

/ THÉMATIQUE 6

Emission/détection de lumière - Interfaçage/mesure avec Arduino

OBJECTIF : fournir des moyens élémentaires de commande/contrôle de sources de lumière, de mesure d'intensité lumineuse, et d'automatisation de mesures/relevés expérimentaux, etc.

CONNAISSANCES REQUISES : aucune, mais être prêt à s'initier au langage Arduino, au câblage simple de composants électroniques, et à s'inspirer des innombrables exemples disponibles sur internet.

NOTIONS ABORDÉES : émission de lumière, détection de lumière, électronique, programmation, relevé expérimental, automatisation de mesures, ...

CONTENU DU KIT « ARDUINO » COMPLÉTANT LA LIGHTBOX :

- 1 module Arduino Uno
- 1 câble d'alimentation USB-USBC
- 1 plaquette test
- Fils de connexion (« jumpers ») :
>10 « mâle-mâle » + >5 « femelle-femelle »
- 5 résistances électriques de 330 Ω (Ohms)
- 3 diodes électroluminescentes (LED) rouges
- 3 LEDs vertes
- 1 LED ultraviolet (UV)
- 1 module LED RGB
- 1 module Laser rouge
- 1 module capteur à photorésistance
- 1 module détecteur de lumière Grove avec sortie VIS et VIS+IR

*Le maître mot, c'est : fouiller
les exemples et les tutoriels sur inter-
net, car c'est presque sûr que ce que l'on
cherche à faire est documenté quelque
part, et qu'un bon exemple fournira
la solution (ou presque) !*

Module Arduino Uno :

Le module Arduino Uno est une carte électronique qui embarque un microcontrôleur programmable, et des entrées/sorties électroniques capables d'interagir avec des composants électroniques ou des modules capteurs divers et variés.

Le module Arduino peut être alimenté par le câble USB fourni.

Dès qu'il est sous tension, le microcontrôleur exécute la série *d'instructions* (programme) qui lui a été transféré en mémoire. Ainsi, il peut exécuter des tâches en *autonomie*, qu'il soit alimenté par l'USB ou via une pile (c'est faisable, pour se passer d'une alimentation USB ou d'un ordinateur).

En revanche, il est nécessaire de raccorder l'Arduino à un ordinateur via le câble USB pour lui « charger » le programme que vous souhaitez lui faire exécuter. De même, pour récupérer et afficher des informations en provenance du module Arduino (et/ou des capteurs qui lui sont connectés), une solution simple est donc de laisser le module connecté à l'ordinateur pendant toute l'expérience. Les informations reçues via le port USB (communication de type port « Série ») pourront être affichées sur l'ordinateur.



Les connexions de la carte Arduino sont de différents types :

- **des bornes à tension constante :**
0V (masse=Ground=GND), +3.3 V et +5V
- **5 bornes d'entrées analogiques (Analog In A0 → A5) :**
permet à la carte de mesurer une tension analogique entre 0 et 5V.
- **14 bornes d'entrées ou sorties digitales (tout (5V) ou rien (0V)) :**
permet à la carte d'envoyer ou de détecter des signaux binaires 0/1. Parmi ces bornes, 6 bornes (3,5,6,9,10,11) peuvent être utilisées en sortie pour générer des signaux « mimant » une sortie analogique entre 0V et 5V en utilisant le principe de la modulation de largeur d'impulsion (PWM). Dans ce cas, le module Arduino génère des impulsions rapides entre 0 et +5V, en modulant le rapport cyclique entre 0% et 100%.

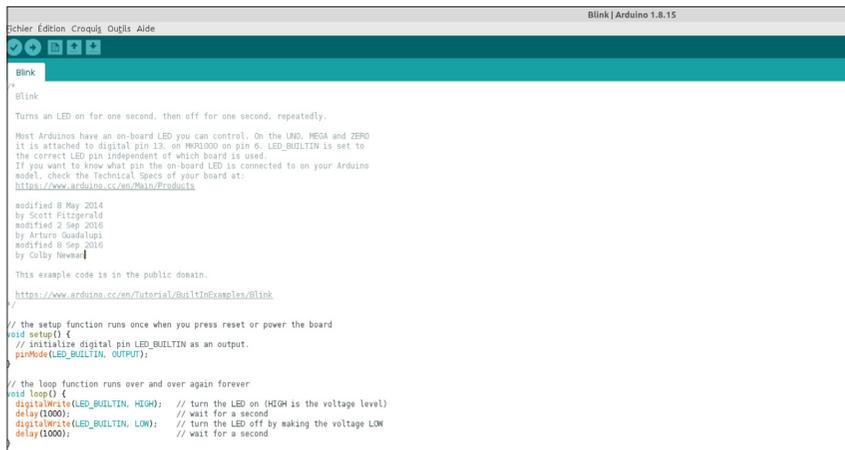
De la sorte, la valeur de tension moyenne dans le temps sera bien une valeur continument réglable entre 0 et 5V selon la valeur du rapport cyclique.

Par exemple, l'instruction « `analogWrite(6,X);` » permettra d'envoyer un signal sur la patte 6, compris entre 0 et 5V, selon la valeur de X entre 0 (0V) et 255 (5V).

- **Enfin, sur le module Arduino Uno, deux bornes supplémentaires peuvent être utilisées pour la communication série avec certains capteurs (par exemple le détecteur Grove fourni) :**
il s'agit des bornes SDA (Signal Data) et SCL (Signal CLock).

Interface/logiciel de programmation :

Pour programmer et charger les instructions sur la carte Arduino, il est nécessaire d'installer l'interface de programmation Arduino (disponible sous toutes les plateformes Windows, Mac, Linux) qui servira d'environnement de programmation, et également à charger les programmes dans le microcontrôleur via le port USB, mais encore à afficher les informations en provenance des capteurs (transmis via le « port Série »).



```

Blink | Arduino 1.8.15
-----
Blink
*
Blink
Turns an LED on for one second, then off for one second, repeatedly.

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
the correct LED pin independent of which board is used.
If you want to know what pin the on-board LED is connected to on your Arduino
model, check the Technical Specs of your board at:
https://www.arduino.cc/en/AboutProducts

modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalup
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

```

Prise en main d'Arduino et son interface de programmation :

Là encore, nous vous invitons à vous munir d'un bon tutoriel pour découvrir les bases de la programmation Arduino. C'est normalement à la portée de tous, moyennant le suivi d'exemples de difficulté croissante. Une fois le module et le langage pris en main, c'est un univers de possibilités qui s'ouvre...

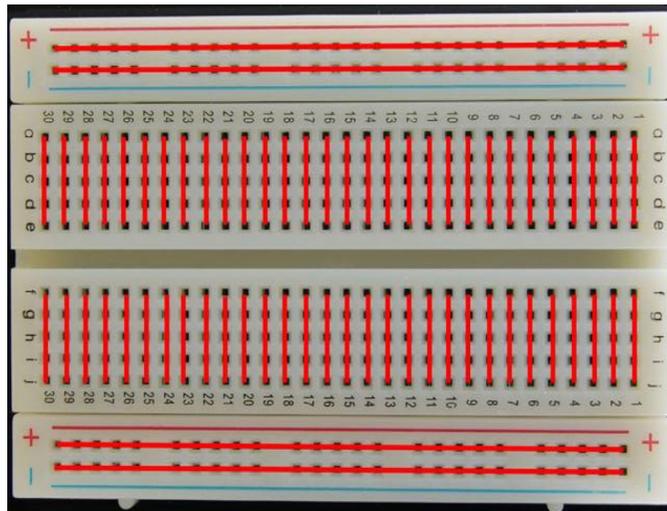
Quelques mises en garde toutefois pour éviter des erreurs courantes au démarrage :

- Pensez à ponctuer toutes les lignes de code par un point-virgule (sauf accolade ouvrantes...).
- Pensez à vérifier que le module Arduino est bien reconnu par votre interface de programmation, et que le port série sélectionné dans le logiciel correspond bien à votre module Arduino.
- Bien déclarer les numéros de broches (pins) dans le programme.
- Déclarer dans le programme l'ouverture d'une communication série avec l'Arduino pour récupérer des valeurs via le « moniteur » série : `Serial.begin(9600)`.
- ...

Plaquette test :

Le rôle de cette plaquette est de pouvoir connecter les **éléments électroniques et opto-électroniques** entre eux, et de les relier au module Arduino grâce aux fils (« jumpers »).

Les « trous » sont interconnectés de la façon suivante (les traits rouges représentent les connexions en « fond » de plaquette) :

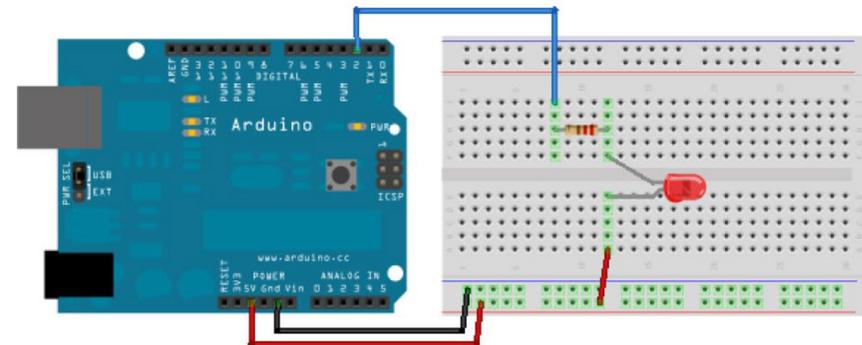


Les grandes lignes de connecteurs pouvant servir à créer des lignes de potentiel fixes

(0 Volts = masse = Ground = GND, et +5V= Vcc).

LEDs rouges, vertes et UV :

Ces diodes électroluminescentes permettent de générer un flux lumineux coloré en les connectant à des sorties digitales de l'Arduino. Attention, pour éviter un endommagement, il faut les connecter en série avec une résistance, et respecter la polarité de la diode (longue patte= anode= borne au potentiel supérieur (vers +Vcc), courte patte= cathode = borne au potentiel inférieur (Vers GND, via une résistance de 330 Ω)).



Module LED RGB :

Ce module contient 3 diodes (rouge, verte et bleue), combinées pour pouvoir créer une lumière différentes couleurs par synthèse additive (blanc, jaune, rose magenta, bleu cyan...). Il est commandé comme les LEDs décrites ci-dessus, via 3 sorties digitales de l'Arduino et 3 résistances pour alimenter les connecteurs R G et B du module. Il faut en outre connecter la patte GND à la masse GND du module Arduino.

Le réglage de la couleur se fait en commandant l'allumage de chaque LED avec une fonction de signal carré avec un rapport cyclique variable (modulation de largeur d'impulsion = PWM). La période du signal carré peut de préférence être inférieure à 50 - 100 ms pour éviter que l'oeil humain ne voie le clignotement (persistance rétinienne). La fonction utilisée est analogWrite.

Module Laser rouge :

Ce module permet d'émettre un faisceau laser continu rouge (633 nm) de puissance constante. Il se commande directement (sans résistance) en tout ou rien (on/off). Il faut connecter la patte G à la masse GND de l'Arduino, la patte V au Vcc=+5V de l'Arduino, et enfin la patte S à une sortie digitale de l'Arduino.

Le clignotement et/ou le réglage de la puissance d'émission se contrôle comme dans le paragraphe précédent, par modulation de la largeur d'impulsion (PWM) ou en utilisant la fonction digitalWrite.

Module capteur à photodétecteur :

Ce module permet d'effectuer une mesure quantitative du flux lumineux reçu par la surface sensible du photodétecteur. Le module (capteur à photorésistance) se câble facilement à l'Arduino : il faut connecter la patte G à la masse GND de l'Arduino, la patte V au Vcc=+5V de l'Arduino, et enfin la patte S à une entrée analogique (A0 par exemple) de l'Arduino. En opérant une lecture de l'entrée analogique A0, le programme Arduino fournira une valeur proportionnelle au flux lumineux reçu par le phototransistor.

Module détecteur de lumière Grove avec sortie VIS et VIS+IR :

ce module (plus élaboré que le capteur à photorésistance, et un peu plus délicat à mettre en oeuvre) intègre un détecteur TSL2561 qui constitue un autre moyen de mesurer le flux lumineux.

Il présente l'intérêt de contenir deux photodiodes : l'une destinée à la mesure du flux « visible », et l'autre fournissant le flux mesuré dans la gamme visible + proche infrarouge. Grâce à ces deux mesures, il est possible de calculer l'éclairement lumineux reçu par la surface sensible du photodétecteur (en unités standardisées de « lux », soit l'intensité lumineuse par unité de surface). C'est cette opération qui est implémentée de façon standard dans la librairie (ou bibliothèque) d'instructions qui sera installée pour utiliser le module Grove (voir plus loin).

En outre, les caractéristiques de détection du module peuvent être modifiées en termes :

- **de gain** (on peut régler un fort (respectivement faible) gain en situation d'éclairement faible (respectivement fort), ou mettre le détecteur en gain automatique (réglage par défaut)).
- **de temps d'exposition** : un temps d'exposition court permet une cadence de mesure très rapide, mais moins de précision. À l'inverse, on peut se régler sur le temps d'exposition le plus long (réglage par défaut) pour avoir une meilleure précision sur la mesure effectuée.

Connexion du capteur

Le module capteur Grove TSL2561 se câble à l'Arduino en connectant la patte GND à la masse GND de l'Arduino, la patte Vcc au Vcc=+5V de l'Arduino, et enfin les deux sorties SDA et SCL doivent être connectés aux bornes SDA et SCL de l'Arduino.

Installation des librairies d'instructions

La lecture du capteur se fait en interrogeant le module Grove TSL2561 via des commandes pré-programmées et référencées dans une librairie d'instructions qu'il faut installer.

Pour l'utilisation de ce module, nous vous conseillons d'installer les librairies suivantes :

- **Adafruit Unified Sensor** (gestion générale des capteurs Adafruit, version testée 1.1.3)
- **Adafruit TSL2561** (gestion du capteur TSL2561, version testée 1.1.0)

Pour installer des librairies : Dans le logiciel de développement Arduino, faire Outils > Gérer les bibliothèques, puis rechercher la bibliothèque souhaitée et l'installer (connexion internet requise pour l'installation)

Pour pouvoir utiliser les instructions des librairies installées, il faudra y faire appel dans votre code en incluant ces lignes en en-tête de votre programme Arduino :

```
#include <Adafruit_Sensor.h>
```

```
#include <Adafruit_TSL2561_U.h>
```

Utilisation du capteur en mesure d'éclairement lumineux

Pour commencer, nous vous conseillons d'utiliser le programme d'exemple fourni avec la librairie Adafruit_TSL2561. Pour cela, dans le logiciel de développement Arduino, cherchez

Fichier > Exemples > Adafruit TSL2561 > sensorapi.

Important

correction du code de l'exemple :

le capteur Grove fourni possède une broche d'« adresse » reliée à la masse, contrairement au capteur utilisé dans l'exemple. Il convient donc de modifier la ligne suivante de l'exemple :

```
Adafruit_TSL2561_Unified tsl = Adafruit_TSL2561_
Unified(TSL2561_ADDR_FLOAT, 12345);
```

→ *Adafruit_TSL2561_Unified tsl = Adafruit_TSL2561_
Unified(TSL2561_ADDR_LOW, 12345);*

Normalement, vous pouvez ensuite télécharger votre programme dans l'Arduino sans erreur, et obtenir, sur le moniteur du port Série (ctrl+shift+M) les résultats suivants :

```
Light Sensor Test
-----
Sensor:      TSL2561
Driver Ver:  1
Unique ID:   12345
Max Value:   17000.00 lux
Min Value:   0.00 lux
Resolution:  1.00 lux
-----

Gain:        Auto
Timing:      13 ms
-----

14.00 lux
15.00 lux
15.00 lux
```

En observant le code Arduino de l'exemple, vous remarquerez que le capteur est interrogé au moyen de la commande :

tsl.getEvent(&event);

qui permet de restituer la valeur de l'éclairement en lux, après calcul dans les fonctions avancées de la librairie installée.

Utilisation avancée du capteur :

Réglage du gain et du temps d'exposition

Vous pourrez ensuite utiliser les fonctions plus avancées du capteur (réglage gain et temps d'exposition) en modifiant les lignes suivantes du code dans la fonction `configureSensor` :

```
void configureSensor(void)
{
  /* You can also manually set the gain or enable auto-gain support */
  // tsl.setGain(TSL2561_GAIN_1X);      /* No gain ... use in bright light to avoid sensor saturation */
  //tsl.setGain(TSL2561_GAIN_16X);     /* 16x gain ... use in low light to boost sensitivity */
  tsl.enableAutoRange(true);          /* Auto-gain ... switches automatically between 1x and 16x */

  /* Changing the integration time gives you better sensor resolution (402ms = 16-bit data) */
  //tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_13MS); /* fast but low resolution */
  tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_101MS); /* medium resolution and speed */
  //tsl.setIntegrationTime(TSL2561_INTEGRATIONTIME_402MS); /* 16-bit data but slowest conversions */
}
```

Accéder aux mesures de deux photodiodes :

Il peut être intéressant pour certains projets ayant trait à de la spectroscopie (rudimentaire) de pouvoir obtenir directement les mesures des 2 photodiodes intégrées dans le composant TSL2561.

Il sera nécessaire pour cela de faire appel à la fonction `getLuminosity` de la librairie, qui permet d'accéder à la mesure du spectre large bande (visible+infrarouge, photodiode 1) et du spectre infrarouge (infrarouge seul, photodiode 2).

Les modifications de code à apporter sont les suivantes :

- Dans la fonction `loop()` : Définir en début de fonction deux variables (entier non signé 16 bits) `broadband` et `infrared` pour les mesures correspondantes au spectre large bande (photodiode 1) et infrarouge (photodiode 2), de la façon suivante :

```
uint16_t broadband;
```

```
uint16_t infrared;
```

Toujours dans la fonction `loop()`, là où l'exemple `sensorapi` fait appel à `tsl.getEvent(&event)` ; on peut accéder aux valeurs des deux spectres en appelant :

```
tsl.getLuminosity(&broadband, &infrared);
```

Il ne reste plus qu'à afficher les valeurs mesurées sur le moniteur série de l'Arduino au moyen des commandes suivantes :

```
Serial.print(«Broadband spectrum measure: «);
```

```
Serial.println(String(broadband));
```

```
Serial.print(«Infrared spectrum measure: «);
```

```
Serial.println(String(infrared));
```

Quelques idées d'expériences/notions abordables avec ces éléments :

MATÉRIEL					
LEDs rouges/vertes	LED UV	Module LED RGB	Module laser rouge	Module capteur à phototransistor	Module détecteur de lumière Grove avec sortie VIS et VIS + IR

IDÉES D'EXPÉRIENCES (DU PLUS SIMPLE AU PLUS COMPLIQUÉ...)						
Niv.1	Commande tout ou rien de LED (clignotement)	X	X	X	X	
Niv.1	Démonstration de la fluorescence (du schweppes par exemple)		X			
Niv.1	Principe de l'émission laser			X		
Niv.1	Mesure de flux lumineux avec un détecteur				X	
Niv.1	Principe du réglage de la puissance par modulation de largeur d'impulsion (PWM)	(X)		X		
	→ Synthèse / décomposition des couleurs / lumière blanche		X			
	→ Mesure de la persistance rétinienne	(X)		X		
	→ Mesure de la bande passante d'un détecteur	(X)		X	X	X
Niv.2	Mesure d'un flux visible et détectino de lumière infrarouge (télécommande télé par ex.) «spectroscopie rudimentaire»					X
Niv.3	Principe élémentaire de la détection synchrone (mesure de fluorescence par ex.)		X		X	
Niv.4	Détection de la lumière avec une LED (par mesure de variation de la capacité du semi-conducteur sous éclairage)	X				
Niv.4	Principe élémentaire de la técetion synchrone		X		X	
	etc...					

Au-delà des quelques idées listées dans le tableau ci-dessous, nous invitons les utilisateurs de la LightBox à fouiner dans les bases de données de projets Arduino existants pour trouver davantage d'idées et de ressources...

-Instructables (<https://www.instructables.com/circuits/arduino/projects/>)
 -Project Hub Arduino (<https://create.arduino.cc/projecthub>)
 -Hackster (<https://www.hackster.io/arduino/projects>)