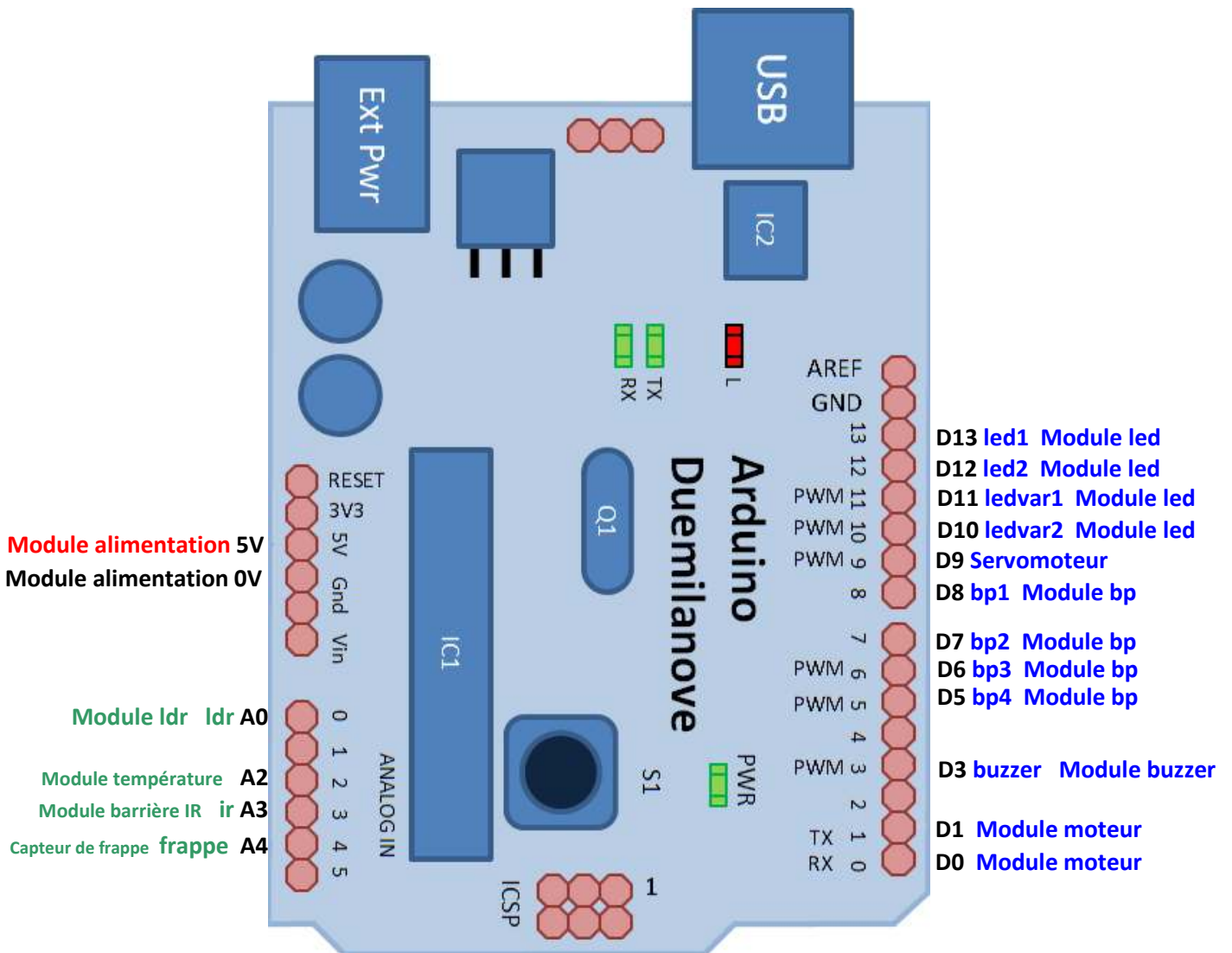


# Arduino, modules et fonctions

## Sommaire :

Câblage des modules .....	Page 2
Module led .....	Page 3
Module bouton poussoir .....	Page 4
Module servomoteur .....	Page 5
Module moteur .....	Page 6
Module ldr (photorésistance) .....	Page 7
Module température .....	Page 8
Module barrière infrarouge .....	Page 9
Module frappe .....	Page 10
Module buzzer .....	Page 11
Programme de base V2 .....	Page 12
Exemple d'application .....	Page 13
Tests, conditions et opérateurs logiques .....	Page 14

## Branchement des modules



Entrées analogiques	Modules
A0	Module ldr
A1	
A2	Module température
A3	Module barrière IR
A4	Module capteur de frappe
A5	

	Entrées
	Sorties

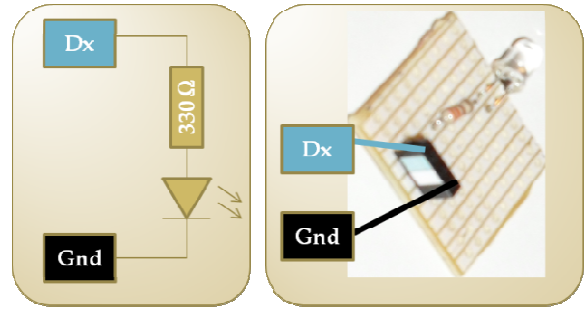
Entrées/Sorties numériques	Nom	Modules
D13	led1	Module led
D12	led2	Module led
D11	ledvar1	Module led
D10	ledvar2	Module led
D9	-	Servomoteur
D8	bp1	Module bp
D7	bp2	Module bp
D6	bp3	Module bp
D5	bp4	Module bp
D3	Buzzer	Module buzzer
D1	-	Module moteur
D0	-	Module moteur

Cette configuration permet l'utilisation de sous programmes en langue française et faisant toujours référence aux mêmes modules, entrées et sorties.

Tout reste configurable pour une application donnée.

## Module led

L'entrée des modules led se connecte soit à D13 et D12 pour allumer ou éteindre la led, soit à D11 ou D10 pour faire varier la luminosité de la led.



Connexion à la broche	Nom à utiliser dans le programme principal	Fonctions à utiliser dans le programme principal	Algorithme correspondant
D13	led1	allumer( <b>nom</b> ) ; eteindre( <b>nom</b> ) ; attendre( <b>temps</b> ) ;	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Allumer <b>nom</b></div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Eteindre <b>nom</b></div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Attendre <b>temps</b></div>
D12	led2	clignoter( <b>nom</b> , <b>temps</b> ) ; commentaire : les <b>temps</b> sont en millisecondes (1000 pour 1s)	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Clignoter <b>nom</b> pendant <b>temps</b></div>
D11	ledvar1	varierled( <b>nom</b> , <b>valeur</b> ) ;	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Faire varier <b>nom</b> à la <b>valeur</b></div>
D10	ledvar2	commentaire : <b>valeur</b> est un nombre compris entre 0 et 255	

## Exemple d'utilisation

Module led connecté sur D13

Le programme fait clignoter une led : 1 seconde allumée et une seconde éteinte.

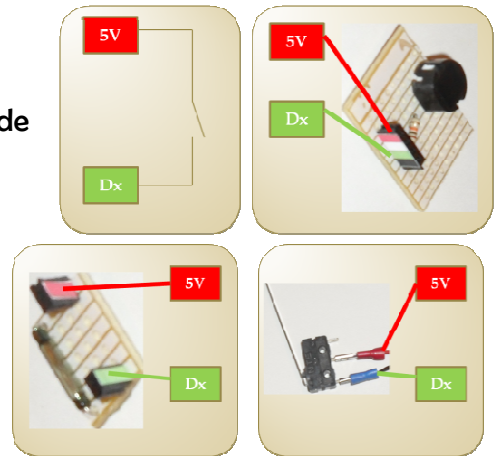
<pre> graph TD     Start([Début]) --&gt; On[Allumer led1]     On --&gt; Wait1[Attendre 1 seconde]     Wait1 --&gt; Off[Eteindre led1]     Off --&gt; Wait2[Attendre 1 seconde]     Wait2 --&gt; On     </pre>	<pre> // Boucle du programme principal// void loop() {   allumer(led1);   attendre(1000);   eteindre(led1);   attendre(1000) ; } </pre>
---	---

## Module bp

Les modules bouton poussoir, ILS et micro rupteur de connectent sur D8, D7, D6, ou D5.

L'interface arduino lit soit HIGH lorsque l'interrupteur est fermé, soit LOW lorsque l'interrupteur est ouvert.

Attention : les ILS sont normalement fermés et s'ouvrent en présence d'un aimant.

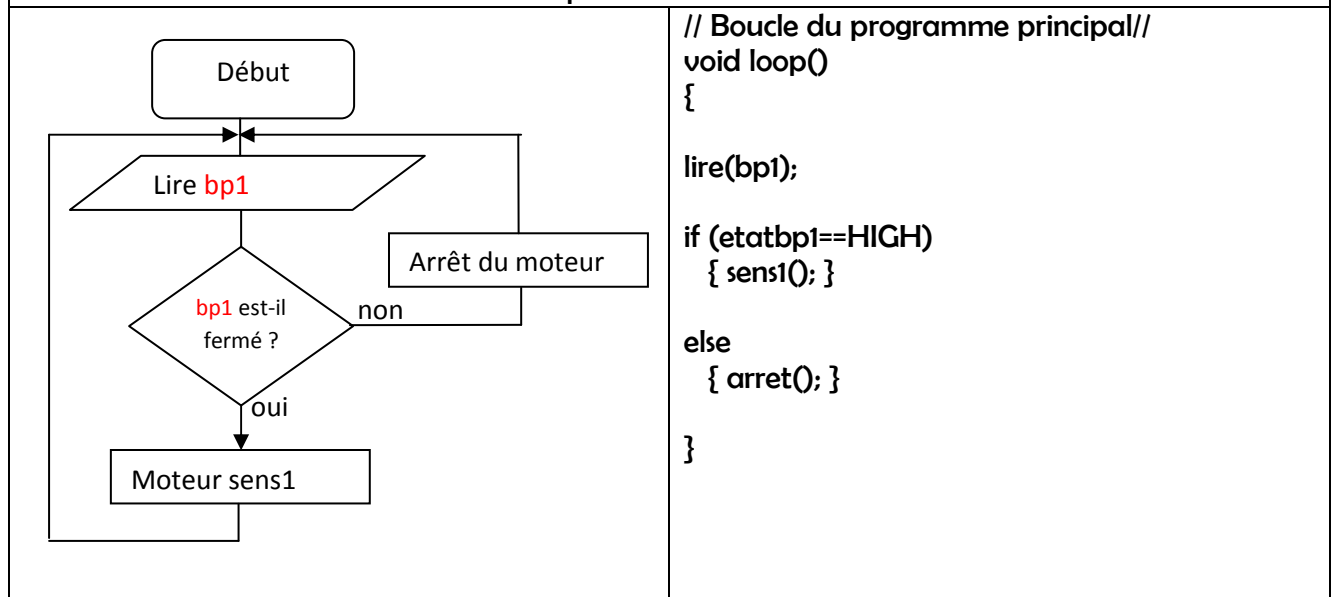


connexion à la broche	Nom à utiliser dans le programme principal	Fonctions à utiliser dans le programme principal	Algorithme correspondant
D8	bp1 etatbp1	<code>lire(nom) ;</code> <code>if (etatnom == HIGH)</code> <code>{instructions 1}</code> <code>else</code> <code>{instructions 2}</code>	
D7	bp2 etatbp2		
D6	bp3 etatbp3		
D5	bp4 etatbp4		

## Exemple d'utilisation

Module bp sur D8, Module moteur sur D1 et D0.

Allume le moteur dans le sens 1 si l'interrupteur est fermé.

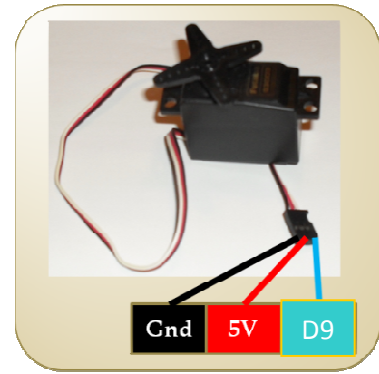


## Servomoteur

Un servomoteur se déplace entre 0° et 180°.

Son déplacement est précis.

Le fil blanc ou jaune est connecté à D9.

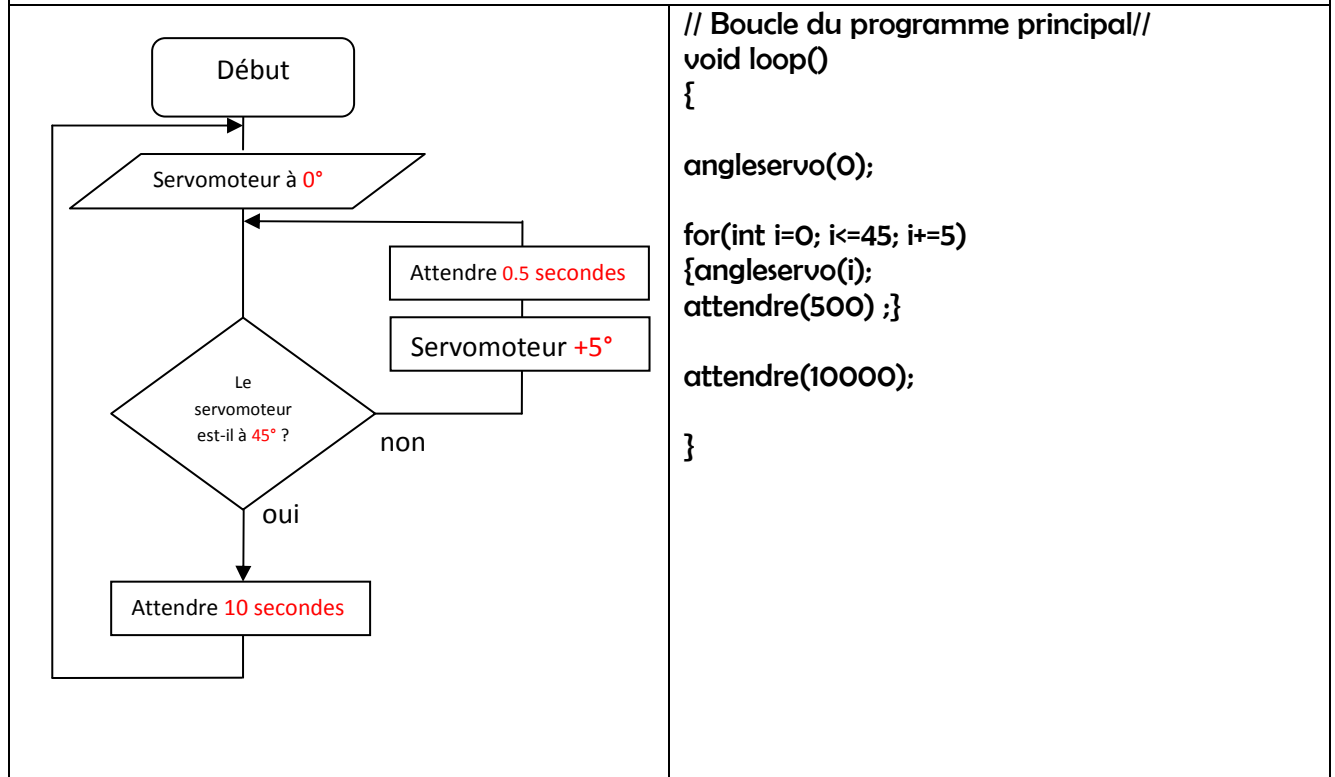


connexion à la broche	Nom à utiliser dans le programme principal	Fonctions à utiliser dans le programme principal	Algorithme correspondant
D9	-	angleservo( <b>angle</b> ) ; commentaire : angle est compris entre 0 et 180°	

## Exemple d'utilisation

Fil blanc ou jaune du servomoteur sur D9

Servomoteur à 0°, augmente de 5° tant que le servomoteur n'est pas 45°. Puis attend 10 secondes.

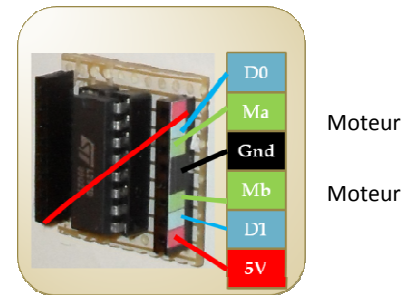


## Module moteur

Le module moteur permet de commander un moteur à courant continu.

La commande se fait via D1 et D0.

Le moteur est connecté à Ma et Mb du module.



connexion à la broche	Nom à utiliser dans le programme principal	Fonctions à utiliser dans le programme principal	Algorithme correspondant
D0	-	arret();	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">Arrêter le moteur</div> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-bottom: 5px;">Moteur dans le sens1</div> <div style="border: 1px solid black; padding: 5px; width: fit-content;">Moteur dans le sens2</div>
D1	-	sens1(); sens2();	

## Exemple d'utilisation

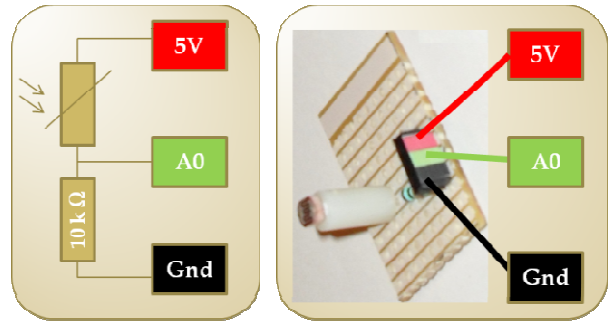
### Module moteur sur D1 et D0

<pre> graph TD     Start([Début]) --&gt; Stop[/Arrêter le moteur/]     Stop --&gt; Wait1[Attendre 1 secondes]     Wait1 --&gt; Sens1[/Moteur dans le sens1/]     Sens1 --&gt; Wait2[Attendre 1 secondes]     Wait2 --&gt; Stop         </pre>	<pre> // Boucle du programme principal// void loop() {   arret();    attendre(1000);    sens1();    attendre(1000); }         </pre>
---	--

## Module Idr (photorésistance)

Le module Idr connecté à l'entrée analogique A0 permet de mesurer la luminosité.

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



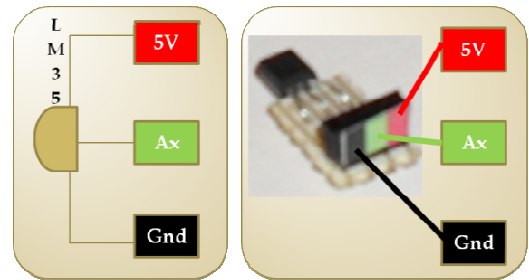
connexion à la broche	Nom à utiliser dans le programme principal	Fonctions à utiliser dans le programme principal	Algorithme correspondant
A0	ldr lumiere	lirecapteur(ldr) ;  if(lumiere<=500) {allumer(led1)} else {eteindre(led1)}	

## Exemple d'utilisation

Module LDR sur A0 et module LED sur D13	
	<pre>// Boucle du programme principal// void loop() { lirecapteur(ldr); if(lumiere&lt;500) {allumer(led1);} else {eteindre(led1);} }</pre>

## Module température

Ce module connecté sur A2 permet de mesurer la température ambiante.



Connexion à la broche	Nom à utiliser dans le programme principal	Fonctions à utiliser dans le programme principal	Algorithme correspondant
A2	temperature	<pre>liretemp() ;  if(temperature&lt;=28) {sens1() ;} else {arret() ;}  Commentaire : stocke la valeur de la température en degrés Celcius dans la variable « temperature »</pre>	

## Exemple d'utilisation

Module température sur A2 et module moteur sur DO et D1. Hélice sur le moteur

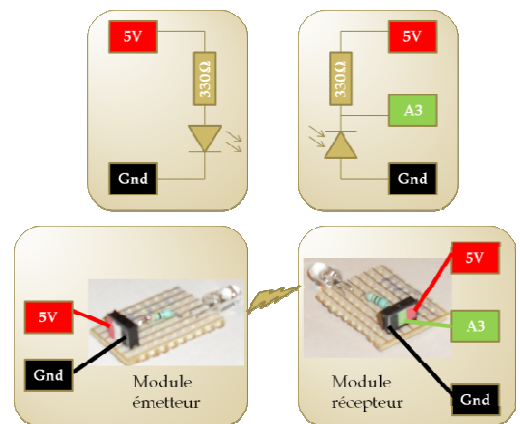
	<pre>// Boucle du programme principal// void loop() {   liretemp();    if(température&lt;28)   {sens1() ;}    else   {arret() ;}  }</pre>
--	---



## Module barrière IR (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.

connexion à la broche	Nom à utiliser dans le programme principal	Fonctions à utiliser dans le programme principal	Algorithme correspondant
A3	ir barriere	lirecapteur(ir) ;  if(barriere>=900) {allumer(led1)} else {eteindre(led1)}	

## Exemple d'utilisation

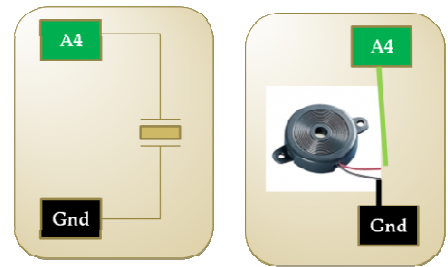
Module recpteur IR sur A3 et module LED sur D13	
	<pre>// Boucle du programme principal// void loop() { lirecapteur(ir); if(barriere&lt;900) {allumer(led1);} else {eteindre(led1);} }</pre>

## Module 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.



connexion à la broche	Nom à utiliser dans le programme principal	Fonctions à utiliser dans le programme principal	Algorithme correspondant
A4	frappe intensite	lirecapteur(frappe) ;  if(intensite>=100) {allumer(led1) attandre(500) ;} else {eteindre(led1)}	

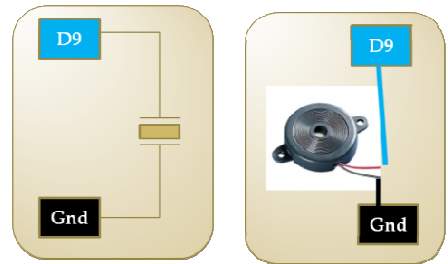
## Exemple d'utilisation

Module capteur de frappe sur A4 et module LED sur D13

	<pre>// Boucle du programme principal// void loop() {   lirecapteur(frappe) ;    if(intensite&gt;=100)   {allumer(led1) ;}   else   {eteindre(led1)}  }</pre>
--	---

## Module buzzer

On peut faire jouer une mélodie pré programmée par le buzzer.



connexion à la broche	Nom à utiliser dans le programme principal	Fonctions à utiliser dans le programme principal	Algorithme correspondant
D3	-	melodie();	Jouer la mélodie

## Exemple d'utilisation

Module buzzer sur D3	
<pre> graph TD     Start([Début]) --&gt; Play1[/Jouer mélodie/]     Play1 --&gt; Wait1[/Attendre 1 seconde/]     Wait1 --&gt; Play2[/Jouer mélodie/]     Play2 --&gt; Wait2[/Attendre 5 secondes/]     Wait2 --&gt; Play1         </pre>	<pre> // Boucle du programme principal// void loop() {     melodie();      attendre(1000);      melodie();      attendre(5000); }         </pre>

## Le programme de baseV2

```
//nom et câblage des entrées et des sorties//
#define led1 13 // module LED sur D13
#define led2 12 // module LED sur D12
#define ledvar1 11 // module LED sur D11
#define ledvar2 10 // module LED sur D10
#define buzzer 3 // module buzzer sur D3
#define bp1 8 // module bp, inter ou ils sur D8
#define bp2 7 // module bp, inter ou ils sur D7
#define bp3 6 // module bp, inter ou ils sur D6
#define bp4 5 // module bp, inter ou ils sur D5
#include <Servo.h> // autorise l'utilisation d'un
servomoteur
Servo servomoteur; // Servomoteur sur D9
#define Ma 1 // D1 sur Ma du module L293D
#define Mb 0 // D0 sur Mb du module L293D
#define ldr 0 // module LDR sur A0
#define temp 2 // module TEMP sur A2
#define ir 3 // module Barrière IR sur A3
#define frappe 4 // module capteur de frappe sur A4

int etatbp1; // variable qui enregistre la valeur du
module bp1
int etatbp2; // variable qui enregistre la valeur du
module bp2
int etatbp3; // variable qui enregistre la valeur du
module bp3
int etatbp4; // variable qui enregistre la valeur du
module bp4
int lumiere; // variable qui enregistre la valeur du
module LDR
int temperature; // variable qui enregistre la valeur du
module TEMPERATURE
int barriere; // variable qui enregistre la valeur du
module Barrière IR(infrarouge)
int intensite; // variable qui enregistre la valeur du
module capteur de frappe

//declaration des entrées et des sorties//
void setup()
{
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(ledvar1,OUTPUT);
  pinMode(ledvar2,OUTPUT);
  pinMode(bp1, HIGH);
  pinMode(bp2, HIGH);
  pinMode(bp3, HIGH);
  pinMode(bp4, HIGH);
  pinMode(buzzer, OUTPUT);
  servomoteur.attach(9);
  pinMode(Ma, OUTPUT);
  pinMode(Mb, OUTPUT);
}

//*****//

/** Boucle du programme principal **//

//*****//

void loop()

{

}

//*****//

//*****//

// fonctions à utiliser dans le programme principal//

void allumer(int quoi) // module LED sur D13
{digitalWrite(quoi,HIGH);}

void eteindre(int quoi) // module LED sur D10
{digitalWrite(quoi,LOW);}

void clignoter(int quoi, int millisecondes) // module LED sur D10
{digitalWrite(quoi,HIGH);
delay(millisecondes);
digitalWrite(quoi,LOW);
delay(millisecondes);}

void varierLed(int quoi,int valeur) // module LED sur D10
{analogWrite(quoi, valeur);}

void attendre(int millisecondes) // module LED sur D10
{delay(millisecondes);}

void sens1() // module LED sur D10
{ digitalWrite(Ma, HIGH);
digitalWrite(Mb,LOW);
delay(5);}

void sens2() // module LED sur D10
{ digitalWrite(Ma, LOW);
digitalWrite(Mb,HIGH);
delay(5);}

void angleServo(int angle) // module LED sur D10
{servomoteur.write(angle);
delay(15);}

void lire(int quoi) // module LED sur D10
{
  if(quoi==bp1){etatbp1=digitalRead(bp1);}
  else if(quoi==bp2){etatbp2=digitalRead(bp2);}
  else if(quoi==bp3){etatbp3=digitalRead(bp3);}
  else if(quoi==bp4){etatbp4=digitalRead(bp4);}
}

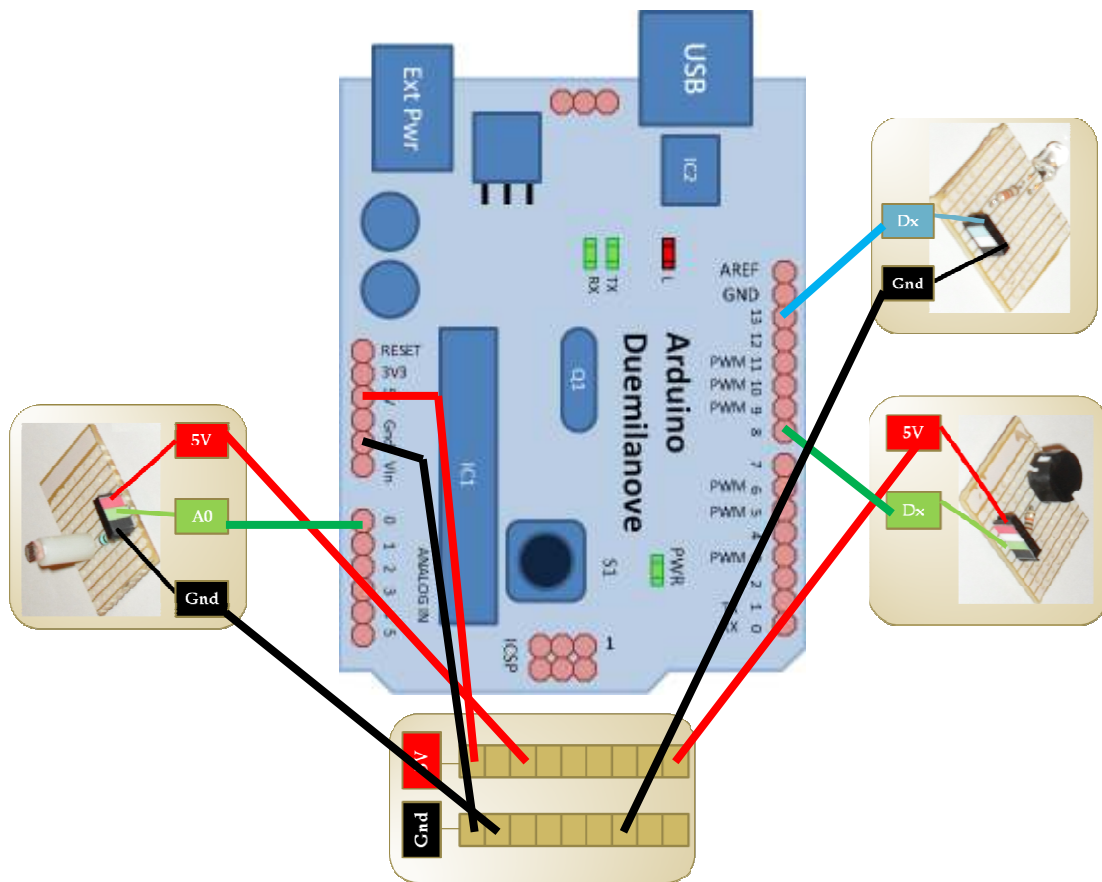
void lirecapteur(int quoi) // module LED sur D10
{
  if(quoi==ldr){lumiere=analogRead(ldr);}
  else if (quoi==ir){barriere=analogRead(ir);}
  else if (quoi==frappe){intensite=analogRead(frappe);}
}

void liretemp() // module LED sur D10
{
  int temps = 0; // variables pour
  stocker les valeurs de températures.
  int samples[8]; // variables pour calculer la température
  avec précision
  int maxi = -100,mini = 100; // valeurs initiales des températures
  min/max
  int i;
  //variable de comptage
  for(i = 0;i<=7;i++){ // enregistre 8 valeurs de températures et
  fait une moyenne
  samples[i] = ( 5.0 * analogRead(temp) *
  temps = temps + samples[i];
  delay(1000);}
  temps = temps/8.0; // calcule la température
  if(temps > maxi) {maxi = temps;} // set max temperature
  if(temps < mini) {mini = temps;} // set min temperature
  delay(1000); // attend 1s avant une nouvelle mesure
  temperature=temps; // enregistre la valeur de la température
}

void melodie()
{
  byte names[] = {'v', 'd', 'e', 'f', 'g', 'a', 'b', 'c'};
  int tones[] = {1915, 1700, 1519, 1432, 1275, 1136, 1014, 956};
  byte melody[] = "2d2a1f2c2d2a2d2c2f2d2a2c2d2a1f2c2d2a2g2p8p8p8";
  // count length: 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
  // 10 20 30
  int count = 0;
  int count2 = 0;
  int count3 = 0;
  int MAX_COUNT = 24;
  digitalWrite(buzzer, LOW);
  for (count = 0; count < MAX_COUNT; count++) {
    for (count3 = 0; count3 <=(melody[count*2] - 48) * 30; count3++) {
      for (count2=0;count2<8;count2++){
        if (names[count2] == melody[count*2 + 1]) {
          digitalWrite(buzzer,HIGH);
          delayMicroseconds(tones[count2]);
          digitalWrite(buzzer, LOW);
          delayMicroseconds(tones[count2]);}}}}
}

//*****//
//*****//
```

## Exemple de câblage



```
void loop
{
lire(bp1);
lirecapteur(ldr) ;

if(etatbp1==HIGH && lumiere<=500)
{ allumer(led1) ;
attendre(10000) ; }
else
{ eteindre(led1) ;}
}
```

## Les tests dans le programme

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

## 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)