourt.



Fils	Connecté à la broche
Fil noir	Gnd
Fil rouge	D3
Fil jaune ou blanc	D9

Fonctions	Commentaires
melodie();	<i>Joue une mélodie préprogrammée.</i>

Exemple : jouer une mélodie préprogrammée et attendre 5 secondes.

```
void loop()
{
  melodie();
  pendant(5000);
}
```

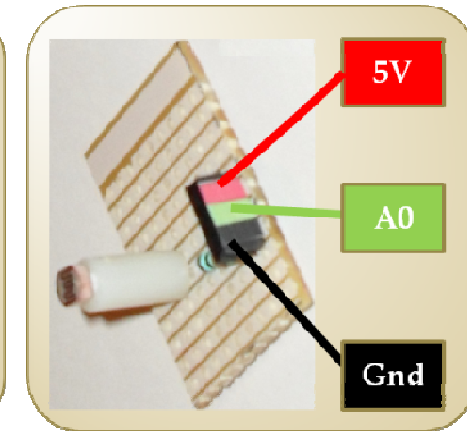
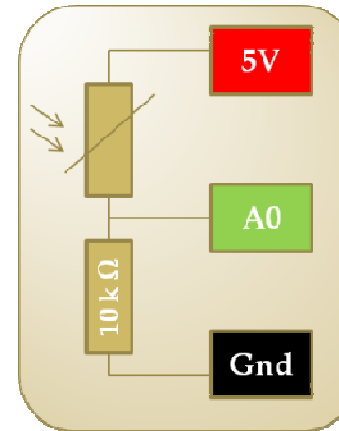
Photorésistance (LDR)

La photorésistance est une résistance qui varie en fonction de la luminosité.

La carte renvoie un nombre compris entre 0 (absence de lumière) et 1023 (luminosité maximum).

Nom à utiliser dans le programme	Connecté à la broche
ldr	A0
lumiere	-

Fonctions	Commentaires
lirecapteur(ldr);	Permet de lire une valeur de luminosité.
<pre> if(lumiere>=seuil) { ACTION DESIREE } else { ACTION DESIREE } </pre>	<p>Déclenche une action si la luminosité est supérieure ou inférieure au seuil. Sinon, déclenche une autre action désirée.</p> <p>seuil : nombre entre 0 (luminosité min.) et 1023 (luminosité max.)</p> <p>Comparaisons possibles : > ou < ou >= ou <=</p>



Exemple : allume la LED si la quantité de lumière captée est inférieure au seuil.

```

void loop()
{
lirecapteur(ldr) ;
if(lumiere<=500)
{allumer(led1)}
else
{eteindre(led1)}
}

```

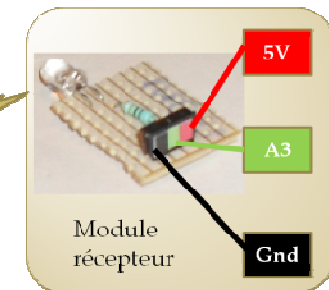
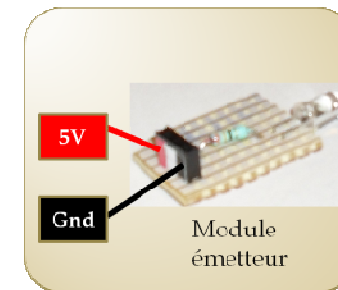
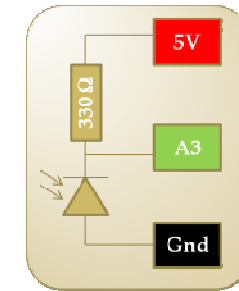
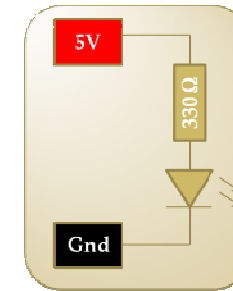
Barrière infrarouge

Le module barrière infrarouge connecté à l'entrée analogique A3 permet de détecter le passage d'un objet entre la LED infrarouge émettrice et le phototransistor infrarouge récepteur.

Le nombre récupéré par l'interface arduino est compris entre 0 et 1023. 1023 est le nombre renvoyé si le faisceau infrarouge est coupé.

Nom à utiliser dans le programme	Connecté à la broche
ir	A3
barriere	-

Fonctions	Commentaires
lirecapteur(<i>irr</i>);	Permet de lire une valeur de luminosité.
<pre> If(barriere>=seuil) { ACTION DESIREE } else { ACTION DESIREE } </pre>	<p>Déclenche une action si la si le faisceau lumineux est coupé. Sinon, déclenche une autre action désirée.</p> <p>seuil : nombre entre 0 (luminosité min.) et 1023 (luminosité max.)</p> <p>Comparaisons possibles : > ou < ou >= ou <=</p>



Exemple : allume la LED si la quantité de lumière captée est inférieure au seuil.

```

void loop()
{
lirecapteur(ir) ;
if(barriere<=900)
{allumer(led1)}
else
{eteindre(led1)}
}

```

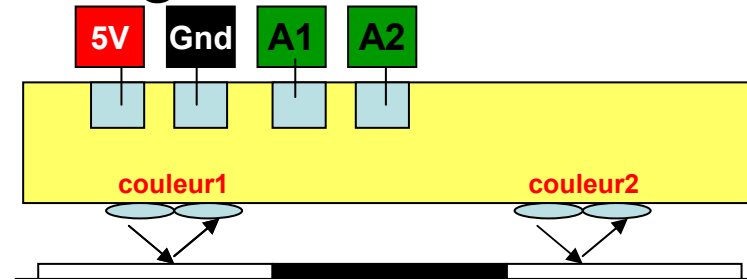


Suiveur de ligne

Le suiveur de ligne fonctionne sur le principe de la barrière infrarouge. La lumière renvoyée par le blanc est supérieure à celle renvoyée par le noir.

0 -> noir

1 -> blanc



Nom à utiliser dans le programme	Connecté à la broche
ligne1	A1
ligne2	A5
couleur1	Stocke le chiffre de la couleur1
couleur2	Stocke le chiffre de la couleur2

Fonctions	Commentaires
lirecapteur(<i>nom</i>);	Permet de lire une valeur de quantité de lumière réfléchie.
<pre> if(couleur1==0) { ACTION DESIREE } else { ACTION DESIREE } </pre>	Déclenche une action si la en fonction des couleurs détectées. Couleur blanche : 0 Couleur noire : 1

Exemple : allume la LED si la quantité de lumière captée est inférieure au seuil.

```

void loop()
{
lirecapteur(ligne1) ;
lirecapteur(ligne2) ;

if(couleur1==0 && couleur2==0)
  {avancer();pendant(1000)}
elseif(couleur1==1 && couleur2==0)
  {gauche();pendant(1000);}
elseif(couleur1==0 && couleur2==1)
  {droite(); pendant(1000);}
elseif(couleur1==1 && couleur2==1)
  {arret();}

}

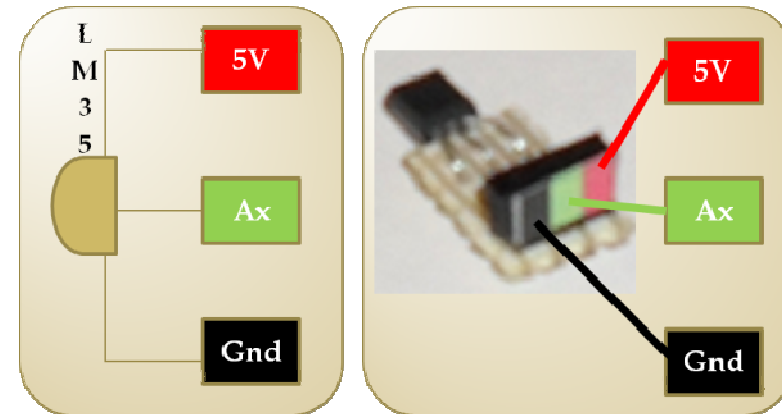
```

Capteur de température

Le capteur de température permet à la carte de calculer la température à proximité de ce dernier.

Nom à utiliser dans le programme	Connecté à la broche
-	A2
temperature	-

Fonctions	Commentaires
liretemp();	Permet de lire une valeur de luminosité.
<pre>if(temperature>=seuil) { ACTION DESIREE } else { ATION DESIREE }</pre>	<p>Déclenche une action si la température est supérieure ou inférieure au seuil. Sinon, déclenche une autre action désirée.</p> <p>seuil : nombre entre 0 et 50 en degré Celsius.</p> <p>Comparaisons possibles : > ou < ou >= ou <=</p>



Exemple : allume la LED si la quantité de lumière captée est inférieure au seuil.

```
void loop()
{
  liretemp() ;
  if(temperature<=28)
    {sens1() ;}
  else
    {arret() ;}
}
```

Capteur de frappe

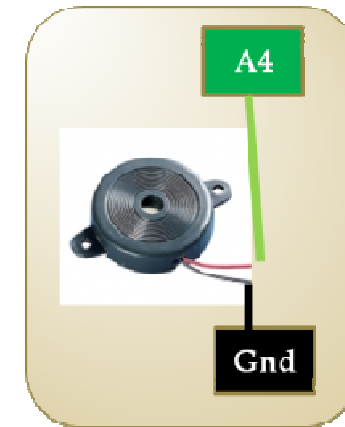
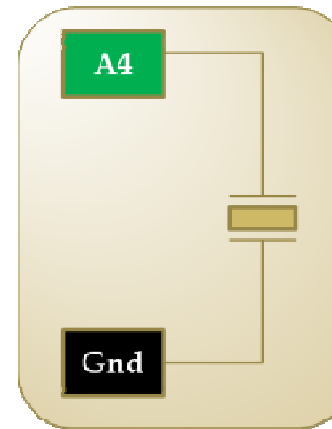
Le module capteur de frappe connecté à l'entrée analogique A4 permet de détecter une frappe sur le buzzer ou sur son support.

Le nombre récupéré par l'interface arduino est compris entre 0 et 1023.

Un seuil de détection de 100 suffit à détecter une petite frappe.

Nom à utiliser dans le programme	Connecté à la broche
frappe	A4
intensite	-

Fonctions	Commentaires
lirecapteur(frappe);	Permet de lire une valeur de luminosité.
<pre> If(intensite>=seuil) { ACTION DESIREE } else { ATION DESIREE } </pre>	<p>Déclenche une action si la frappe est supérieure ou inférieure au seuil. Sinon, déclenche une autre action désirée.</p> <p>seuil : nombre entre 0 (rien) et 1023 (frappe max.)</p> <p>Comparaisons possibles : > ou < ou >= ou <=</p>



Exemple : allume la LED si la frappe captée est supérieure au seuil.

```

void loop()
{
  lirecapteur(frappe) ;
  if(intensite>=100)
    {allumer(led1)}
  else
    {eteindre(led1)}
}

```


Les tests dans le programme

<i>Test</i>	<i>Dans le programme</i>
Si ...	<code>if(condition) {instructions}</code>
Si ... Sinon	<code>if(condition) {instructions} else {instructions}</code>
Si ... Sinon Si ...	<code>if(condition) {instructions} elseif(condition) {instructions}</code>
Faire ... Tant que...	<code>do {instructions} while(condition) ;</code>
Tan que ...	<code>while(condition) {instructions}</code>
Compteur	<code>for(variable ; condition ; comptage) {instructions}</code> <p><i>Exemples : exécute les instructions 5 fois.</i></p> <code>for(variable=0 ; variable == 4 ; variable+=1) {instruction}</code> <code>for(int i=0 ; i==4) {i++ instructions}</code>

Les conditions

<i>Condition</i>	<i>Dans le programme</i>
Egale à	==
Supérieur	>
Inférieur	<
Supérieur ou égal	>=
Inférieur ou égal	<=
Non égal	!=

Opérateurs logiques

<i>Opérateur logique</i>	<i>Dans le programme</i>	<i>Exemple</i>
ET logique Vraie si les deux conditions sont remplies	&&	if(condition 1 && conditions 2)
OU logique Vrai si l'une ou l'autre des conditions est remplie	!!	if(condition 1 !! conditions 2)
NON logique Vrai si la condition est fausse	!	if(! condition)

Exemple de câblage

