

ARDUINO LES CAPTEURS

LUMINOSITE & PRESENCE

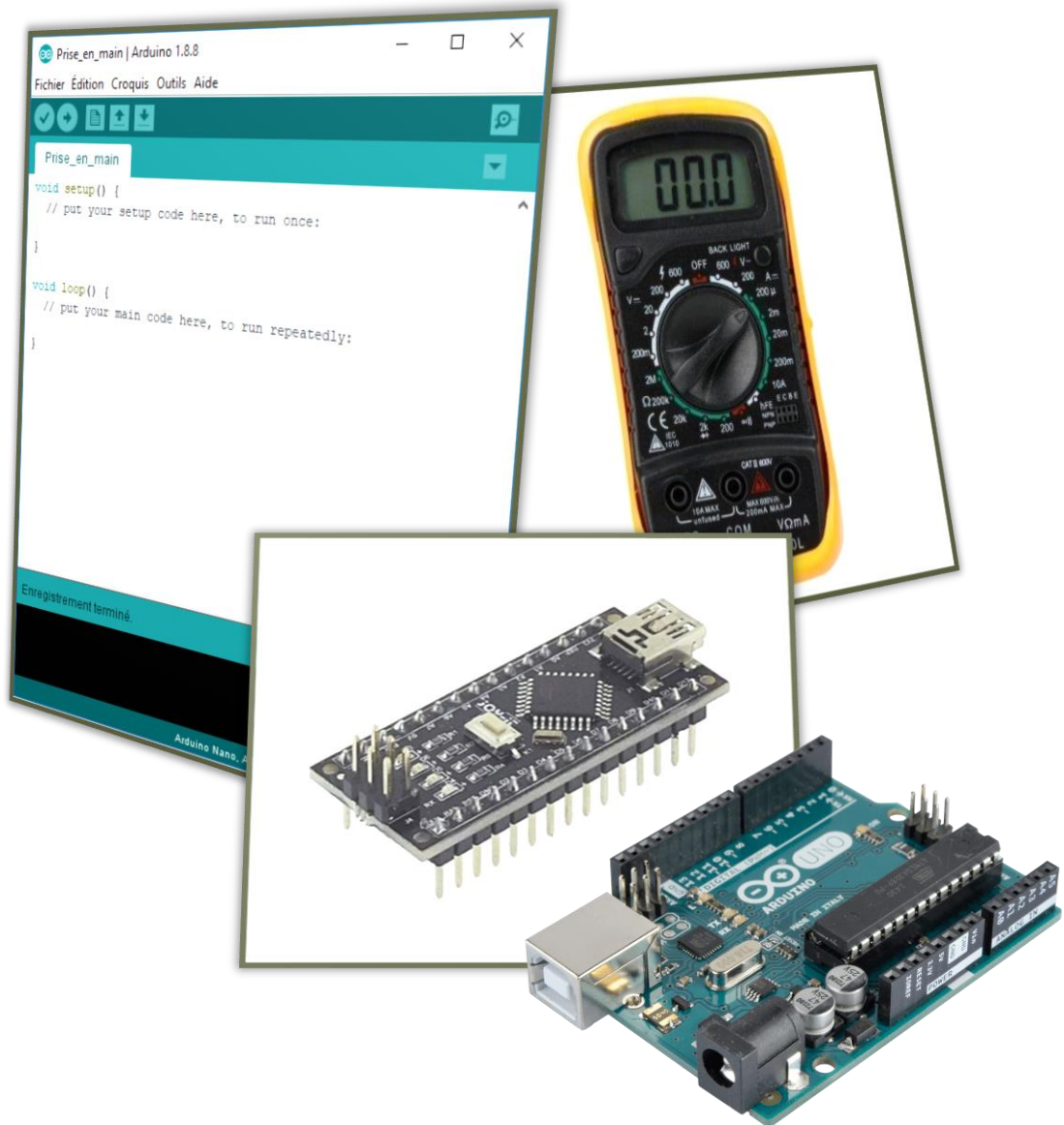


TABLE DES MATIERES

Introduction	2
Objectifs	2
Consignes	2
Matériels nécessaires	2
Le capteur de luminosité	3
Le capteur PIR	3
Capteur de luminosité	4
Analyse Du capteur	4
Câblage Du capteur	5
Observation du signal	5
Traitement du signal	6
Exploitation du capteur dans le cadre du projet	8
Capteur de présence	8

INTRODUCTION

OBJECTIFS

Le but de ce document est de vous apprendre à trouver par vous-même le fonctionnement d'un capteur et sa programmation.

Pour ce TP nous utiliserons le capteur de luminosité comme exemple.

CONSIGNES

Lire ce tutoriel dans l'ordre et en entier.

A chaque encadré comme celui-ci se trouve une manipulation à effectuer ou une question à répondre.

Vous répondrez sur un document Word soigneusement présenté et intitulé

« **NOM_Arduino_Capteurs** ».

Commencez par créer votre document Word de réponse et enregistrez-le dans votre espace personnel.

MATERIELS NECESSAIRES

Lorsque vous ferez de la programmation Arduino, vous serez amené à utiliser soit une carte Nano, soit une carte UNO.

Carte Nano :

- Carte Nano
- Câble USB-Mini-B
- Plaque de montage
- Capteur de luminosité
- Capteur PIR

Carte UNO :

- Carte UNO
- Câble USB-B
- Capteur de luminosité
- Capteur PIR

LE CAPTEUR DE LUMINOSITE

Il permet de mesurer une valeur de luminosité.



LE CAPTEUR PIR

Il permet de capter une présence via un mouvement.



CAPTEUR DE LUMINOSITE

ANALYSE DU CAPTEUR

Dans un premier temps il est important d'aller regarder la fiche technique et les caractéristiques du capteur. Pour cela, la meilleure des solutions est d'aller sur le site marchand où le capteur a été acheté.

<https://www.gotronic.fr/art-module-capteur-de-lumiere-mm109-24614.htm>

Cliquer sur le lien ci-dessus pour aller sur le site marchand.

Plusieurs données apparaissent qui seront utile pour l'exploitation de ce capteur :

DESCRIPTION	FICHE TECHNIQUE	AVIS
<p>Ce module capteur de lumière est une solution simple et pratique pour détecter le niveau de luminosité. Il communique avec un microcontrôleur type Arduino ou compatible via une liaison analogique.</p> <p>Ce circuit est livré avec un cordon et un connecteur à souder en fonction de l'utilisation.</p> <p>Caractéristiques: Alimentation: 5 Vcc Consommation: 20 mA Angle de détection: 60 ° Signal de sortie: - 0,5 Vcc dans l'obscurité - 5 Vcc en luminosité maxi Dimensions: 21 x 21 x 5 mm Référence Velleman: MM109</p>		

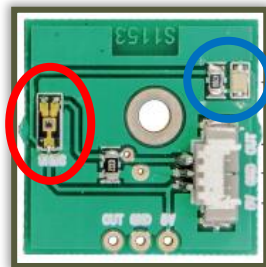
Dans la description on peut lire que ce capteur de luminosité est un capteur analogique.

Un capteur analogique est un capteur prenant un grand nombre de valeur entre un minimum et un maximum (ici, une tension entre 0 et 5V variant suivant la luminosité). Un capteur logique prendra lui seulement deux valeurs (0 ou 5V dans le cas d'un capteur Arduino).

On sait donc que notre capteur se branchera sur une entrée analogique de la carte Arduino.

Cliquer sur l'onglet « fiche technique », télécharger la fiche technique et l'ouvrir.

Des informations encore plus détaillées sont maintenant accessibles. Comme beaucoup de documentation celle-ci est en Anglais, il est donc important de pouvoir lire l'Anglais pour décoder rapidement les informations qui nous sont nécessaires.



Question 1 – Copier-coller l'image ci-dessus sur le document Word et donner le rôle de l'élément entouré en rouge et l'élément entouré en bleu.

CABLAGE DU CAPTEUR

Pour câbler un capteur à la carte Arduino nous allons utiliser les 3 câbles connectés au capteur.

Il y a des codes couleurs à respecter lorsque vous ferez des branchements et lorsque vous utiliserez des éléments déjà câblés :

- Le fil rouge représente l'alimentation en tension (le +). Il faut donc le brancher sur la borne « 5V » de la carte Arduino.
- Le fil noir représente la masse (le -). Il faut donc le brancher sur la borne « GND » (« Ground »).
- Le fil jaune représente le signal, ici analogique, qui variera en fonction de la luminosité. Il faut donc le brancher sur une borne analogique.

Câbler le capteur sur la carte Arduino en branchant les fils comme indiqué ci-dessus. Le fil jaune ira sur la borne A0.

OBSERVATION DU SIGNAL

Pour comprendre comment le signal varie en fonction de la luminosité et ainsi pouvoir traiter l'information il est nécessaire d'observer le signal.

Pour cela nous allons utiliser le moniteur série vu précédemment dans le « Tutoriel_Arduino_Introduction ».

Ouvrir le logiciel Arduino situé dans le dossier « STIDD » du bureau.

Enregistrer le programme dans votre espace personnel en la nommant « NOM_Arduino_Capteurs ».

```
1 const int Capteur_Lum = A0;
2 int Valeur_Lum;
3
4 void setup() {
5
6   Serial.begin(9600);
7
8 }
9
10 void loop() {
11
12   Valeur_Lum = analogRead(Capteur_Lum);
13   Serial.println(Valeur_Lum);
14   delay(100);
15
16 }
```

Recopier ce programme sur Arduino.

Téléverser-le sur la carte (si vous avez oublié la procédure reportez-vous au chapitre téléversement dans le « Tutoriel_Arduino_Introduction »).

Ouvrir le moniteur série et observer le résultat en changeant de place le capteur pour modifier la luminosité.

Question 2 – Commenter les résultats obtenus.

Pour les lignes 1, 2, 6, 12, 13 et 14, mettre un commentaire dans le programme permettant d'expliquer le rôle de ces lignes.

Pour placer un commentaire il suffit de placer deux slashes (« // ») et de mettre le commentaire à la suite.

Question 3 – Copier-coller votre programme commenté sur le document Word.

TRAITEMENT DU SIGNAL

Le but est maintenant de programmer notre carte pour effectuer une action en fonction du signal du capteur.

Imaginons que l'on souhaite programmer une lumière extérieure pour qu'elle s'allume lorsqu'il fait nuit et qu'elle se rallume lorsqu'il fait jour. Pour cela il est donc nécessaire de prendre en compte le jour et la nuit autrement dit de capter la luminosité extérieure. Nous avons le capteur parfait pour cela. La LED de l'Arduino jouera le rôle de la lumière extérieure.

Dans un premier temps il faut se fixer une **valeur de seuil** pour laquelle si la valeur du capteur de luminosité est inférieure à cette valeur de seuil alors la LED s'allumera.

En effectuant des tests grâce au programme précédent et au moniteur série, choisir une valeur de seuil.

Question 4 – Ecrire la valeur de seuil.

Il est maintenant nécessaire de rajouter la LED dans le programme Arduino et de définir une condition IF permettant de déterminer à quelle condition une action doit être effectuée.

La condition IF signifie SI. Elle se présente de la façon suivante :

```
if (condition) {  
    action;  
}
```

La condition doit être remplacée par une comparaison mathématique par exemple :

- $A == B$ (A équivalent à B) (**C'est bien un double égal !**)
- $A != B$ (A différent de B)
- $A < B$ (A inférieur à B)
- $A > B$ (A supérieur à B)
- $A <= B$ (A inférieur ou égal à B)
- $A >= B$ (A supérieur ou égal à B)

Si la condition est respectée alors on effectue le programme qui se situe entre les deux accolades qui suivent.

Compléter le programme ci-dessous sur Arduino.

Téléverser-le sur la carte.

Ouvrir le moniteur série et observer le résultat.

Question 5 – Commenter les résultats obtenus.

```

1 const int Capteur_Lum = A0;
2 const int LED = 13;
3 int Valeur_Lum;
4 int Seuil = 300;
5
6 void setup() {
7
8     Serial.begin(9600);
9     pinMode(LED, OUTPUT);
10
11 }
12
13 void loop() {
14
15     Valeur_Lum = analogRead(Capteur_Lum);
16     Serial.println(Valeur_Lum);
17
18     if (Valeur_Lum < Seuil){
19         digitalWrite(LED, HIGH);
20     }
21
22     delay(100);
23
24 }

```

Pour les lignes 2, 4, 18 et 19, mettre un commentaire dans le programme permettant d'expliquer le rôle de ces lignes.

Question 6 – Copier-coller votre programme commenté sur le document Word.

Vous avez dû remarquer que la LED s'allume correctement mais qu'elle ne s'éteint pas. C'est normal puisque nous ne l'avons pas programmé.

De la même manière que la structure IF, il existe la structure IF ELSE.

La condition IF ELSE signifie SI SINON. Elle se présente de la façon suivante :

```

if (condition) {
    action;
}
else {
    action02;
}

```

La condition doit toujours être remplacée par une comparaison mathématique.

Si la condition est respectée alors on effectue le programme qui se situe entre les deux accolades qui suivent (action01) SINON on effectue le programme qui se situe après le « else » entre les deux accolades (action02).

Programmer la nouvelle structure permettant d'éteindre la LED si la condition par rapport au seuil n'est pas respectée. (Contraire de HIGH : LOW)

Question 7 – Copier-coller la structure IF ELSE sur le document Word.

EXPLOITATION DU CAPTEUR DANS LE CADRE DU PROJET

Pour le projet du bloc porte lumineux il est nécessaire de se placer dans le cas réel d'utilisation.

Le bloc porte lumineux sera situé dans le couloir. Le but sera de créer un mode automatique qui permet de d'allumer, d'éteindre et d'adapter l'éclairage du bloc porte en fonction de la luminosité du couloir.

Un problème qui peut intervenir concerne l'instabilité de la sortie. En effet, si la valeur du capteur de luminosité oscille rapidement autour de la valeur de seuil (si la valeur de seuil est de 300 et que la valeur de luminosité passe de 299 à 301, de 301 à 299, etc.), le bloc porte ne fera que s'éteindre et se rallume, ce qui n'est pas désirable. Pour remédier à cela il est possible de choisir 2 valeurs de seuil. Un seuil haut (variable Seuil_Haut) pour lequel le bloc porte s'éteindra lorsque la luminosité est supérieure à celui-ci et un seuil bas (variable Seuil_Bas) pour lequel le bloc porte s'allumera lorsque la luminosité est inférieure à celui-ci.

Ces deux valeurs devront être suffisamment écartées.

Définir les valeurs de seuils haut et bas en faisant les mesures dans le cas réel d'utilisation.

Question 8 – Ecrire les valeurs de seuils.

Programmer la solution et tester votre programme.

Question 9 – Copier-coller la solution sur le document Word.

CAPTEUR DE PRESENCE

Dans une démarche favorisant le développement durable il est nécessaire de réduire l'impact énergétique du bloc porte lumineux. Pour cela le bloc porte devra s'allumer seulement lorsqu'une personne est détecté devant le système.

Pour détecter une personne nous utilisons un capteur de présence.

Voici le lien d'achat du capteur :

<https://www.gotronic.fr/art-mini-capteur-pir-101100001-23850.htm>

Ici, nous avons affaire à un capteur de présence : Soit il détecte une présence soit il n'en détecte pas. C'est donc un capteur logique qui se branchera sur les bornes digitales.

En recherchant des informations sur le capteur et en suivant la procédure effectuée pour le capteur de luminosité, étudier le capteur de présence.

Dans le « Tutoriel_Arduino_Introduction » nous avons vu comment piloter la LED de l'Arduino.

Question 10 – Ecrire un programme permettant d'allumer la LED de l'Arduino lorsque le capteur détecte une présence. Copier-coller le programme sur le document Word.