

Arduino ressource Numéro 9



Sources

duinoedu.com





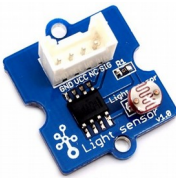


Ressource en ligne pour les algorithmes

<https://www.draw.io/>

Commande de l'allumage d'un couloir avec capteur de luminosité et ultrason

L'exercice, un couloir avec un interrupteur et une lampe, jusque là rien de bien original ! On rajoute un détecteur à ultrason pour allumer la lumière quand une personne passe dans le couloir et une photorésistance pour déclencher tout cela quand il fait bien sombre dans le fameux couloir. Pour compléter, on rajoute un afficheur LCD pour vérifier les valeurs de la luminosité et de la distance... au travail !

Matériel nécessaire pour réaliser cet exercice

Matériel	Matériel
Carte UNO + cordon USB 	Interrupteur avec connectique Grove 
LED avec connectique Grove <small>Idem mais avec LED rouge</small> 	Adaptateur Grove/Dupont 
Photorésistance 	Afficheur LCD 
Capteur de distance à ultrason 	

Première activité, nous allons commencer par une chose simple, quand j'appuie sur le bouton poussoir, la LED s'allume

```

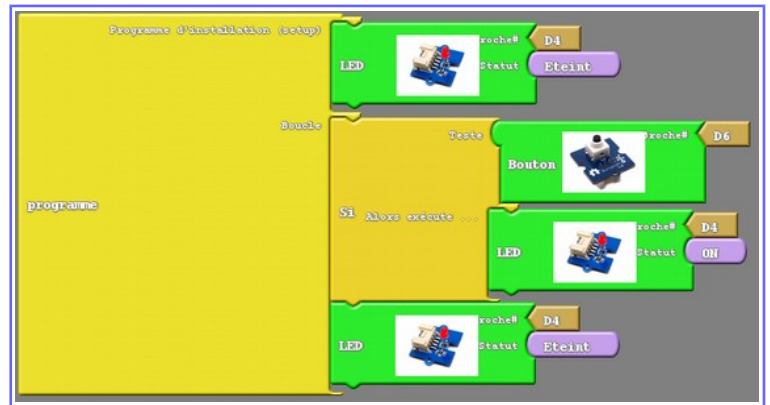
sketch_jun01a | Arduino 1.0.5-r2
Fichier Édition Croquis Outils Aide
Enregistrer

sketch_jun01a $
boolean __ardublockDigitalRead(int pinNumber)
{
  pinMode(pinNumber, INPUT);
  return digitalRead(pinNumber);
}

void setup()
{
  pinMode( 4 , OUTPUT);
  digitalWrite(4 , LOW);
}

void loop()
{
  if (__ardublockDigitalRead(6))
  {
    digitalWrite(4 , HIGH);
  }
  digitalWrite(4 , LOW);
}

Téléversement terminé
Taille binaire du croquis : 1 080 octets (d'un max de 32 256 octets)
25 Arduino Uno en COM5
  
```



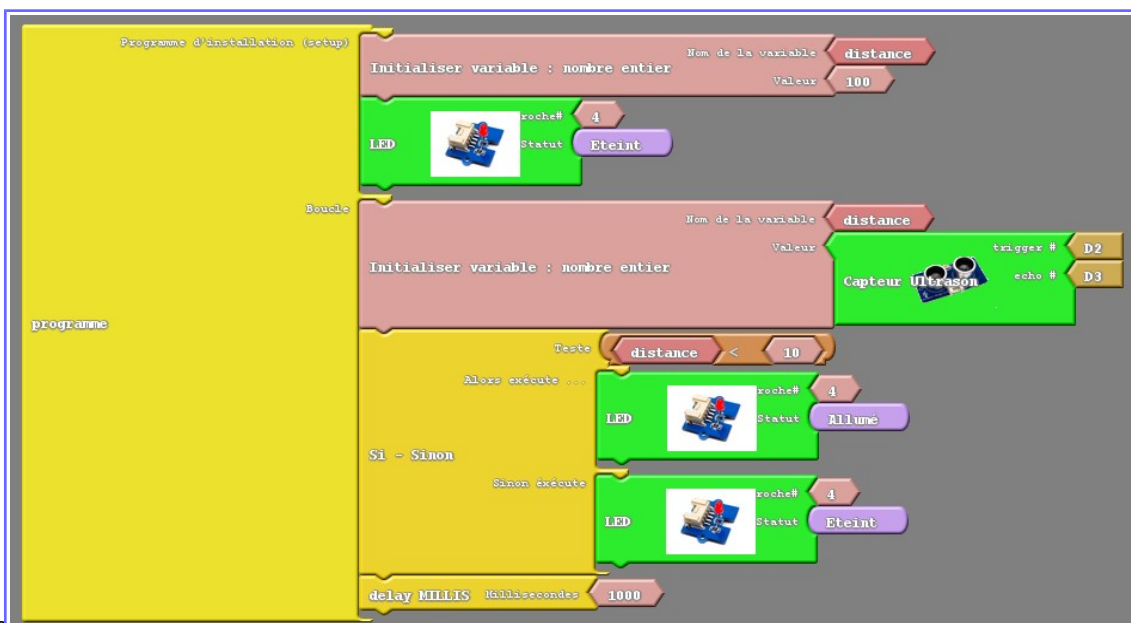
Maintenant, nous allons introduire le capteur à ultrason...

Le câblage est simple, même sans connecteur Grove, il suffit de bien repérer les sorties « Echo » et « Trigger » ici respectivement en D3 et D2

Nous allons initialiser une variable et lui affecter une valeur quelconque (ici 100)
 Tout comme la LED qui à l'initialisation est éteinte

On crée une boucle en prenant la valeur de la photorésistance pour la mettre dans la variable

Ensuite, on décide que si cette valeur de variable est inférieure à ici « 10 » la LED doit s'allumer et on effectue cette mesure toutes les 1 seconde...



```
sketch_jun01a | Arduino 1.0.5-r2
Fichier Édition Croquis Outils Aide
sketch_jun01a $
int _ABVAR_1_distance = 0 ;
int ardublockUltrasonicSensorCodeAutoGeneratedReturnCM(int trigPin
{
    long duration;
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(20);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    duration = duration / 59;
    if ((duration < 2) || (duration > 300)) return false;
    return duration;
}

void setup()
{
    pinMode( 4 , OUTPUT);
    digitalWrite( 2 , LOW );

    _ABVAR_1_distance = 100 ;

    digitalWrite(4 , LOW);
}

void loop()
{
    _ABVAR_1_distance = ardublockUltrasonicSensorCodeAutoGenerated
    if (( ( _ABVAR_1_distance ) < ( 10 ) ))
    {
        digitalWrite(4 , HIGH);
    }
    else
    {
        digitalWrite(4 , LOW);
    }
    delay( 1000 );
}
```

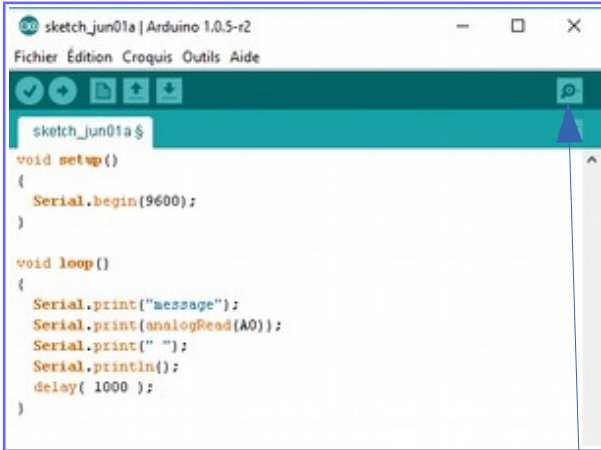
Vous constatez qu'en changeant la distance, la LED déclenche plus ou moins tôt...

Nous allons maintenant utiliser la photorésistance, pour connaître ses valeurs de déclenchement, (Arduino ressource numéro 5)

Attention, nous allons maintenant utiliser les entrées/sorties analogiques (A0 à A3), en effet le signal n'est 1 ou 0 mais une variation suivant l'exposition du capteur ! **Ici raccordement en A0**

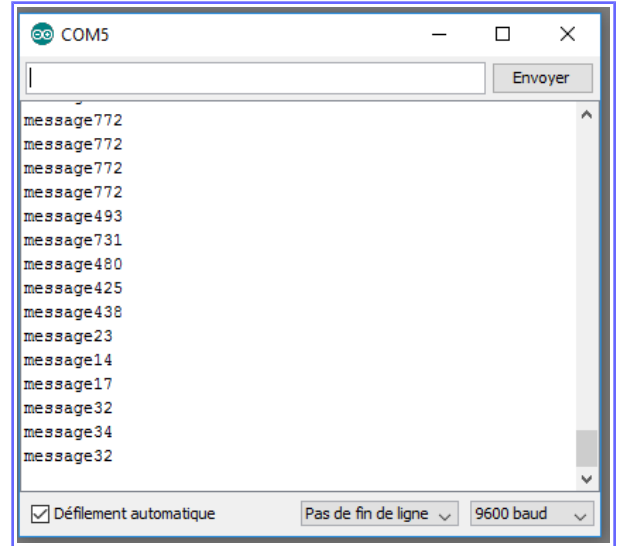
Pour pouvoir utiliser correctement la photorésistance, il faut tout d'abord pouvoir lire sa valeur. Voici ci-dessous le petit programme pour la connaître.

Il faut ensuite utiliser le port série du logiciel de programmation (IDE) pour visualiser cette valeur...



```
sketch_jun01a $
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.print("message");
  Serial.print(analogRead(A0));
  Serial.print(" ");
  Serial.println();
  delay( 1000 );
}
```

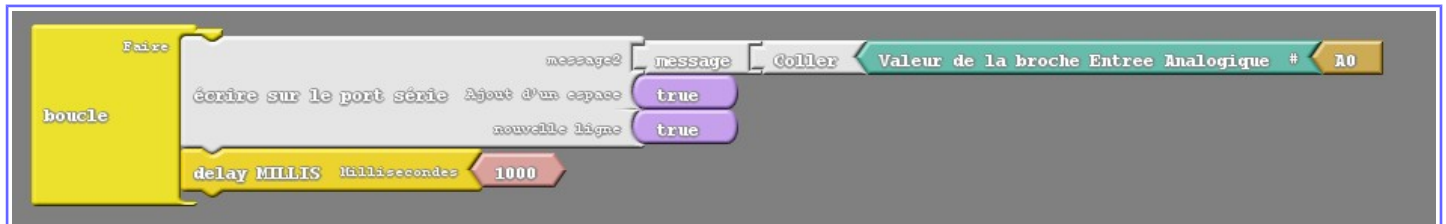


Vous constaterez que les valeurs en cliquant ici que les valeurs varient suivant que l'on cache ou non le récepteur de la photorésistance (ici les valeurs varient entre 772 et 32), nous allons donc faire déclencher l'allumage quand la valeur de la photorésistance sera inférieure à «400 » (c'est un exemple!)

Mesure à effectuer pour connaître les caractéristiques de votre photorésistance.

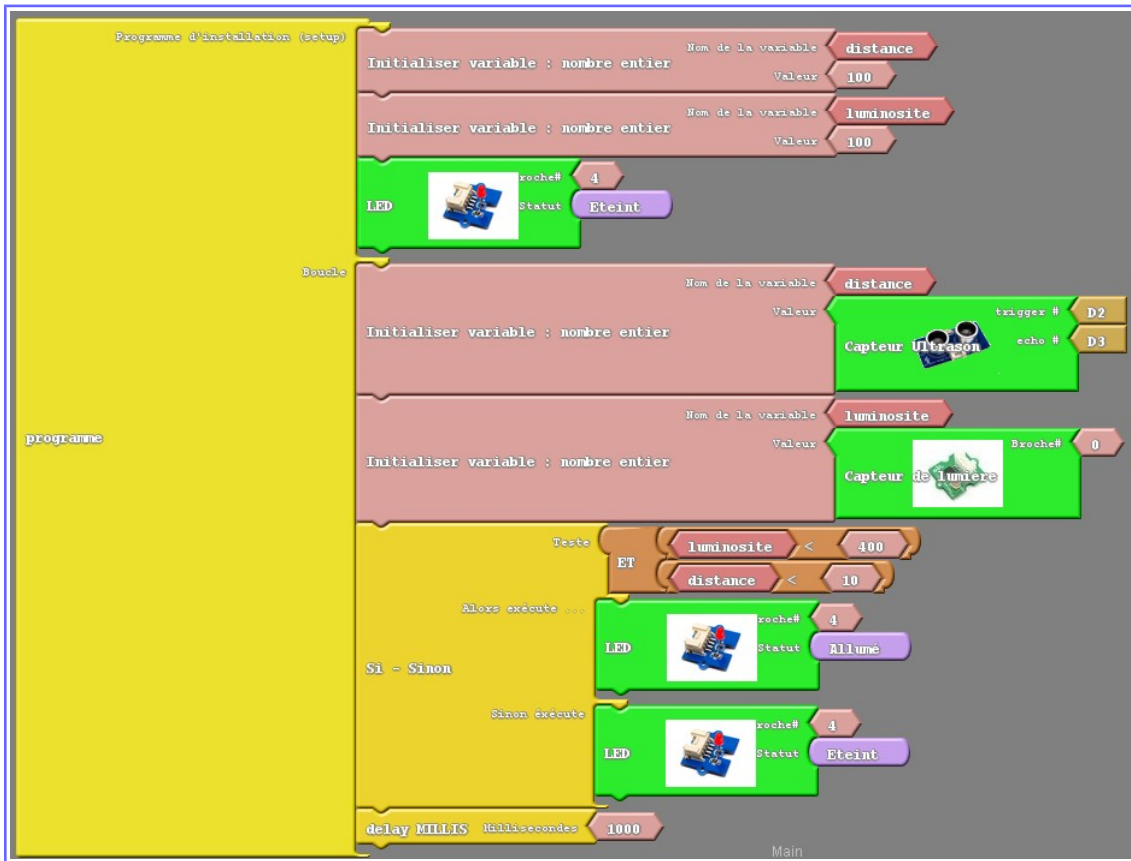
Voici les blocs utilisés pour cette mesure...

Vous constatez un écart avec la « Ressource Arduino numéro 5 » car depuis nous utilisons une nouvelle version d'Ardublock.



Pour cela, nous allons créer une deuxième variable « luminosité », variable qui sera comme pour la variable « distance » sera mise à jour en continu par la valeur de la photorésistance.

Maintenant pour l'allumage, il nous faut **deux** conditions, la distance (inférieur à 10) et la luminosité (inférieure à 400)



```

sketch_jun01a | Arduino 1.0.5-r2
Fichier Édition Croquis Outils Aide

sketch_jun01a $
int _ABVAR_1_distance = 0 ;
int _ABVAR_2_luminosite = 0 ;
int arduinoUltrasonicSensorCodeAutoGeneratedReturnCM(int trigPin
{
  long duration;
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(20);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  duration = duration / 59;
  if ((duration < 2) || (duration > 300)) return false;
  return duration;
}

void setup()
{
  pinMode( 4 , OUTPUT);
  digitalWrite( 2 , LOW );

  _ABVAR_1_distance = 100 ;
  _ABVAR_2_luminosite = 100 ;

  digitalWrite(4 , LOW);
}

void loop()
{
  _ABVAR_1_distance = arduinoUltrasonicSensorCodeAutoGeneratedReturnCM( 2 , 3 ) ;
  _ABVAR_2_luminosite = analogRead(0) ;
  if ( ( ( _ABVAR_2_luminosite ) < ( 400 ) ) && ( ( _ABVAR_1_distance ) < ( 10 ) ) )
  {
    digitalWrite(4 , HIGH);
  }
  else
  {
    digitalWrite(4 , LOW);
  }
  delay( 1000 );
}
Téléversement terminé
Taille binaire du croquis : 1 874 octets (d'un max de 32 256 octets)
  
```

La suite >>>>>>>>>>>>>>>>>>>>>>

Nous allons maintenant utiliser l'afficheur LCD (Ressource Arduino numéro 6)

Nous allons faire afficher (en rouge) sur la première ligne la distance (valeur extraite du capteur à ultrason) et la luminosité (valeur extraite de la photorésistance) mais attention, la photorésistance est raccordée sur un port série (ici A0)!

Nous allons compéter les blocs comme cela...

(nous reprenons à partir de la dernière ligne «delay1000»)



```
sketch_jun01a | Arduino 1.0.5-r2
Fichier Édition Croquis Outils Aide

sketch_jun01a $
#include <Wire.h>
#include <rgb_lcd.h>

int _ABVAR_1_distance = 0 ;
int _ABVAR_2_luminosite = 0 ;
int ardublockUltrasonicSensorCodeAutoGeneratedReturnCM(int trigPin, int echoPin)
{
  long duration;
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(20);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  duration = duration / 59;
  if ((duration < 2) || (duration > 300)) return false;
  return duration;
}

//libraries at http://duinoedu.com/dl/lib/grove/EDU LCD RGB Backlight Grove/
rgb_lcd monRgb;

void setup()
{
  pinMode( 4 , OUTPUT);
  digitalWrite( 2 , LOW );

  Serial.begin(9600);
  monRgb.branch();
  _ABVAR_1_distance = 100 ;

```

la suite

```
_ABVAR_2_luminosite = 100 ;

digitalWrite(4 , LOW);

}

void loop()
{
  _ABVAR_1_distance = ardublockUltrasonicSensorCodeAutoGeneratedReturnCM( 2 , 3 ) ;
  _ABVAR_2_luminosite = analogRead(0) ;
  if ( ( ( _ABVAR_2_luminosite ) < ( 400 ) ) && ( ( _ABVAR_1_distance ) < ( 10 ) ) )
  {
    digitalWrite(4 , HIGH);
  }
  else
  {
    digitalWrite(4 , LOW);
  }
  delay( 1000 );
  Serial.print("message");
  Serial.print(_ABVAR_2_luminosite);
  Serial.print(" ");
  Serial.println();
  monRgb.retroéclairage(1000,0,0);
  monRgb.placerCurseurEn(0,0);
  monRgb.ecrire("Distance" );
  monRgb.ecrire(_ABVAR_1_distance );
  monRgb.placerCurseurEn(1,0);
  monRgb.ecrire("Luminosite" );
  monRgb.ecrire(_ABVAR_2_luminosite );
}

```

Bien entendu, toutes les valeurs seront à reprendre avec votre propre maquette....