12.8 Systèmes séquentiels

Avant d'aborder les systèmes à événements discrets, nous allons définir les systèmes séquentiels.

La plupart des traitements ne sont pas uniquement combinatoires mais souvent séquentiels. Dans un traitement séquentiel le système doit pouvoir mémoriser certaines valeurs pour pouvoir les réutiliser.

Un système est dit séquentiel, lorsque la ou les sorties dépendent de la combinaison des entrées et de l'état précédent des sorties.

Une même cause (même combinaison des entrées) peut produire des effets différents et l'effet peut persister si la cause disparaît.

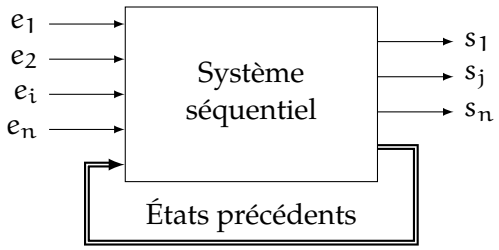
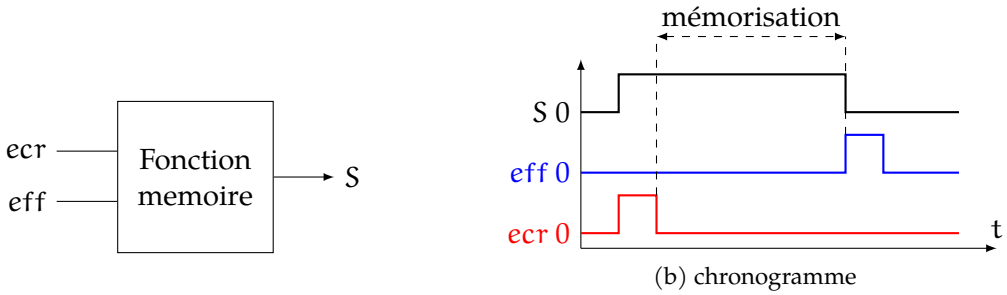


FIGURE 12.18 – Système séquentiel

12.8.1 Fonction mémoire

À l'apparition du signal *ecr* (écriture), la sortie passe à l'état 1, à la disparition du signal la sortie reste dans le même état (figure 12.19). À l'apparition du signal *eff* (effacement), la sortie repasse à l'état 0.

Le maintien de la sortie correspond à l'effet mémoire.



(b) chronogramme

FIGURE 12.19 – Fonction mémoire

a) Fonction mémoire à effacement prioritaire

Soit un système constitué de deux boutons poussoirs, m (marche) et a (arrêt) et d'un voyant V.

Un appui sur m éclaire le voyant V, si on relâche le bouton, la sortie reste à l'état 1, si on appuie sur a, quel que soit l'état du voyant V et de m, le voyant s'éteint. Nous allons construire la table de vérité traduisant ce comportement.

	m	a	V
les deux boutons sont relâchés et le voyant est éteint	0	0	0
le bouton m est appuyé, le voyant s'éclaire	1	0	1
le bouton m est relâché, le voyant reste éclairé	0	0	1
le bouton a est appuyé, le voyant s'éteint	0	1	0
les boutons m et a sont appuyés, le voyant est éteint	1	1	0

On constate sur cette table que le fonctionnement n'est pas combinatoire, en effet, pour deux états identiques, ($m = 0$ et $a = 0$) la sortie V a deux états différents.

Pour mettre en évidence l'état interne mémorisé, nous allons introduire une nouvelle entrée dans le tableau, $V_{t-\delta t}$ qui représente l'état précédent de V.

On construit la table en remplissant dans l'ordre les différents états des entrées et sorties, en reportant l'état de la sortie V dans $V_{t-\delta t}$ de la ligne suivante.

m	a	$V_{t-\delta t}$	V
0	0	0	0
1	0	0	1
1	0	1	1
0	0	1	1
0	1	1	0
0	1	0	0
1	1	0	0
1	1	1	0

En ajoutant l'état transitoire qui recopie la sortie sur l'entrée, la table comporte maintenant 8 combinaisons différentes des 3 variables m, a, et $V_{t+\delta t}$. Le fonctionnement est maintenant combinatoire (8 combinaisons pour $2^3 = 8$ possibles).

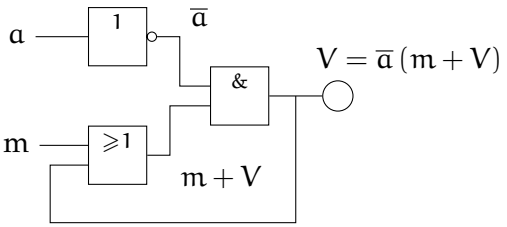
D'où l'équation logique du voyant :

$$\begin{aligned}
V &= m \cdot \overline{a} \cdot \overline{V_{t-\delta t}} + m \cdot \overline{a} \cdot V_{t-\delta t} + \overline{m} \cdot \overline{a} \cdot V_{t-\delta t} \\
V &= \overline{a} \cdot \left(m \cdot \overline{V_{t-\delta t}} + m \cdot V_{t-\delta t} + \overline{m} \cdot V_{t-\delta t} \right) \\
V &= \overline{a} \cdot \left(m \cdot \left(\overline{V_{t-\delta t}} + V_{t-\delta t} \right) + \overline{m} \cdot V_{t-\delta t} \right) \\
V &= \overline{a} \cdot \left(m + \overline{m} \cdot V_{t-\delta t} \right) \\
V &= \overline{a} \cdot \left(m + V_{t-\delta t} \right)
\end{aligned}$$

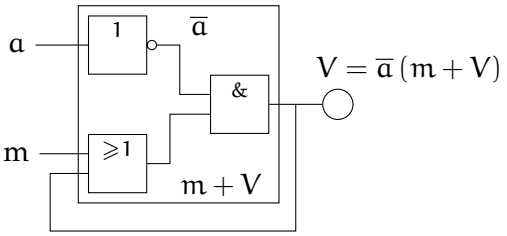
Le temps δt peut être aussi petit que possible, on peut donc écrire :

$$V = \overline{a} \cdot (m + V)$$

Et le schéma logique :



que l'on peut mettre sous la forme :



où on retrouve, la structure d'un système séquentiel.

On dit que le fonctionnement de cette mémoire est à « effacement prioritaire », en effet en cas de demande de mise à 1 et à mise à 0 simultanée, la mise à 0 l'emporte (dernière ligne de la table de vérité). On retrouve le fonctionnement décrit sur le chronogramme de la figure 12.20.

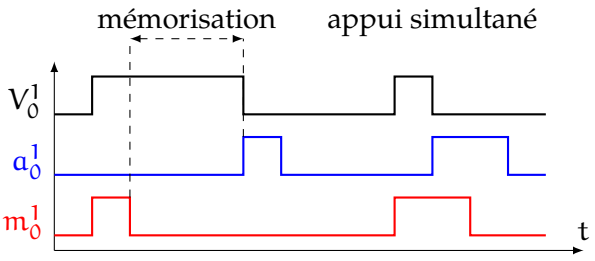


FIGURE 12.20 – Mémoire dite à « effacement prioritaire »

b) Réalisation d'une fonction mémoire en technologie électrique

Pour réaliser une fonction mémoire, il est nécessaire d'utiliser un relais qui permet avec un circuit d'auto-maintien de mémoriser l'impulsion sur un bouton (figure 12.21).

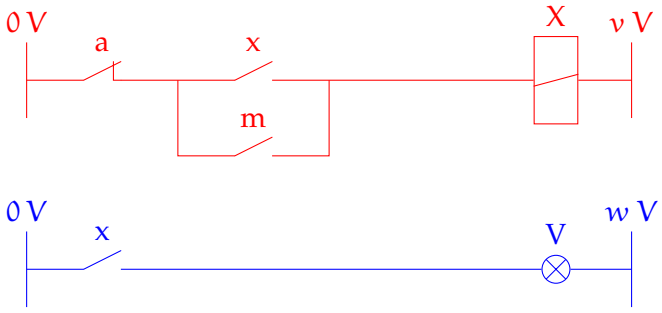


FIGURE 12.21 – Fonction mémoire à effacement prioritaire avec auto-maintien

12.9 Machine à états - *State machine diagram*

Il n'est pas toujours possible de décrire le comportement d'un système séquentiel, à partir d'une simple table de vérité. Le nombre de combinaisons possibles à étudier devenant très rapidement un obstacle à toute synthèse. (2^n combinaisons pour n variables).

Plusieurs outils, en général graphiques, ont été développés pour faciliter cette description.

Nous allons utiliser un diagramme du langage SysML pour décrire les systèmes à événement discret.

Le diagramme des machines à états du langage SysML permet de décrire l'évolution des états d'un système en fonction d'événements extérieurs.

Le diagramme d'état est un diagramme constitué d'états représentés par des blocs reliés par des arcs supportant les transitions.

12.9.1 États

On distingue 3 états de base.

— **L'état initial** représenté par un cercle plein noir.

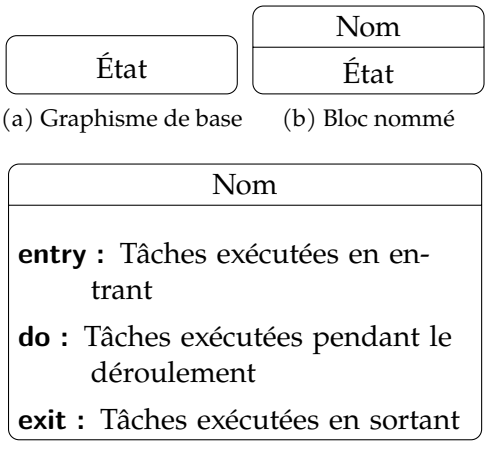
Cet état (pseudo état) est nécessaire, il indique le point d'entrée dans un graphe.

— **L'état final** représenté par un cercle plein noir cerclé.

Ce pseudo état n'est pas toujours nécessaire, il décrit l'état final d'un comportement.

— **L'état** proprement dit. Il représente un phase de vie du système, le système reste dans l'état décrit tant que les conditions d'évolution ne sont pas remplies. Il est représenté par un cadre précisant l'état et/ou les actions à réaliser pendant cette phase de vie.

Un état est représenté par un cadre (figure 12.22).



(a) Graphisme de base (b) Bloc nommé

(c) Bloc avec liste des tâches

FIGURE 12.22 – Graphismes d'un bloc

12.9.2 Les transitions

Les transitions d'un état à l'autre (ou sur lui-même) sont représentées par un lien orienté, la condition de transition est décrite sur le lien. Une transition représente le passage instantané d'un état vers un autre. On appelle état source l'état de départ d'une transition et état destination l'état d'arrivée. Une transition n'est évaluée que si l'état source est actif.

La syntaxe d'une transition peut être relativement complexe et comporter, jusqu'à trois éléments distincts, elle est notée **Événement[Condition de garde]/Action** :

- **Événement** : c'est la condition principale qui déclenche le passage d'un état à un autre. Cela peut être,
  - le changement d'état d'une variable (b : passage de 0 à 1 de la variable b et !b passage de 1 à 0),
  - la validation d'une condition notée avec le mot clef **when** (température atteinte par exemple : **when**$(T > T_0)$),
  - une condition temporelle : **after(2s)**.
- **Condition de garde** : c'est une condition booléenne qui complète l'événement et qui en contraint l'effet. Si une transition comporte un événement et une condition de garde : **événement[garde]** entre un état source et un état de destination, pour que le système évolue, il faut que l'événement soit vrai et que la condition de garde soit vraie.
- **Action** ou **activité** : décrit les actions qui sont exécutées pendant le franchissement.
- Chaque terme de la transition est optionnel.

La transition est franchie lorsque les conditions de franchissement associées sont vraies :

- Si l'événement apparaît (déclencheur) et si la condition de garde est vraie à ce moment alors la transition est franchissable : si l'état source était actif alors il se désactive et l'état de destination s'active. L'activité, si elle existe, est exécutée avant celles situées dans l'état de destination. La condition de garde n'est prise en compte qu'au moment de l'occurrence de l'événement.
- Si aucun événement n'est écrit, alors il est toujours vrai.
- Si aucune condition de garde n'est écrite, alors elle est toujours vraie.
- Une transition qui ne contient aucune condition de franchissement est dite automatique, elle est franchie dès que la tâche à l'intérieur de l'état source est terminée.

a) Transition réflexive

Arc orienté qui relie le même état (source = destination). Cela permet d'exécuter les activités associées aux instants « exit » et « entry » de nouveau.

b) Transition alternative

Lorsqu'un choix se présente entre deux ou plusieurs évolutions, il est possible d'utiliser le mot clef **else** pour préciser l'alternative. On peut aussi préciser l'alternative en utilisant le pseudo état de **point de décision** (◇)dans la transition (figure 12.23).

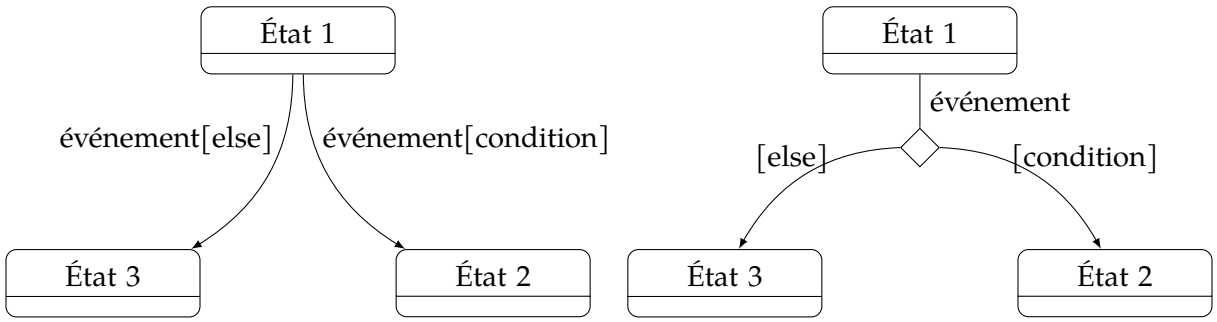


FIGURE 12.23 – Transition alternative

La clause **else** ne peut se trouver que dans une condition de garde.

12.9.3 Les blocs

Un bloc est donc une entité qui précise l'état du système lorsque le bloc est actif.

Le graphisme de base est un rectangle précisant l'état et/ou les actions à réaliser pendant cette phase de vie (figure 12.22a). Cette description peut être précisée en nommant le bloc (figure 12.22b).

Il est parfois nécessaire de préciser le comportement d'un état, on peut ainsi le détailler en prenant en compte 3 phases (figure 12.22c) :

— ce qui doit se passer en entrant dans l'état, cette phase est introduite par le mot clef :**entry**. Les taches ou actions décrites dans cette partie sont exécutées à chaque fois que l'on rentre dans le bloc, quelle que soit la durée de ce bloc ;

— ce qui doit se passer tant que l'état est actif, le mot clef **do** introduit cette phase. Les tâches du **do** sont réalisées tant que l'état est actif. Une action réalisée dans le **do** ne prend fin que si un événement est présent ;

— ce qui doit se passer en quittant le bloc, cette phase est décrite dans le mot clef **exit**. Les tâches ou actions décrites dans cette partie sont exécutées à chaque fois que l'on rentre dans le bloc, quelle que soit la durée de ce bloc

a) États composites

Un bloc peut aussi contenir une sous-machine à état. Cette encapsulation permet de décrire des fonctionnements plus complexes sans trop alourdir le graphisme. Un tel état est appelé état **composite** ou **super état**.

— L'activation de l'état composite entraîne l'activation du pseudo-état initial interne (rond noir).

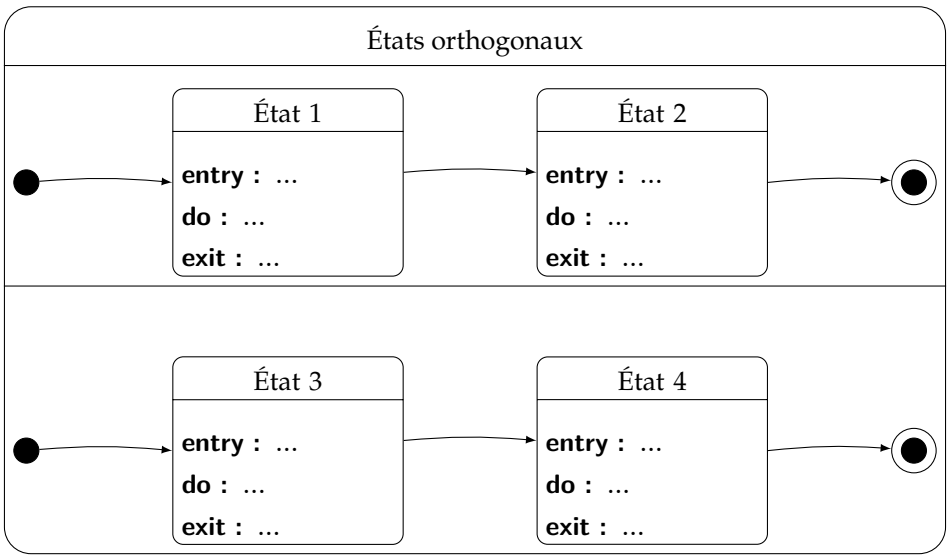
— La désactivation de l'état composite entraîne la désactivation de l'état actif.

— Chaque sous-état peut aussi être un état composite et ainsi de suite...

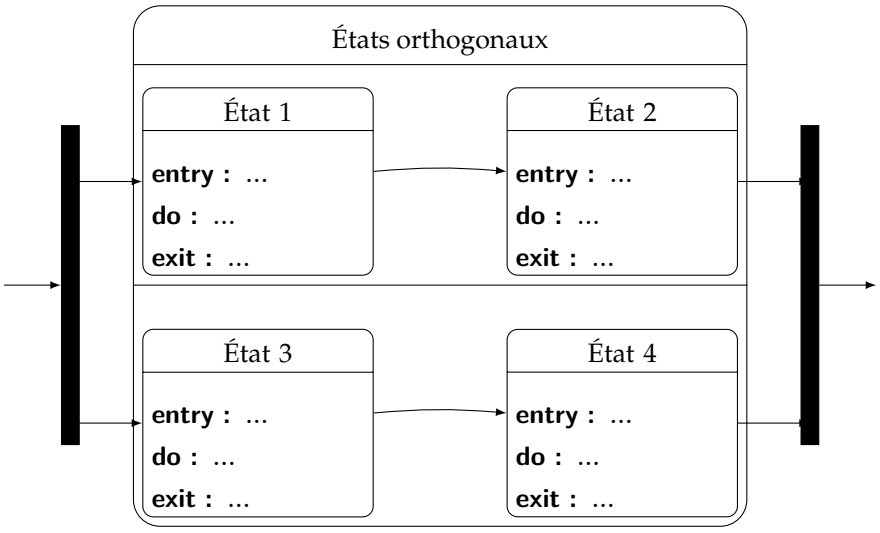
— Un état composite peut contenir des activités associées à entry, do et exit. entry et do sont traités avant l'activation du pseudo-état initial de la région, exit est traité au moment de la désactivation de l'état composite.

b) États orthogonaux

Un bloc peut aussi contenir plusieurs états dits **orthogonaux**, les états orthogonaux sont exécutés simultanément dès l'activation de l'état encapsulant (figure 12.24).



(a) Cycles exécutés simultanément



(b) Transitions fork et join

FIGURE 12.24 – États orthogonaux

Quand un état orthogonal est terminé, les autres états orthogonaux peuvent rester actifs.
Une transition sortant de l'état englobant termine simultanément tous les états orthogonaux.

Transition fork et join : Les deux pseudos états **fork** et **join** sont représentés par une barre épaisse :

— le fork permet de d'activer plusieurs états orthogonaux ;

— le join de regrouper plusieurs transitions issues de plusieurs états orthogonaux.

c) Prise en compte de l'historique

Il est parfois nécessaire de mémoriser l'état actuel avant de le quitter afin de le retrouver lors du retour dans l'état et de poursuivre le déroulement.

L'historique, est précisé par le pseudo-état (H) ou (H*) :

— *shallow history* (H) : permet à un état de niveau hiérarchique supérieur (état composite) de se souvenir du dernier sous-état, avant qu'il n'évolue vers un autre état,

— *deep history* (H*) : idem que précédemment mais avec la propagation de l'historique à tous les sous-états composites de niveaux hiérarchiques inférieurs.

12.9.4 Exemples

Exemple guidé : Lave-vaisselle

On considère un lave-vaisselle dont le fonctionnement est le suivant :

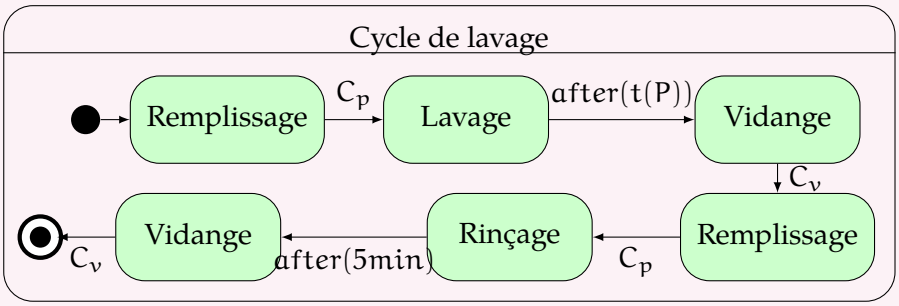
— Le cycle de lavage comporte un cycle de remplissage/lavage/vidange suivi d'un cycle remplissage/rinçage/vidange. La durée du cycle de lavage dépend du programme de lavage choisi.

— Pendant les cycles de lavage et rinçage, l'eau est chauffée et maintenue à la température prévue par le cycle.

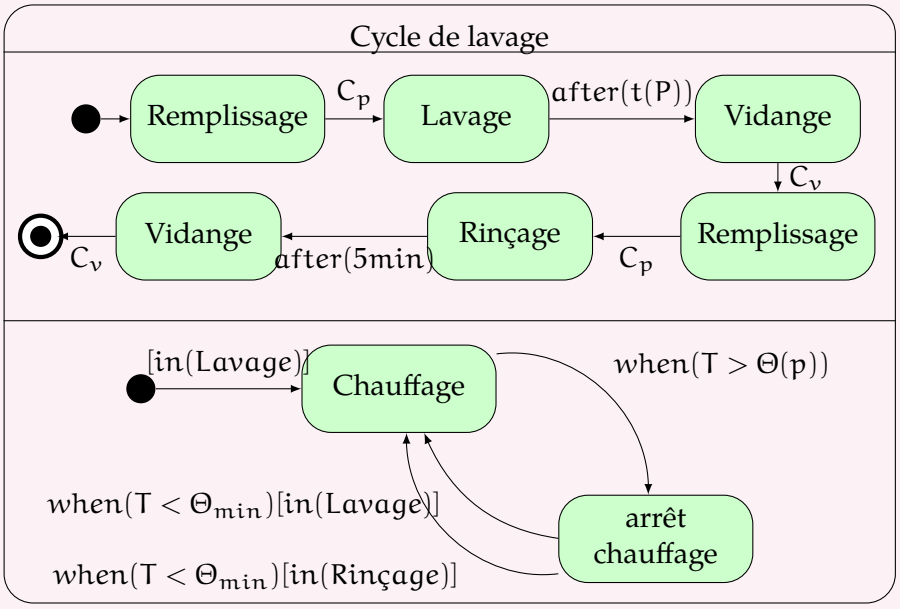
— Après la mise sous tension, l'utilisateur choisit parmi les cinq cycles de lavage en appuyant sur le bouton à impulsion `prog` qui incrémente le numéro du programme. Le cycle démarre après l'appui sur le bouton `start`.

— L'ouverture de la porte interrompt tout cycle en court, le cycle poursuit son déroulement lorsque la porte est refermée et après un appui sur `start`.

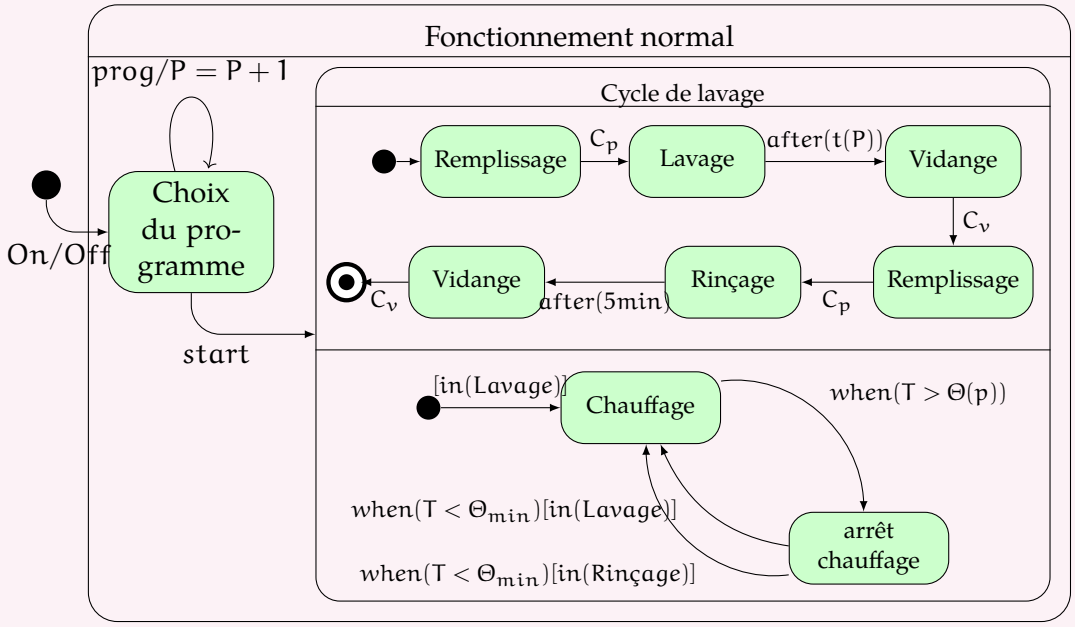
Cycle de lavage seul : Le graphe ci-dessous décrit le cycle de lavage seul, les événements cuve pleine et cuve vide sont notés C_p et C_v et $after(t(P))$ la durée du lavage pour le programme numéro P.



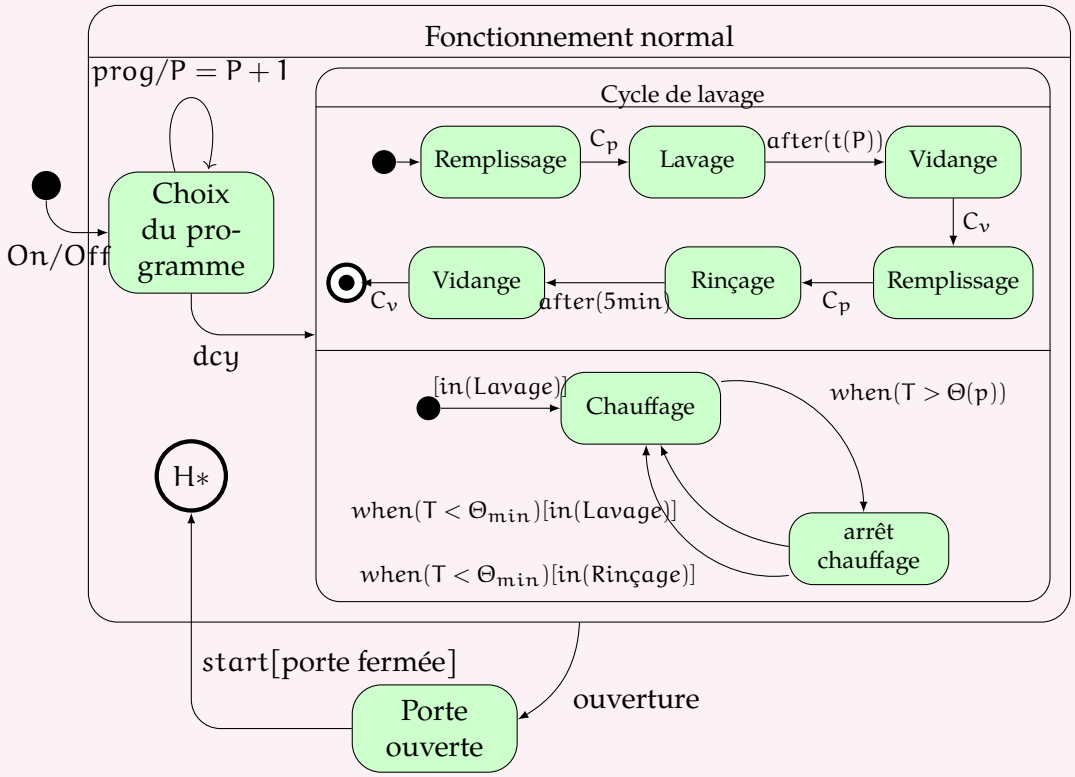
Lavage et chauffage : Les tâches de lavage et de chauffage se déroulent en même temps, il est donc nécessaire de préciser deux états orthogonaux. Le chauffage ne doit se réaliser que lorsque le système se retrouve dans l'état **Lavage** (mot clef **in**), cesser lorsque la température prévue par le programme n°P est atteinte (mot clef **when**) et reprendre si elle descend au-dessous d'une valeur minimale, à la condition que le système se trouve dans les états **Lavage** ou **Rinçage**.



Choix du programme et lancement : À chaque impulsion sur le bouton prog, le numéro du programme est incrémenté, le bouton start lance le cycle de lavage.



Ouverture de la porte : L'ouverture de la porte provoque l'arrêt du cycle de lavage, quel que soit l'état dans lequel il se trouve. Lorsque la porte est refermée, un appui sur start permet le redémarrage du cycle à partir de l'état dans lequel il était (on note **H***) précisant la reprise sur un historique profond.



a) Lampe - Bouton poussoir

Une lampe L est éclairée par un bouton poussoir B_p , une impulsion sur ce même bouton, l'éteint.

La figure 12.25 montre deux possibilités analogues pour décrire ce fonctionnement

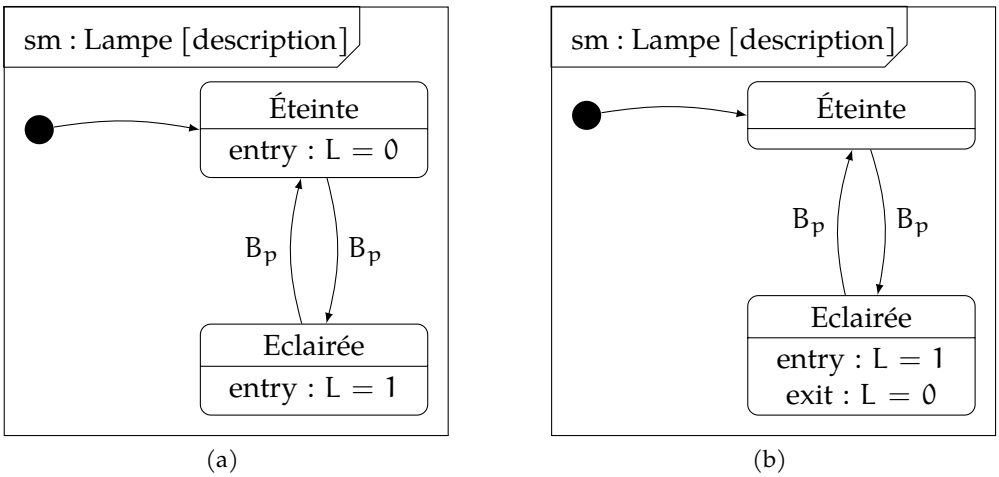


FIGURE 12.25 – Lampe et bouton poussoir : deux descriptions d'un même fonctionnement

On note dans cet exemple :

— l'utilisation de **entry** et **exit** ;

— les multiples représentations d'une même description.

Ces deux solutions ne sont pas les seules possibles.

b) Lampe temporisée

Un éclairage extérieur est mis en route par le passage d'une personne sous le détecteur (d), il fonctionne alors pendant 30 s. Si une nouvelle personne se présente alors que l'éclairage est actif, celui-ci redémarre pour 30 s.

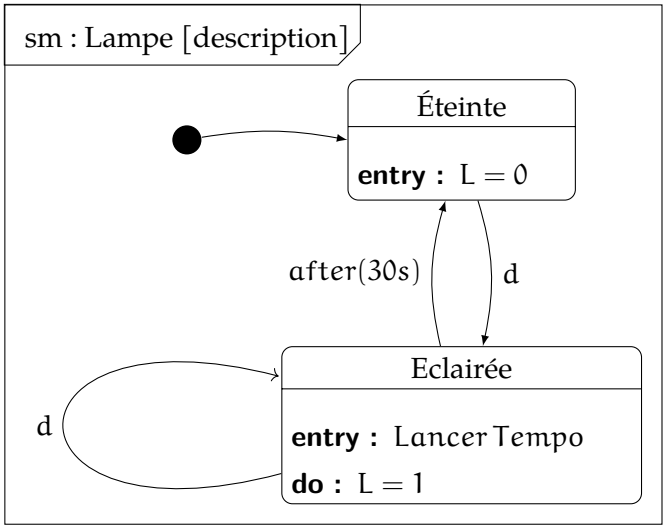


FIGURE 12.26 – Lampe temporisée

La figure 12.26 décrit le fonctionnement attendu, la boucle ré-entrante dans l'état « Éclairée » permet de relancer l'attente.

On note dans cet exemple :

— l'utilisation de **after** pour représenter une temporisation ;

— l'utilisation d'une transition réflexive ;

— l'action décrite dans le **do**, n'est pas arrêtée par la transition réflexive.

c) Fonctionnement d'une porte motorisée

La porte peut être dans l'un des trois états : « Ouverte », « Fermée » ou « Verrouillée ».
Le système peut répondre aux événements « Ouvrir », « Fermer », « Verrouiller » et « Déverrouiller ».
Le matin, la porte est déverrouillée et le soir, elle est verrouillée. La commande est réalisée par un bouton à clef placé à l'extérieur.
Les boutons « Ouvrir » et « Fermer » sont placés à l'intérieur, dans le local de surveillance.
Un détecteur permet de vérifier que le passage est libre.
Un voyant orange clignotant est alimenté lorsque la porte se déplace.

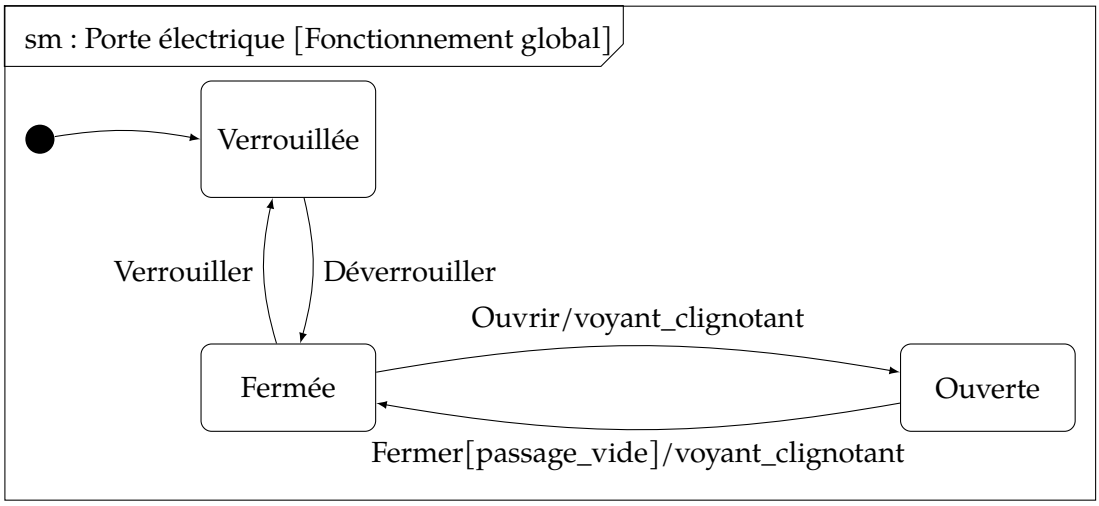


FIGURE 12.27 – Porte électrique

— À la mise sous tension (état initial) la porte est verrouillée.
— L'événement « Déverrouiller » met la porte dans l'état « Fermée ».
— L'événement « Verrouiller » met la porte dans l'état « Verrouillée ».
— La condition de passage « Ouvrir/voyant clignotant » met la porte dans l'état « Ouverte ». Pendant le déplacement, l'action « voyant clignotant » a lieu.
— L'évolution de l'état « Ouverte » à l'état « Fermée » ne peut se faire que si l'événement « Fermer » est vrai et que la condition de garde « [passage vide] » est vraie.
— Pendant le déplacement, l'action « voyant clignotant » a lieu.

12.10 Feuille de travaux dirigés n°12-b

Exercice 4 - Renault Twizy
Mines Ponts MP 2017 *Corrigé page 38*

Mise en situation

.1. Présentation

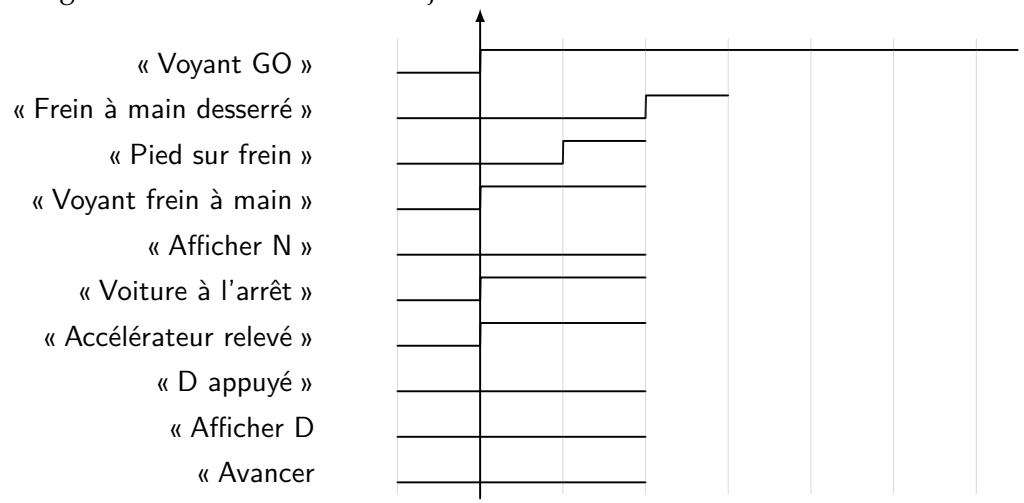
Dans le contexte actuel d'économie des énergies fossiles et de réduction des émissions de gaz nocifs, la Twizy est un quadricycle à propulsion électrique fabriqué par le constructeur automobile Renault. Elle constitue une alternative aux modes de déplacement urbains actuels. Se situant entre un scooter et une voiture, elle adopte un mode de propulsion entièrement électrique pour une autonomie d'environ 100 km. Son rayon de braquage très court et ses dimensions réduites lui permettent de stationner perpendiculairement au trottoir. Revers de la médaille, la Renault Twizy ne propose que deux places en tandem et un compartiment de 31 dm^3 sous le siège arrière. Logée sous le siège avant, la batterie, d'une capacité de 6,1 $kW \cdot h$ (105 $A \cdot h$), se charge complètement en 3 h 30 sur une simple prise secteur via un câble d'une longueur de trois mètres.

FIGURE 12.28 – Renault Twizy

Le diagramme fourni sur la figure 12.29 reprend la description du cycle de démarrage et d'arrêt du fonctionnement du véhicule Twizy en respectant la norme des diagrammes d'état définie dans le langage SysML.

Remarque : quand toutes les conditions sont réunies, dans l'état « Marche Avant », il faut appuyer sur l'accélérateur pour faire avancer le véhicule.

Q1. Après lecture du diagramme d'état, compléter le chronogramme dans la configuration où le démarrage moteur du véhicule est déjà effectué et où l'on souhaite avancer.



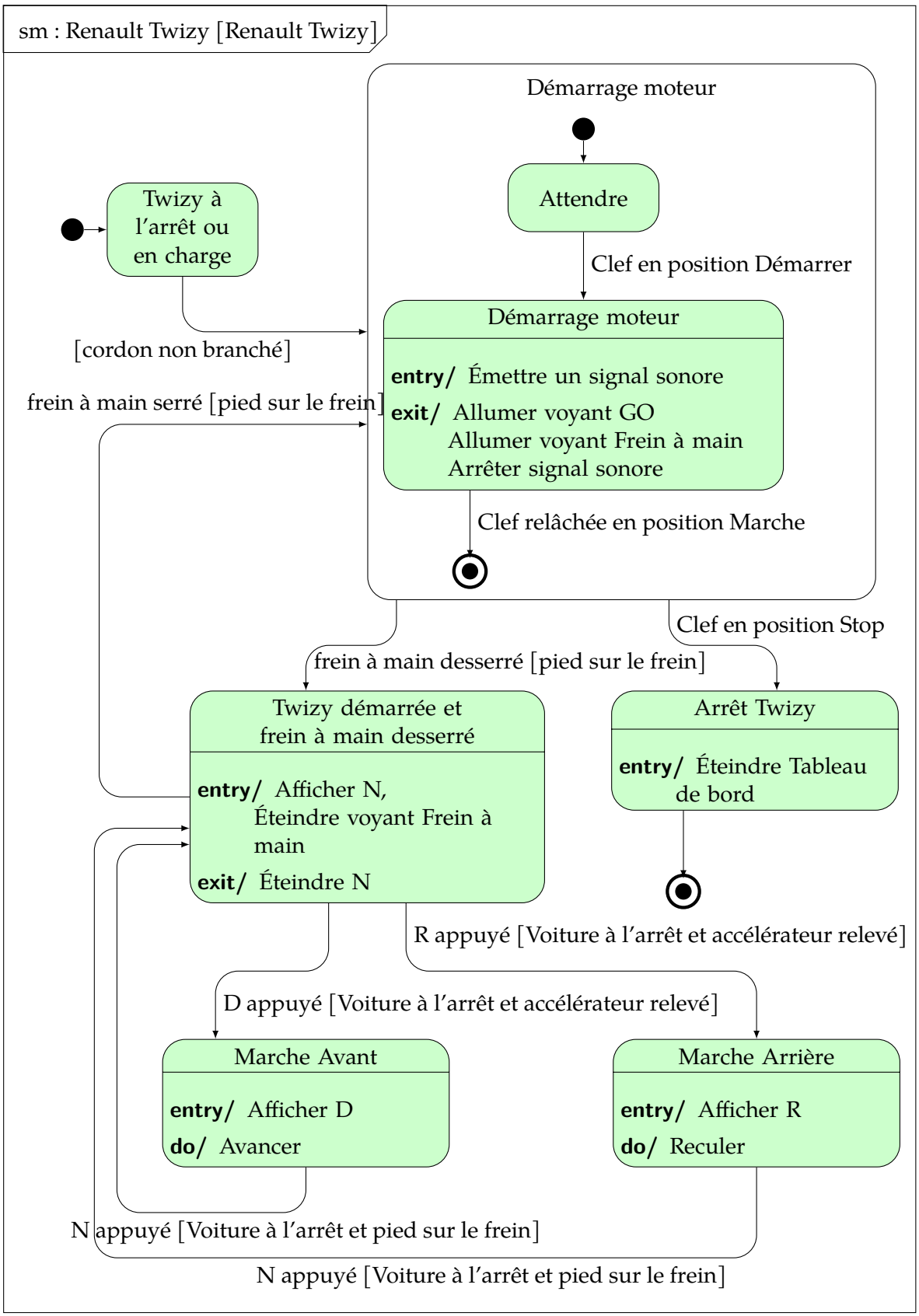


FIGURE 12.29 – Diagramme d'état partiel

Exercice 5 - Lève-vitre
CCINP PSI 2017 *Corrigé page 38*

A. Mise en situation

A.1. Fonctionnement

L'utilisateur demande simplement à ce que l'ouvrant se déplace jusqu'à une position prédéfinie. Une brève action sur l'interrupteur de sa part entraîne le déplacement complet de l'ouvrant. Dès lors, le système de contrôle/commande gère le déplacement de l'ouvrant dans le cas normal, mais aussi en cas de dysfonctionnement (perte de fonctionnalité ou présence d'un obstacle sur le trajet de la vitre). Il faut donc assurer un fonctionnement sûr et robuste du système d'ouvrant piloté automatisé pour éviter que le système blesse un occupant.

Le réducteur du lève-vitre est constitué d'un dispositif roue et vis sans fin. La roue possède $Z = 53$ dents et la vis est constituée d'un filet (figure 12.31). Le câble s'enroule sur le tambour de diamètre $D = 41{,}5\,\text{mm}$, solidaire de la roue. Le câble est solidaire du coulisseau sur lequel est fixée la vitre.

On prendra dans la suite la valeur $r = 0{,}39\,\text{mm} \cdot \text{rad}^{-1}$.

Q1. Justifier la valeur de r.

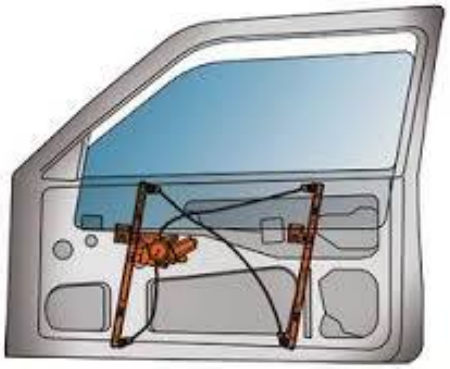


FIGURE 12.30 – Lève-vitre

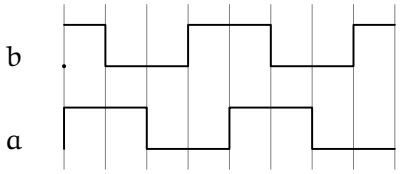
A.2. Modélisation du contact avec un obstacle

Dans le cas d'un lève-vitre, l'obstacle est souvent une main. Des études montrent que les phalanges sont très résistantes et peuvent supporter des efforts allant de 250 à 1 150 N en fonction des différentes phalanges. On modélise donc l'obstacle entre le châssis et la vitre par une raideur k (cette raideur peut varier de 10 à 50 N/mm).

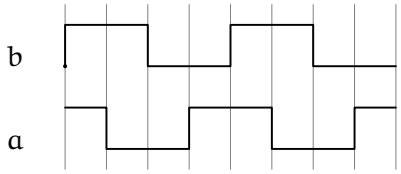
La position de la vitre est détectée à l'aide de capteurs à effet Hall situés près du moteur (figure 12.31). Une roue magnétique possédant 2 paires de pôles nord/sud est solidaire de l'axe du rotor du moteur. Deux capteurs à effet Hall sont placés en quadrature et repèrent les changements de champ magnétique (fronts montants et descendants) de la roue en fonction de la rotation du moteur.

Q2. Tracer l'évolution des signaux des deux capteurs à effet Hall (a et b) pour un tour du moteur dans le sens trigonométrique. On considère que le signal émis par le capteur a le niveau logique 1 lorsque le pôle nord de l'aimant est face au capteur.

sens trigonométrique



sens anti-trigonométrique



Q3. Quels sont les intérêts d'utiliser deux capteurs à effet Hall placés en quadrature ?

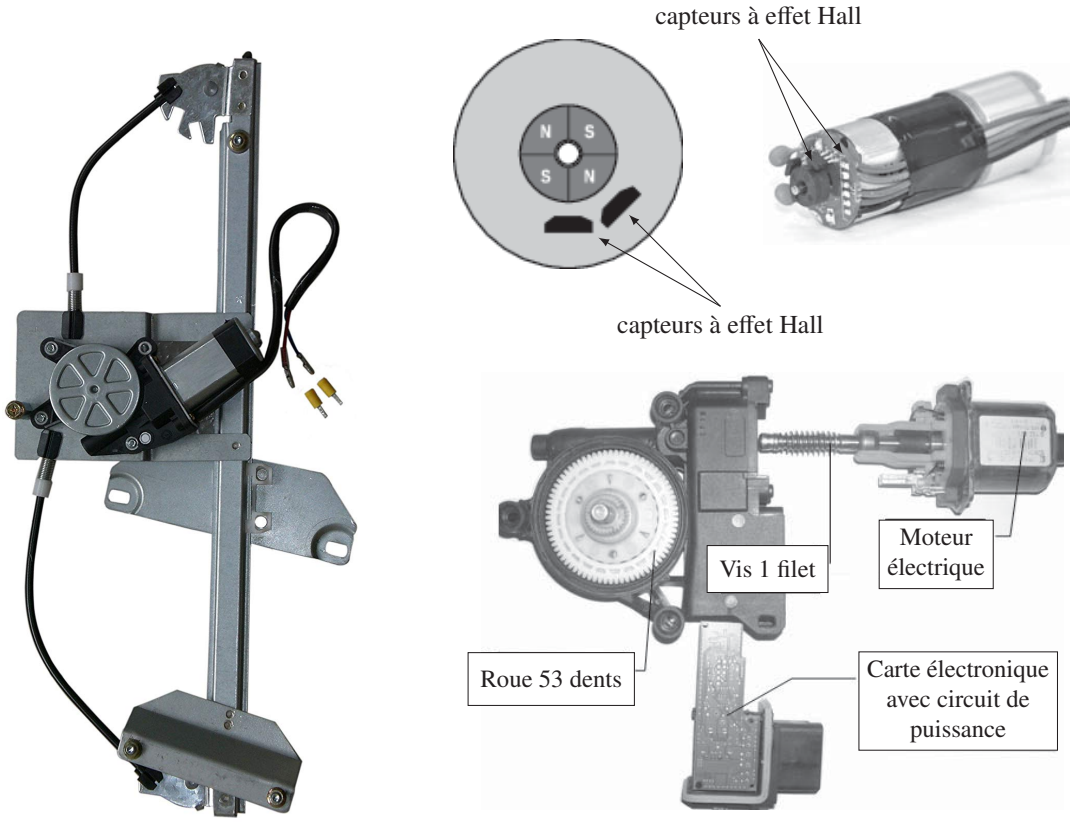
Q4. Montrez que $c = a \oplus b$ permet d'obtenir une précision de $1/8^{e}$ de tour du moteur.

Q5. Déterminer le plus petit déplacement de la vitre en mm qu'il est possible de mesurer avec ce capteur.

Q6. En prenant une raideur d'obstacle $k = 20\,\text{N} \cdot \text{mm}^{-1}$ correspondant à la dernière phalange de l'auriculaire, combien d'impulsions auront été comptées à partir du moment où la phalange commence à être écrasée jusqu'à ce que l'effort dans la phalange soit de 50 N ? Commenter ce résultat.

A.3. Algorithme de commande

L'algorithme finalement mis en place se base sur la variation des temps mesurés entre deux impulsions successives. Après la détection d'une impulsion, un prédicteur temporel permet de déterminer le temps auquel la prochaine impulsion est attendue. Si la nouvelle impulsion intervient avant le



(a) structure du lève vitre (b) Moteur et capteur à effet Hall

FIGURE 12.31 – Constituants du lève vitre

temps prédit, alors il n'y a pas de blocage, sinon un blocage est détecté et une alarme est déclenchée. En réalité, cette technique conduit à de fausses détections et une modification permettant d'améliorer la robustesse est de ne déclencher l'alarme qu'au bout de 3 dépassements du temps prédit. Cet algorithme est résumé sur la figure suivante pour lequel :

appui bouton haut est un événement qui survient quand le bouton « monter la vitre » est actionné,

M+ est la variable permettant de faire tourner le moteur dans le sens de la montée de la vitre, **M0** permet d'arrêter le moteur, cet événement permet soit la montée soit l'arrêt,

impulsion est un événement qui survient à chaque nouvelle impulsion envoyée par les capteurs,

fin course haut est un événement permettant de détecter l'arrivée en position haute de la vitre,

prediction() est une fonction qui renvoie le temps auquel la prochaine impulsion est attendue,

alarme permet d'activer l'alarme.

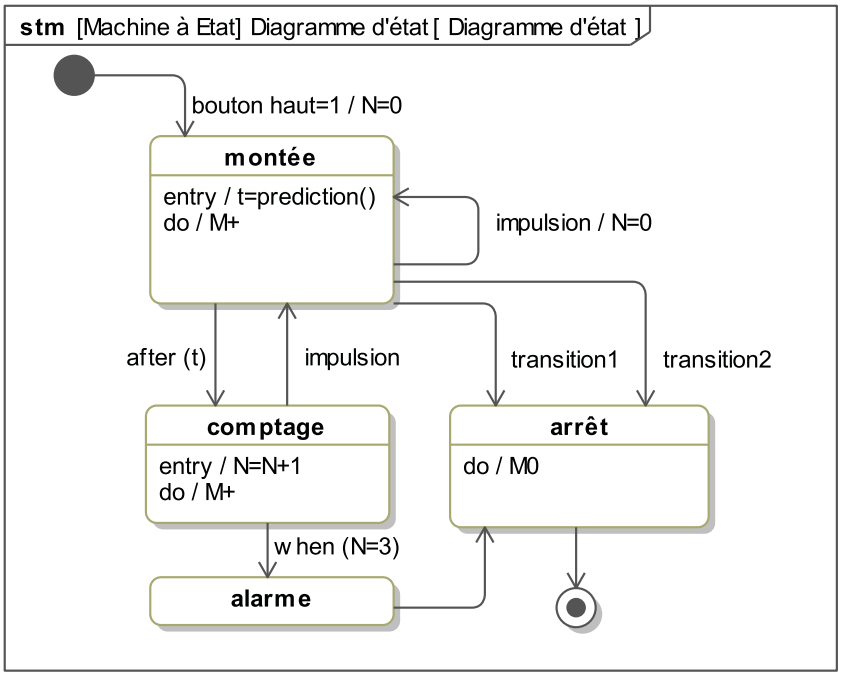
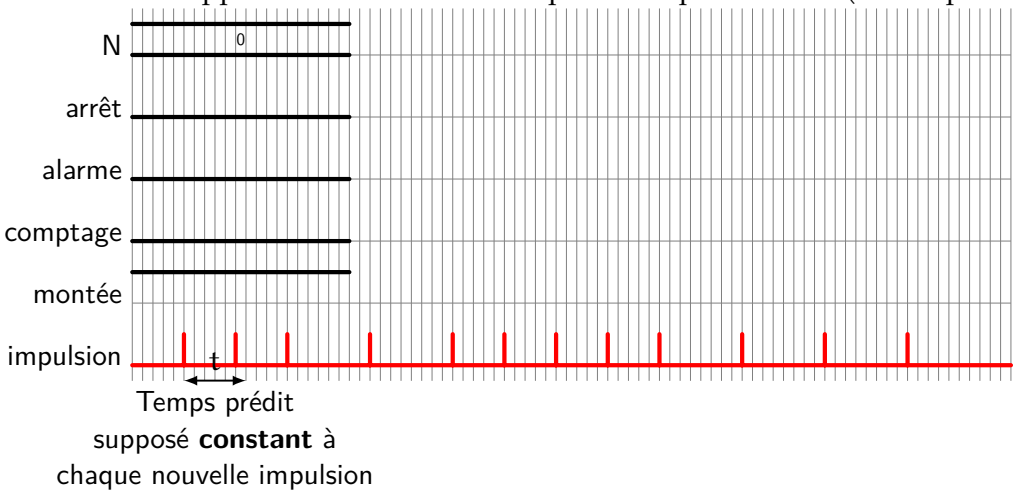


FIGURE 12.32 – Algorithme du fonctionnement

Q7. Donner l'expression des deux conditions notées « transition 1 » et « transition 2 » permettant de passer de l'état « montée » à l'état « arrêt » directement.
Q8. Compléter le chronogramme du document réponse DR4 en indiquant par des créneaux les durées pendant lesquelles un état est activé et l'évolution du contenu de la variable N. La durée de l'alarme et de l'arrêt est supposée très faible et sera représentée par un dirac (une impulsion).



Exercice 6 - Béquille électrique

Oral CCINP *Corrigé page 39*

A. Présentation

Une moto en stationnement peut être maintenue verticalement en équilibre grâce à une béquille centrale mécanique (figures 12.33). L'action de la part du pilote pour manœuvrer cette béquille mécanique peut nécessiter, pour les motos de grosse cylindrée, un effort très important. La masse à lever pouvant atteindre plusieurs centaines de kilogrammes.

Un kit de béquillage électrique est proposé en option sur certaines motos. C'est l'objet de l'étude qui suit. Ce dispositif présente les avantages :

— De permettre au pilote, assis sur la moto, de « béquiller » puisque la commande s'effectue directement à partir du tableau de bord de la moto.

— De soulever la moto, son pilote et ses bagages soit une masse maximale de 370 kg sans effort physique.
— D'assurer une protection antivol, le débéquillage n'étant possible qu'en mettant le contact électrique général de la moto.

B. Description du fonctionnement

Le contact général de la moto doit être enclenché pour que la béquille puisse fonctionner. La manœuvre de béquillage/débéquillage s'effectue à l'aide d'un bouton à 3 positions (B_{pd}, B_{pm}, position neutre) ajouté au tableau de bord. Le cycle de fonctionnement est régi par le module de commande. L'actionneur est un moteur électrique associé à un réducteur fixé sur la béquille elle-même. Le pignon de sortie extérieur au réducteur se déplace sur un secteur denté. Ce secteur denté est solidaire du châssis de la moto grâce à une bride de fixation. Deux capteurs fin de course (F_{ch} et F_{cb}) informent le module de gestion des positions "rentrée" et "sortie" de la béquille. Un buzzer signale au pilote que la béquille est en mouvement. La protection contre les surcharges (d'intensité) est assurée par un dispositif de contrôle du courant moteur.

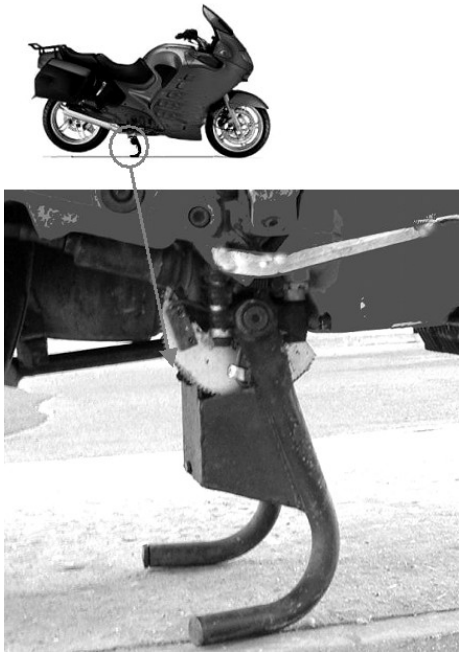


FIGURE 12.33 – Moto sur sa béquille

Cycle

— Débéquillage
  — la moto à l'arrêt sur les béquilles
  — contact
  — le conducteur appuie sur le bouton B_{pm} (une impulsion brève), la béquille monte (la moto descend), le buzzer fonctionne pendant le déplacement
  — arrivée en position rentrée (haute), le moteur de la béquille n'est plus alimenté, le buzzer stoppe.
— Béquillage
  — Le motard, sur la moto (avec passagers et/ou bagages), à l'arrêt, appuie sur le bouton B_{pd}.
  — La béquille ne descend que si le bouton reste appuyé. le buzzer fonctionne.
  — Si le motard relâche le bouton, la béquille s'arrête, mais le buzzer reste actif.
  — Un capteur vérifie en permanence que le moteur n'est pas soumis à une sur-intensité (effort trop important pour soulever la moto). En cas de dépassement, la béquille remonte automatiquement.
  — Après avoir allégé la moto, le motard peut redemander la descente de la béquille.
  — En position complètement sortie, le moteur s'arrête ainsi que le buzzer.

C. Module de commande

La commande de la béquille est réalisée par un circuit spécialisé (figure 12.34). Celui-ci à partir des consignes de commande (B_{pd} et B_{pm}) du motocycliste, de la position de la béquille (F_{ch} et F_{cb}) et de la sécurité électrique (S) génère les signaux de commande de la béquille (M et D) et active le buzzer (B).

— Avec le bouton BPm le pilote commande la montée de la béquille (la moto descend), et BPd, la descente (la montée de la moto).
— Le capteur F_{ch} détecte la position haute de la béquille et F_{cb} la position basse.
— L'information s est vraie si l'intensité parcourant le moteur dépasse la valeur maximale autorisée.

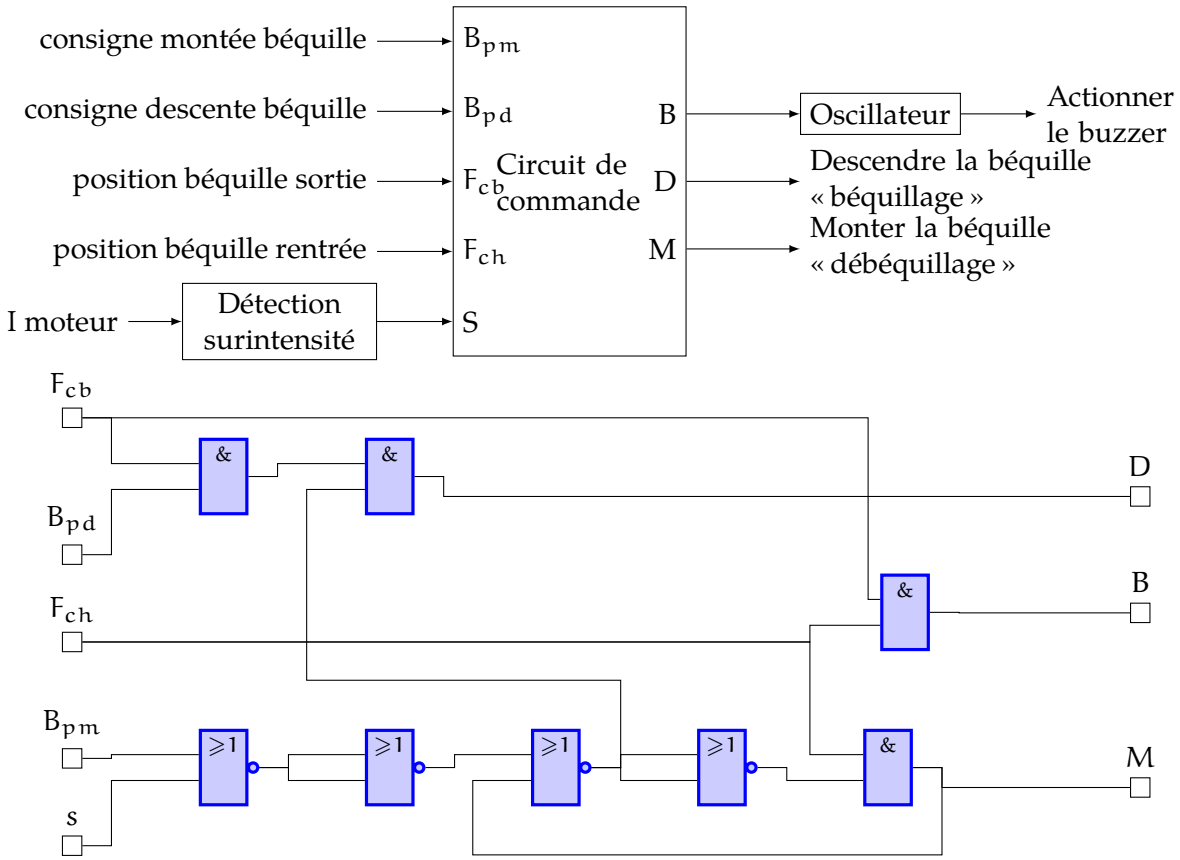


FIGURE 12.34 – Circuit de commande et codage des signaux

D. Étude de la commande

La commande de la béquille est réalisée par une carte électronique spécialisée, donc le schéma est fourni (figure 12.34).

Les deux boutons poussoirs, sont réalisés par des contacts à établissement de circuit, les deux fins de course (F_{cb} et F_{ch}) sont réalisées par des contacts à coupure de circuit.

Le capteur de sécurité s est au niveau logique 1 lorsque le courant dépasse une valeur de seuil.

Q1. À partir du schéma logique de la carte de commande :

Q1a. Déterminer les équations logiques de B, D et M en fonction de F_{cb}, F_{ch}, BP_d, BP_m, S et M.

Q1b. Quelle est la particularité de l'équation de M. Commenter.

Q2. Compléter le chronogramme de la figure 12.35, commenter.

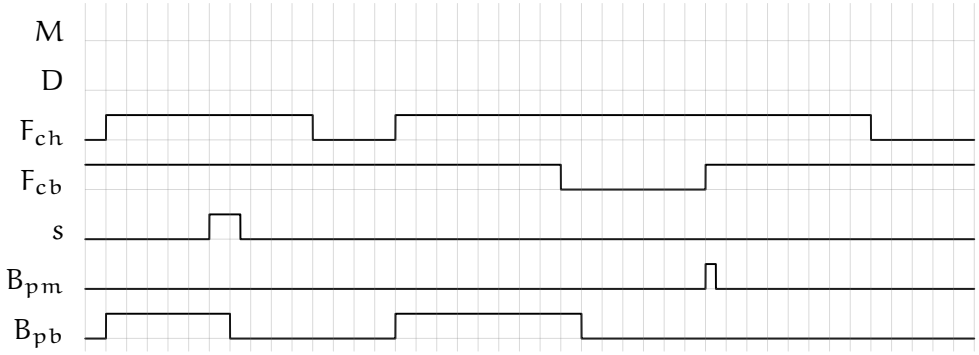


FIGURE 12.35 – Chronogramme à compléter

Q3. À partir de la description, proposer une description par une machine à états.