**MCU / ARDUINO**

**SEQUENCE : BASES**

26/02/2023

[CONSIGNES 2](#__RefHeading___Toc487_2873694027)

[PARTIE 1 3](#__RefHeading___Toc489_2873694027)

[​ 1.1 Exercice 1 3](#__RefHeading___Toc491_2873694027)

[PARTIE 2 4](#__RefHeading___Toc493_2873694027)

[​ 2.1 Exercice 2 4](#__RefHeading___Toc495_2873694027)

[​ 2.2 Exercice 3 5](#__RefHeading___Toc497_2873694027)

[PARTIE 3 5](#__RefHeading___Toc499_2873694027)

[​ 3.1 Exercice 4 5](#__RefHeading___Toc501_2873694027)

[​ 3.2 Exercice 5 6](#__RefHeading___Toc503_2873694027)

# **CONSIGNES**

Vous allez mettre en place des montages électroniques pilotés par la carte Arduino. Vous réaliserez ces montages à l’aide du simulateur Tinkercad.

Renseignez vos nom et prénoms ci-dessous, ainsi que dans le nom du fichier que vous retournerez. Dans tinkercad, bien nommer votre projet avec le numéro de l’exercice.

Ce fichier sera complété avec vos réponses aux questions et avec les liens vers les montages que vous aurez réalisé.

Dans tinkercad, un lien vers votre projet sera obtenu en cliquant sur le bouton « envoyer vers » en haut à droite de l’éditeur tinkercad. Puis dans le cadre « partager via message instantané ou email » cliquez sur le bouton « inviter des utilisateurs ». Copier le lien fourni et l’insérer dans ce document.

Vous serez évalués sur :

- votre compréhension des montages électroniques et de leur dimensionnement (loi d’ohm, etc...). La clarté des montages réalisés (disposition, couleur des fils….)

- le fonctionnement de votre projet

- la rédaction technique et claire

- la bonne application de vos connaissances en langage C (fonctions, variables, commentaires, etc..)

De nombreux exemples sur internet proposent des exemples utilisant la fonction delay() qui est une fonction bloquante. C’est à dire que votre montage ne réagit plus. L’utilisation de la [fonction millis()](https://www.arduino.cc/reference/en/language/functions/time/millis/) est préférable, vous l’utiliserez si il vous reste du temps disponible et vous donnera un bonus.

Nom :

Prénom :

# **PARTIE** 1

## 1.1 Exercice 1

Matériel requis :

* 1 carte Arduino
* 1 platine d’expérimentation (breadboard)
* 1 LED
* 1 photorésistance
* des résistances

Objectif :

Nous souhaitons intégrer à l’éclairage public la capacité de s’adapter en fonction des conditions de luminosité. L’idée est de mettre en œuvre une LED et d’en ajuster la luminosité en fonction de la luminosité mesurée autour d’une photorésistance. Plus la luminosité relevée sera élevée, moins la LED devra éclairer.

Réalisation :

1. Comment fonctionne une photorésistance ?
2. Quelles sont les grandeurs physiques qui interviennent dans le comportement de notre système ?
3. La carte Arduino manipule uniquement des tensions en entrée et en sortie. Comment faites-vous le lien avec les grandeurs physiques précédentes ?
4. Quelles sont les entrées et sorties qui devront être pilotées par l’Arduino ? Est-ce qu’il s’agit de données numériques ou analogiques ?
5. Quelles sont les valeurs récupérées par l’Arduino sur une entrée analogique ?
6. Comment sont interprétées les tensions en entrée sur une entrée numérique ? Quelles sont les valeurs définissant un « 0 » et un « 1 » ?
7. Quelles sont les valeurs possibles en sortie de l’Arduino via PWM ?
8. Réaliser le circuit électronique dans tinkercad. Réaliser les calculs permettant de dimensionner les composants.
9. Comment mettez-vous en relation les valeurs récupérées par l’Arduino en entrée (la luminosité mesurée par la photo-résistance) et les valeurs souhaitées en sortie (la luminosité de la LED)
10. Réaliser le code

Lien vers votre montage :

# **PARTIE** 2

Matériel requis :

* 1 carte Arduino
* 1 platine d’expérimentation (breadboard)
* 4 LED
* des résistances
* 1 bouton poussoir
* 1 condensateur

## 2.1 Exercice 2

Objectif :

Nous souhaitons simuler le comportement d’un circuit doté d’un télérupteur et allumer et éteindre une LED à l’aide d’un bouton poussoir. Le comportement attendu est le suivant : un appui sur le bouton, la diode s’allume, un appui supplémentaire la diode s’éteint. L’opération est indéfiniment répétable.

Réalisation :

1. Schématiser le fonctionnement du télérupteur en représentant l’évolution des états du bouton et de la LED dans le temps
2. Il existe 2 méthodes différentes pour réaliser l’allumage d’une LED. Réaliser un montage tinkerCAD permettant de mettre en œuvre les 2 montages. Vous expliquerez les avantages et les inconvénients de chacun d’eux.
3. Lors de l’appui sur un bouton poussoir, il existe un phénomène de « rebonds » qui fait alterner les états lus aux bornes du bouton poussoir. Il existe 2 types de systèmes anti-rebonds, un système utilisant un condensateur, un autre système entièrement logiciel. Expliquez les avantages et les inconvénients de ces 2 systèmes.
4. Réaliser un programme simple permettant d’allumer la LED à l’appui du bouton
5. Réaliser le programme complet, tel que décrit ci-dessus

Lien vers votre montage :

## 2.2 Exercice 3

Objectif :

Avec le même circuit électronique, nous désirons implémenter le fonctionnement d’un clignotant. Au premier appui sur le bouton, 4 diodes disposées en ligne s’allument l’une après l’autre (et restent allumées). Au terme d’un cycle, elles s’éteignent toutes, puis se rallument selon le même procédé. Une pression supplémentaire sur le bouton poussoir stoppe l’allumage des LEDs après complétion du cycle en cours.

Réalisation :

1. Schématiser le fonctionnement du clignotant en représentant l’évolution des états du bouton et des LEDs dans le temps
2. Réaliser le programme permettant de réaliser le cycle d’allumage des LEDs
3. Ajouter le contrôle par le bouton poussoir pour obtenir le programme complet

Lien vers votre montage :

# **PARTIE** 3

## 3.1 Exercice 4

Objectif :

Nous voulons désormais simuler le fonctionnement de feux tricolores à une intersection. Pour simplifier, le montage sera uniquement constitué de 2 feux synchronisés. Ces feux seront modélisés à l’aide de deux séries de trois LEDs de couleurs rouge, orange et verte.

Le comportement à adopter vis-à-vis d’un feu et leur fonctionnement est défini dans le code de la route en France. Pour la simulation, le feu restera au vert pendant 6s, il restera au orange pendant 2s. Lorsque le feu passe du orange au rouge, on marquera encore une pause d’1s avant que l’autre feu ne passe au vert.

Réalisation :

1. Schématiser le fonctionnement des 2 feux en représentant l’évolution des états des LEDs dans le temps
2. Lister le matériel nécessaire pour réaliser le montage
3. Identifier les entrées/sorties qui devront être pilotées par l’Arduino
4. Réaliser le montage
5. Réaliser le code

Lien vers votre montage :

## 3.2 Exercice 5

Ces feux disposent désormais d’un mécanisme de détection de véhicules. Par défaut le feu est au rouge. A l’arrivée d’un véhicule (détecté par une photodiode), le feu passe au vert. On aura pour objectif de limiter le temps d’attente à un feu. Pour ce faire les véhicules sont traités par leur ordre d’arrivée. Si une file de véhicules est vide, le feu passe au rouge, pour céder le passage à l’autre file, si au moins un véhicule se présente. Enfin, si les files sont toutes deux saturées, et pour garantir une circulation fluide au carrefour, les feux ne sont au vert que pour une durée limitée.

Lien vers votre montage :