

# C12\_TP01 : Modélisation des systèmes à événements discrets

## Machines à états



### Table des matières

Problématique	1
Description du système (Lampe automatisée)	1
Cahier des charges	1
Création des diagrammes	1
Cahier des charges : Fonctionnement temporisé	2
Le cahier des charges évolue.	2
Cahier des charges : Intensité Variable	2
Cahier des charges : Détecteur de présence	3
Cahier des charges : Détecteur de présence (suite)	3
Ajout d'un détecteur de présence	3
Plusieurs éclairages indépendants	4

## Problématique

On se propose dans ce TP/TD de modéliser et simuler le fonctionnement d'un système, à événements discrets, simple.

La modélisation et la simulation du fonctionnement sera réalisée à l'aide du logiciel **Yakindu**.

Vous pouvez télécharger le logiciel soit la version professionnelle (for academic use) ou la version standard (qui devrait suffire). <https://info.itemis.com/state-machine/download-yakindustatechart-tools>

Une vidéo précise la procédure d'installation <https://youtu.be/UmdQPZvP5x4>

Vous devez avoir sur votre système une machine java <https://www.java.com/fr/download/> installée pour que le logiciel fonctionne.

## Description du système (Lampe automatisée)

### Cahier des charges

Le fonctionnement d'un éclairage extérieur est décrit par le diagramme ci-contre.

- À l'état initial l'éclairage est éteint.
- Un appui sur le bouton éclaire.
- Un nouvel appui sur le même bouton éteint.

Ce diagramme comporte deux états, un pseudo état (l'état initial) et deux transitions avec un événement associé. Lorsqu'un état est actif, les actions associées sont réalisées. Le passage d'un état actif à un autre n'est possible que lorsque l'événement associé à une transition paraît.

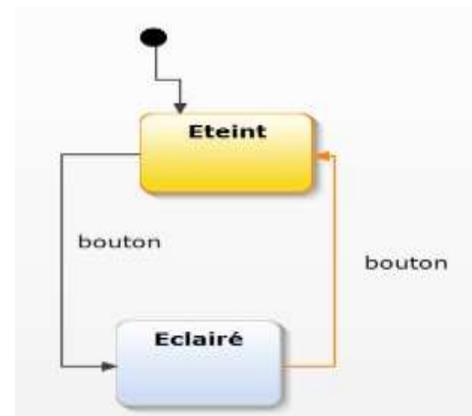


FIGURE 1 – Diagramme d'états du fonctionnement

## Création des diagrammes

- Lancer Yakindu
- Créer le projet et le dossier de travail pour cela suivre la procédure décrite dans la vidéo.
- Construire le diagramme

Ce diagramme simple comporte deux états (Éteint/Éclairé), un pseudo état (état initial) et deux transitions.

Pour les besoins de la simulation, l'événement « **bouton** » est un événement interne.

Yakindu permet de simuler le fonctionnement en animant l'évolution des états (<https://youtu.be/a1OrLmdLpOw>).

### Cahier des charges : Fonctionnement temporisé

Le fonctionnement initial est conservé mais si l'éclairage fonctionne plus de x secondes, il s'éteint automatiquement.

Le graphe ci-contre, Figure 2, décrit ce fonctionnement.

Le mot clef « **after** » est un mot réservé du langage qui déclenche un évènement interne lorsque le délai est écoulé.

La vidéo <https://youtu.be/yEO2w6q6yVA> décrit la procédure.

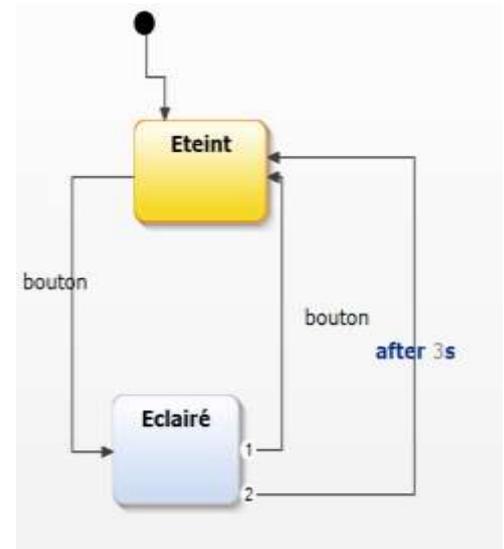


FIGURE 2 – Fonctionnement temporisé

### Le cahier des charges évolue.

### Cahier des charges : Intensité Variable

On ajoute la possibilité d'augmenter la luminosité à la demande de l'utilisateur.

On commence par ajouter une **interface** utilisateur qui permettra de communiquer par la suite avec l'extérieur.

Dans cette interface, on précise deux évènements d'entrée **in event**, **bouton\_on** et **bouton\_off** et une variable de type entier **var brilliance :integer**

Ce fonctionnement est décrit sur la figure 3.

La vidéo <https://youtu.be/an1kX9E2M-k> décrit la procédure.

On remarque dans ce diagramme d'autres particularités du langage.

Une transition peut :

- Ne comporter qu'un **évènement** : **user.bouton\_off** et **after 10s**. Si l'état précédent est actif et que l'évènement est vrai, alors l'état est désactivé et l'état suivant est activé.
- Comporter un **évènement** et une **action** : **user.bouton\_on/user.brilliance=1**. Si l'état précédent est actif et que l'évènement est vrai, alors la transition est franchie et pendant le franchissement, l'action associée est réalisée.
- Comporter un **évènement**, une **condition** et une **action** : **user.bouton\_on[user.brilliance<10]/user.brilliance+=1**.
- Le passage d'un état au suivant ne peut avoir lieu que si l'évènement est vrai ainsi que la condition booléenne.

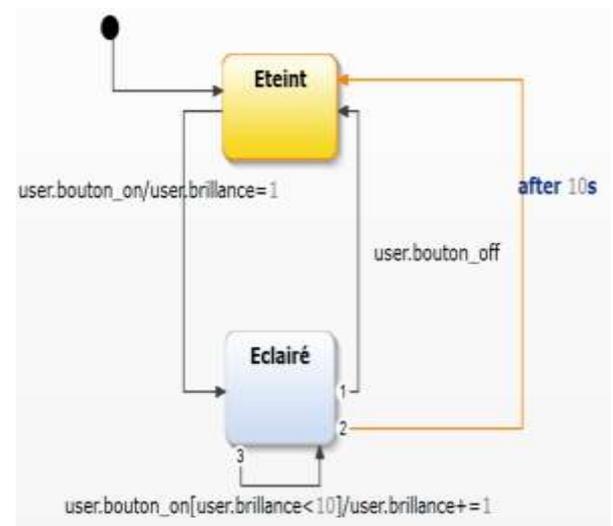
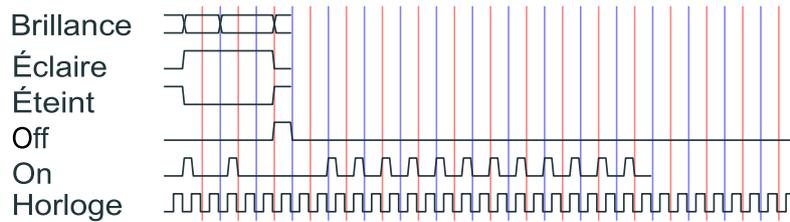


FIGURE 3 – Fonctionnement avec intensité lumineuse variable

À partir de la simulation compléter le chronogramme ci-dessous (la temporisation est réglée sur 4s) :



Cahier des charges : Détecteur de présence

On considère maintenant que l'éclairage est équipé d'un détecteur de présence.

- Au **repos** l'éclairage est éteint.
- Si le **capteur de présence** détecte un mouvement, la lumière s'éclaire.
- Au bout de **30 s** l'éclairage s'éteint si aucun mouvement n'est détecté, dans le cas contraire, la temporisation redémarre.
- L'utilisateur a la possibilité de forcer en manuel l'éclairage en appuyant sur le **bouton\_on**.
- L'utilisateur peut ajuster l'intensité en appuyant plusieurs fois sur le même bouton.
- La sortie du mode manuel intervient soit au bout de 10 s soit en appuyant sur **bouton\_off**.

Le diagramme de la figure 4 décrit ce fonctionnement.

La vidéo <https://youtu.be/XXcSAMDRvvc> précise la procédure.

Cahier des charges : Détecteur de présence (suite)

On trouve dans ce diagramme de nouveaux éléments du langage.

Un état **composite** : l'état **Automatique**, celui-ci comporte un diagramme dans l'état. On peut ainsi décrire un système en structurant l'étude d'un niveau le plus large vers un niveau plus précis. On remarque l'état initial qui est actif à l'instant de l'entrée dans l'état.

les deux instants

- **entry /** : les actions associées à cet instant sont réalisées inconditionnellement à l'entrée dans l'état : **entry / user.brillance=5**, à l'entrée dans l'état l'éclairage est allumé.
- **exit /** : les actions associées à cet instant sont réalisées inconditionnellement à la sortie de l'état : **exit / user.brillance=0**, à l'entrée dans l'état l'éclairage est éteint.

Ajout d'un détecteur de présence

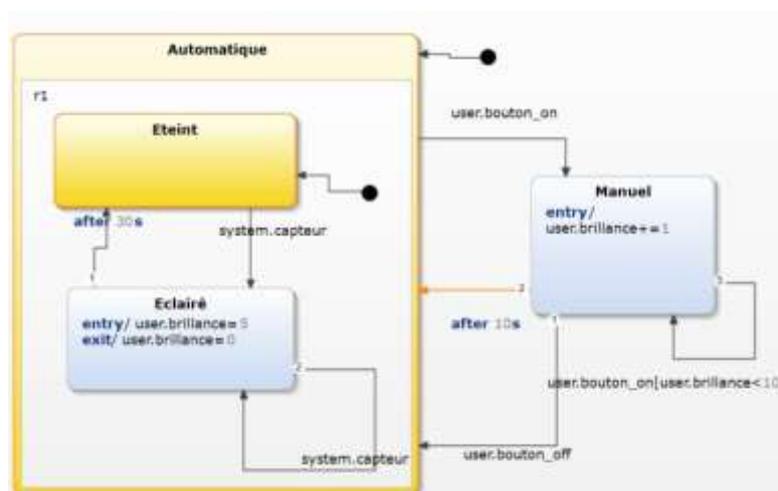


FIGURE 4 – Fonctionnement avec un détecteur de mouvement

## Plusieurs éclairages indépendants

La zone à éclairer comporte plusieurs espaces chacun équipé d'un détecteur qui doivent s'éclairer indépendamment les uns des autres avec des durées d'éclairage différentes.

Pour chaque espace, on retrouve donc le même fonctionnement que dans l'étude précédente. L'utilisateur pouvant bien sûr éclairer simultanément les deux zones.

Le fonctionnement est décrit par le diagramme de la figure 5 et la vidéo [https://youtu.be/w8Fgq2ECE\\_o](https://youtu.be/w8Fgq2ECE_o).

On trouve ici, dans l'état **Automatique** deux diagrammes qui décrivent chacun le fonctionnement d'une des lampes. Les deux diagrammes sont chacun dans une région. Les deux diagrammes sont dits **orthogonaux**.

Chaque diagramme est activé à l'activation de l'état encapsulant puis chacun se déroule selon l'évolution des états. La désactivation de l'état encapsulant désactive les deux diagrammes.

Ajout d'un détecteur de présence

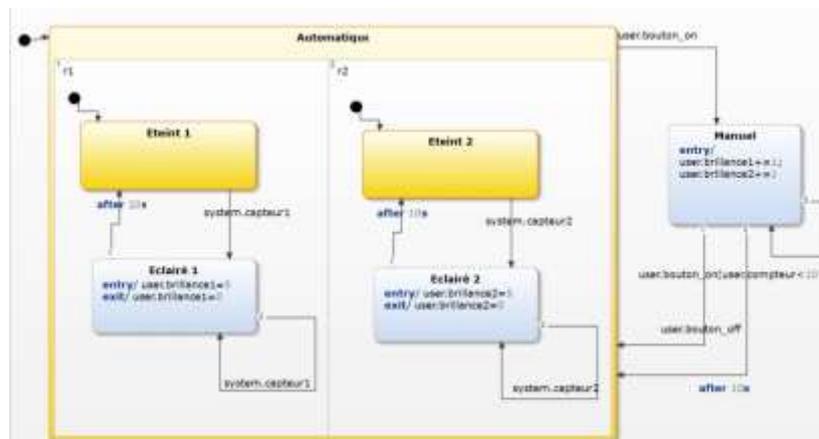


FIGURE 5 – Fonctionnement avec plusieurs éclairages indépendants