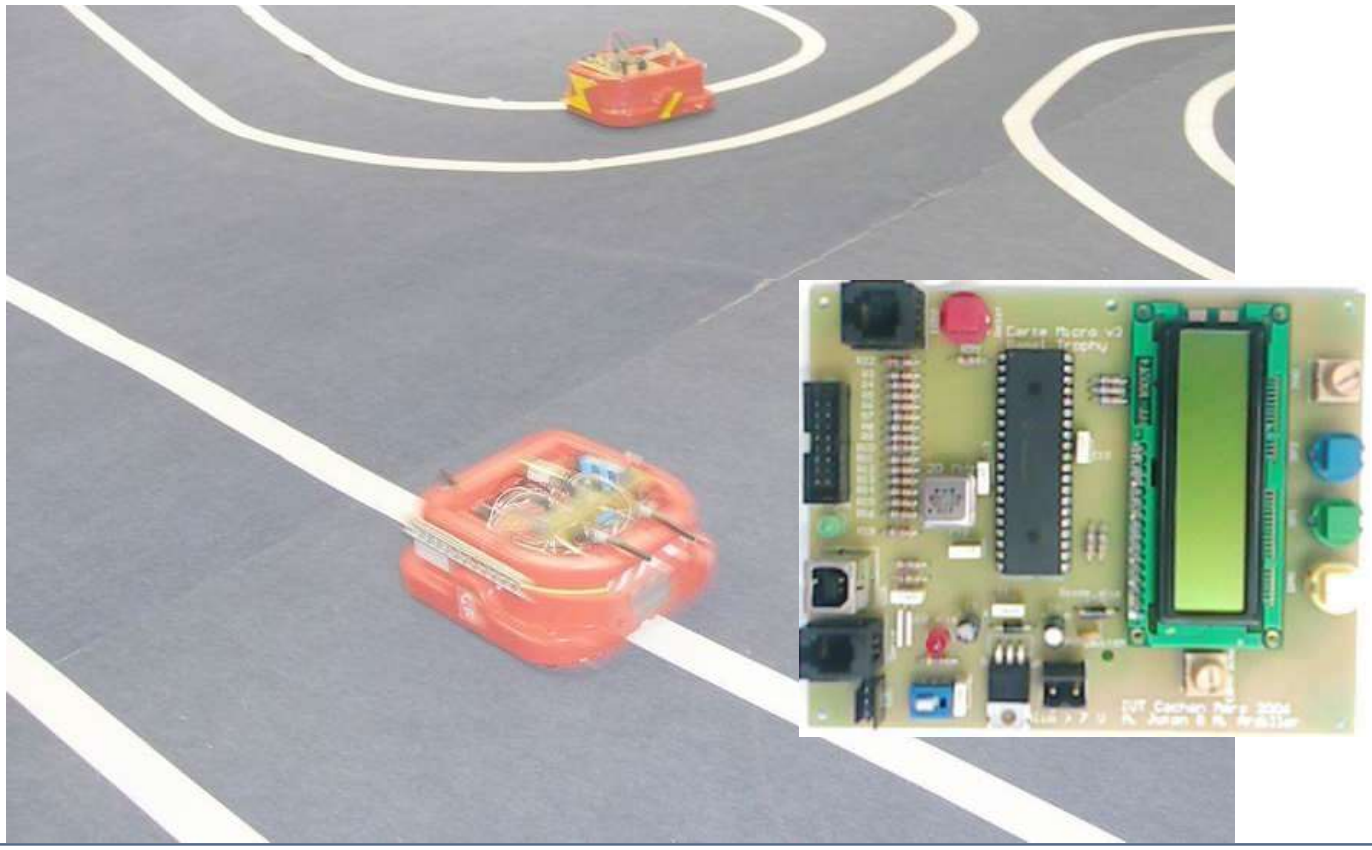
Programmer un système séquentiel



Gamel Trophy

Programmer un système séquentiel

1. Système séquentiel
2. Représentation d'une machine de Moore
3. Programmation en langage C

Programmer un système séquentiel

1. **Système séquentiel**
2. Représentation d'une machine de Moore
3. Programmation en langage C

Exemple de processus séquentiel simple

1. Le robot doit démarrer lorsqu'on enlève la fiche jack.
2. Le robot doit s'arrêter lorsqu'on actionne le contact de fin de course.
3. Le robot doit rester arrêté lorsqu'on replace la fiche jack. On peut alors reprendre le fonctionnement à la première étape.

Le nombre d'**états** dans lesquels le robot peut se trouver est fini :

- Attente de départ : le robot est à l'arrêt, la fiche jack est en place
- En marche : le robot avance
- Arrêt sur obstacle : le robot a heurté un obstacle et s'est arrêté

On parle de machine à états finis ou d'automate fini.

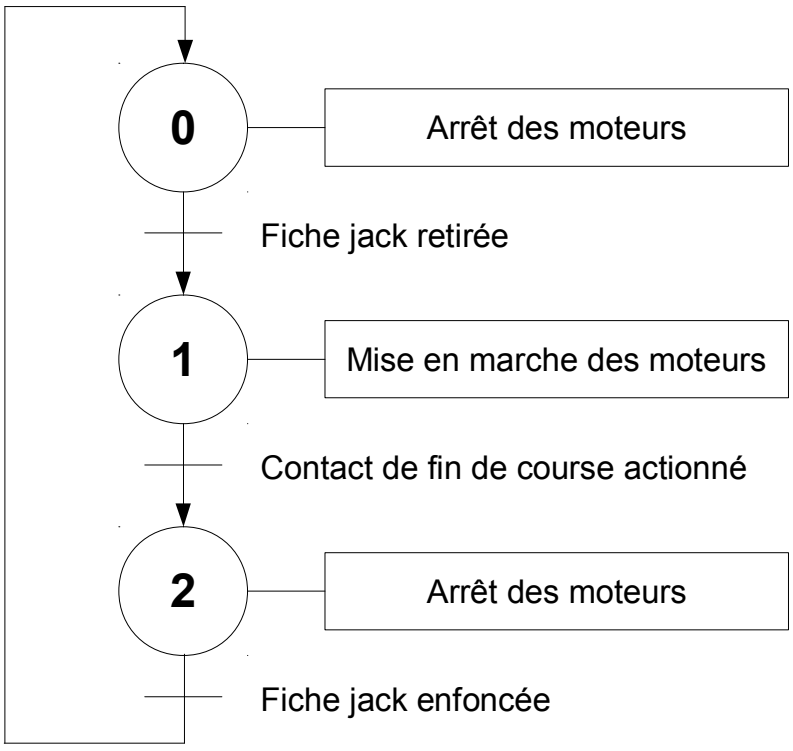
Les sorties du système ne dépendent que de l'état actuel.
Si on connaît l'état du robot, on sait comment agir sur les sorties : les moteurs peuvent fonctionner ou non.

Pour passer d'un état à l'autre, c'est-à-dire les transitions, on observe les entrées du système.

Cette machine à états finis particulière est une machine de Moore très utilisée en informatique industrielle.

Programmer un système séquentiel

1. Système séquentiel
2. **Représentation d'une machine de Moore**
3. Programmation en langage C



2. Représentation d'une machine de Moore

Un rond représente un état du système.

Chaque état est numéroté de manière unique.

A un instant donné, le système se trouve dans un et un seul état.

Une flèche représente une transition possible d'un état vers un autre.

A chaque flèche est associée une condition.

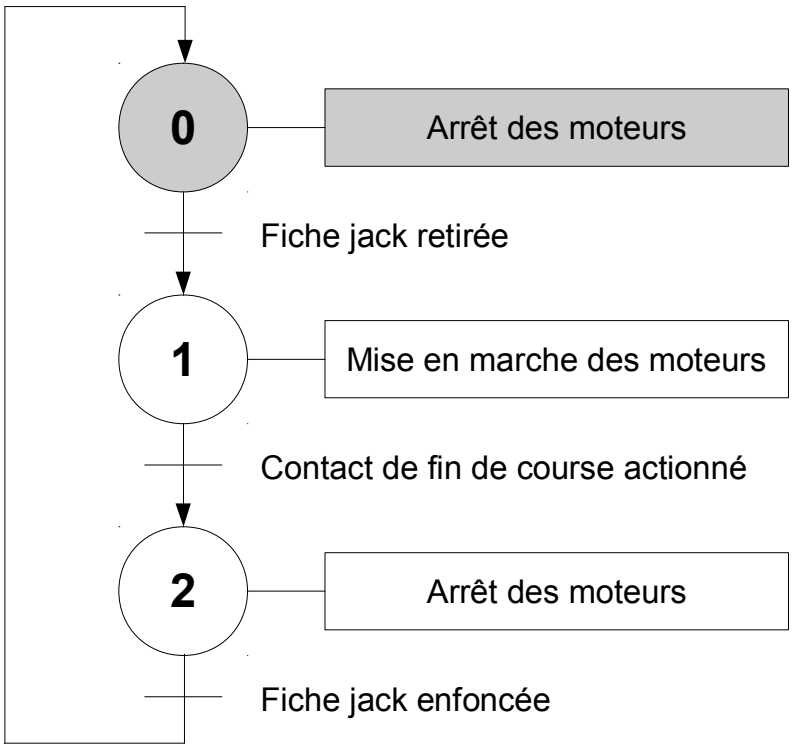
Si on se trouve dans un état et qu'une transition est vraie, alors on change d'état.

Les transitions qui partent d'un même état doivent être exclusives.

Si aucune transition ne peut être réalisée, l'état actuel est conservé.

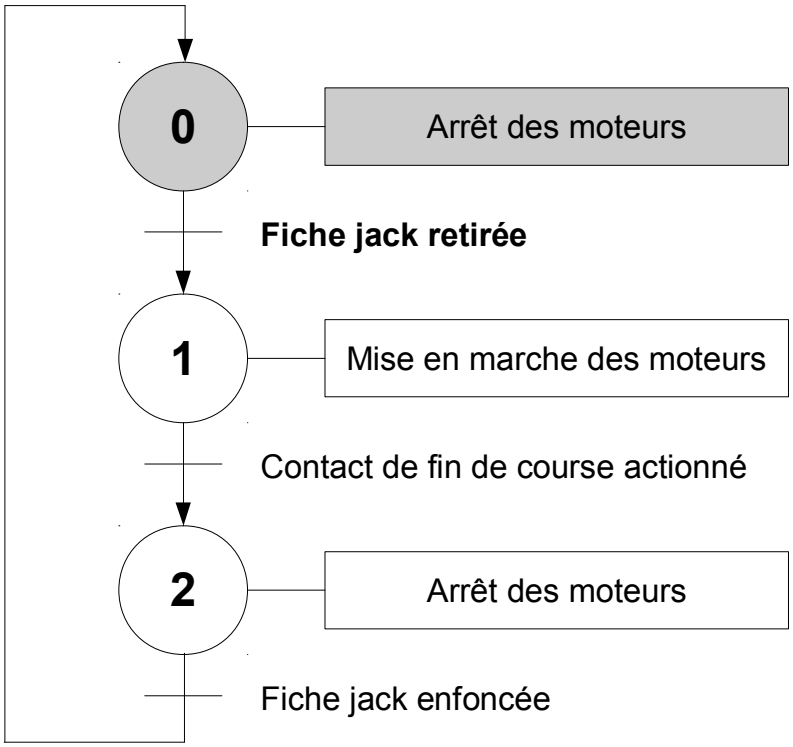
2. Représentation d'une machine de Moore

Le robot est mis sous tension.



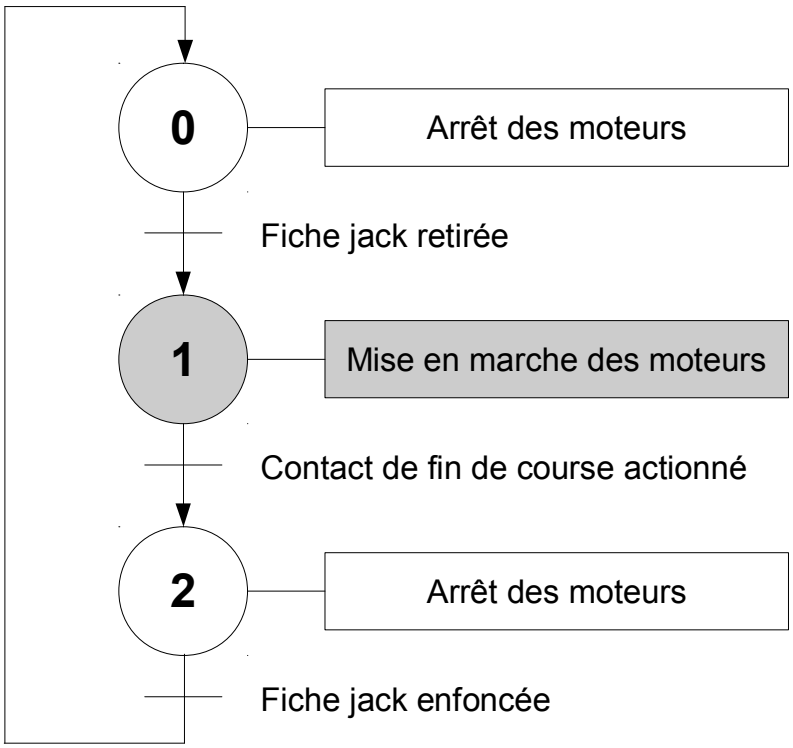
2. Représentation d'une machine de Moore

On retire la fiche jack, la transition de l'état 0 à l'état 1 devient vraie.



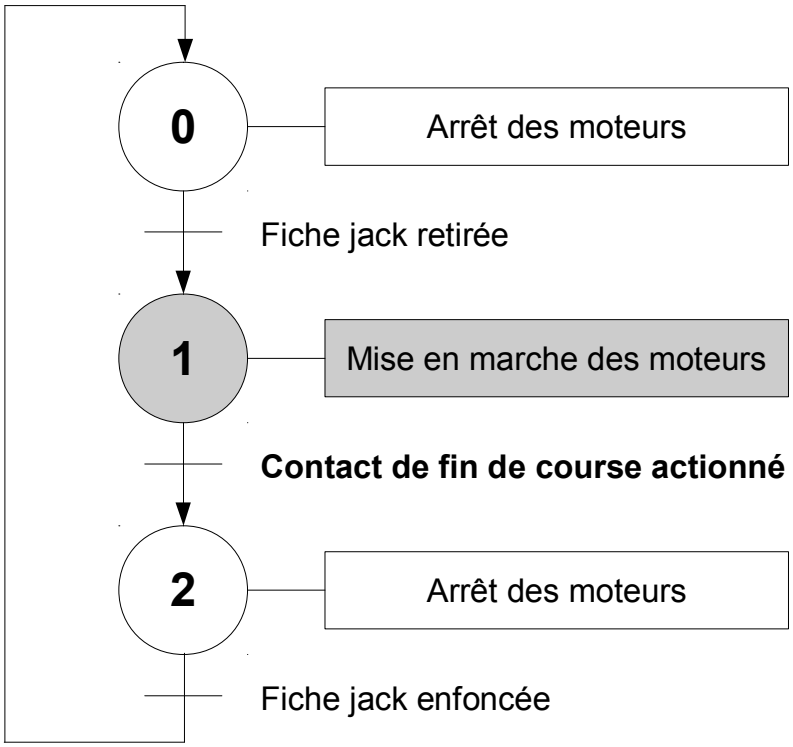
2. Représentation d'une machine de Moore

Le robot se met à rouler.



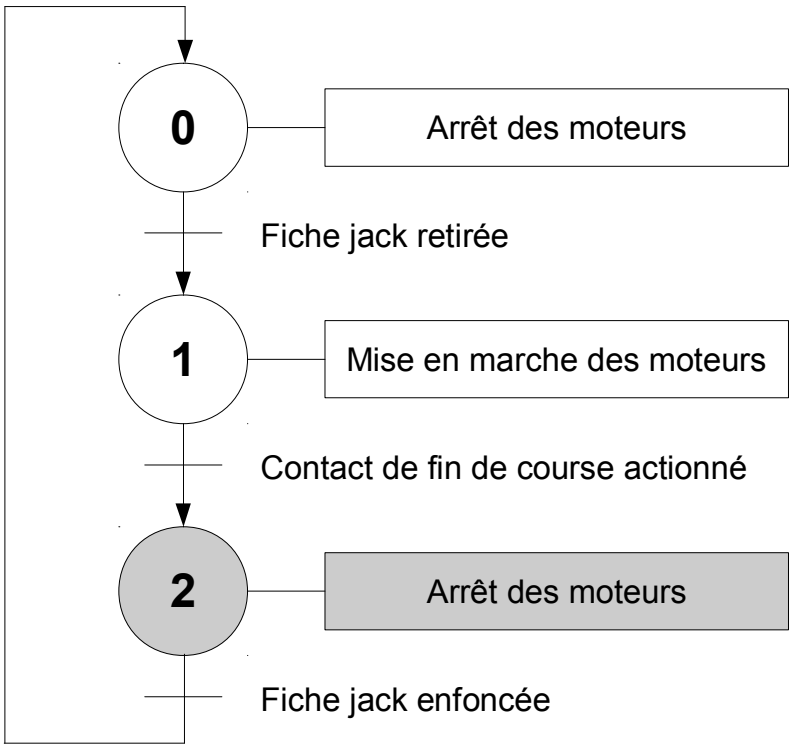
2. Représentation d'une machine de Moore

Le robot percute un obstacle, la transition de l'état 1 à l'état 2 devient vraie.



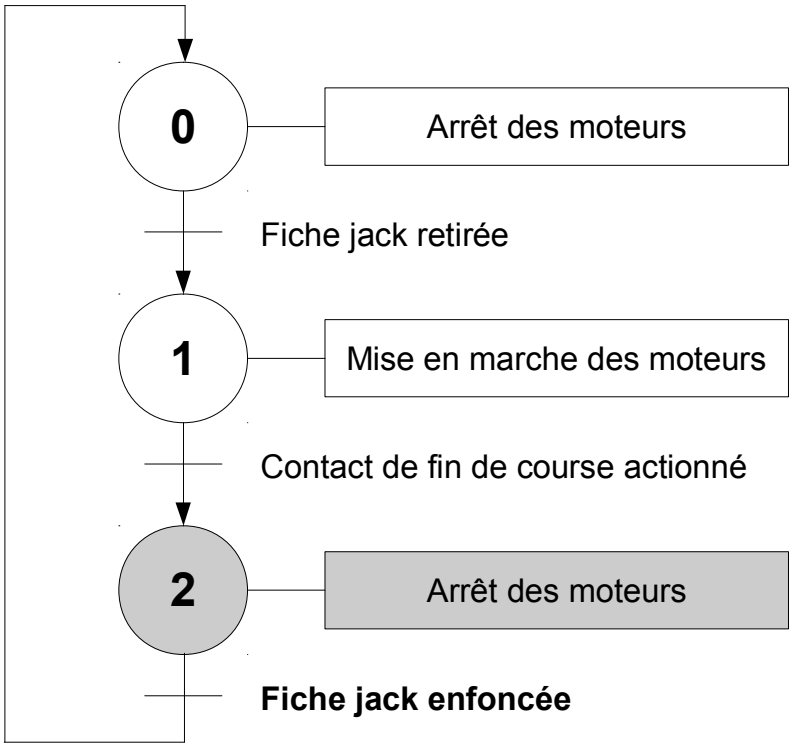
2. Représentation d'une machine de Moore

Le robot est arrêté.



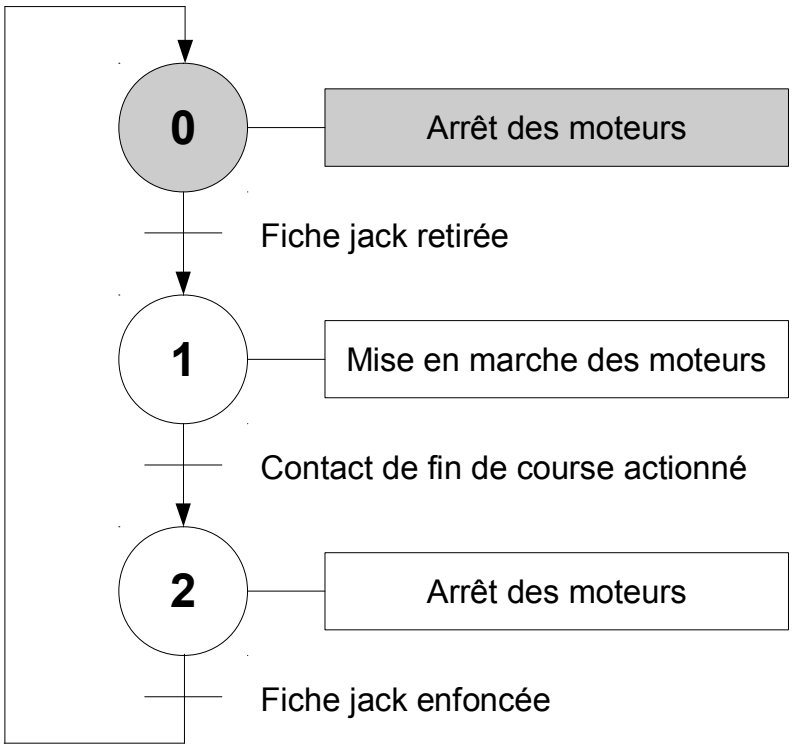
2. Représentation d'une machine de Moore

On remet la fiche jack en place, la transition de l'état 2 à l'état 0 devient vraie.



2. Représentation d'une machine de Moore

Le robot est arrêté et prêt à repartir.



Programmer un système séquentiel

1. Système séquentiel
2. Représentation d'une machine de Moore
3. **Programmation en langage C**

On définit une variable **`etat`** pour coder l'état du système.

On initialise **`etat`** à la valeur de l'état initial du système.

A l'intérieur de la boucle infinie du programme, on exécute une instruction **`switch`** pour choisir le code à exécuter en fonction de la valeur de **`etat`**.

Chaque cas traité par l'instruction **`switch`** comporte deux parties :
- le calcul des sorties du système.
- le calcul des transitions pour déterminer un changement d'état.

```
void main(void)
{
   // Définitions et initialisations des variables
   char etat = 0;
   char jack, fdc; // fiche jack et contact de fin de course

   // Configuration des périphériques
   while (1) {

      // Lecture des entrées
      // Lecture de jack et de fdc

      // Analyse et traitements
      switch (etat) {

            // détails du code sur la diapo suivante

      } // fin du switch

      // Écriture des sorties
   } // fin du while
} // fin du main
```


On suppose que **jack** vaut 0 si la fiche jack est enfoncée et 1 sinon.
On suppose que **fdc** vaut 0 si le robot percute un obstacle et 1 sinon.

```
...
switch (etat) {
   case 0 :
      // arrêt des moteurs
      if (jack != 0) etat = 1;
      break;
   case 1 :
      // mise en marche des moteurs
      if (fdc == 0) etat = 2;
      break;
   case 2 :
      // arrêt des moteurs
      if (jack == 0) etat = 0;
      break;
   default :
      // ce cas ne devrait jamais se produire
      etat = 0;
} // fin du switch
...
```