ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Programmation d'une machine d'état

Microcontrôleurs pour la commande de systèmes mécaniques

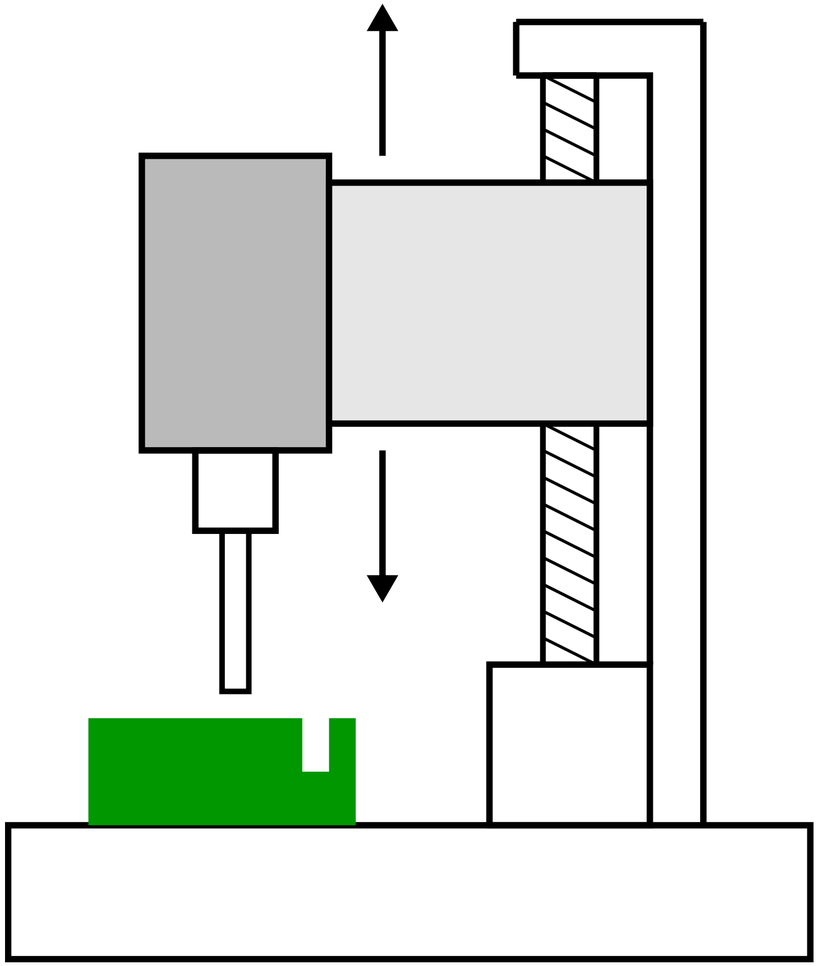
Pierre-Yves ROCHAT

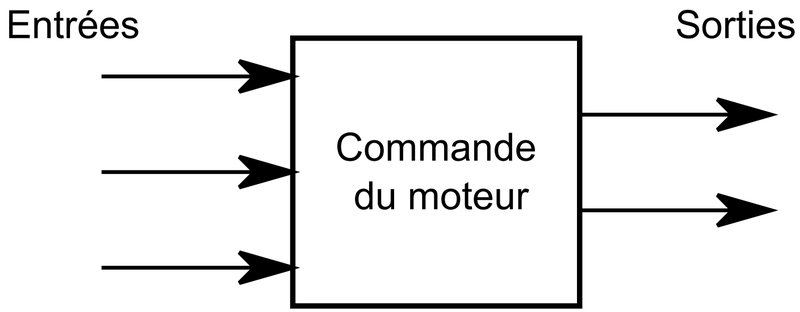
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

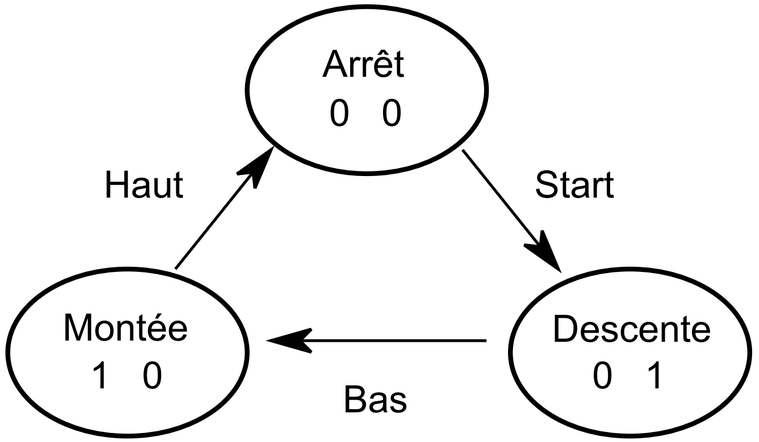
Programmation d'une machine d'état

- Rappel du problème
- Recherche d'un algorithme
- Utilisation d'une variable d'état

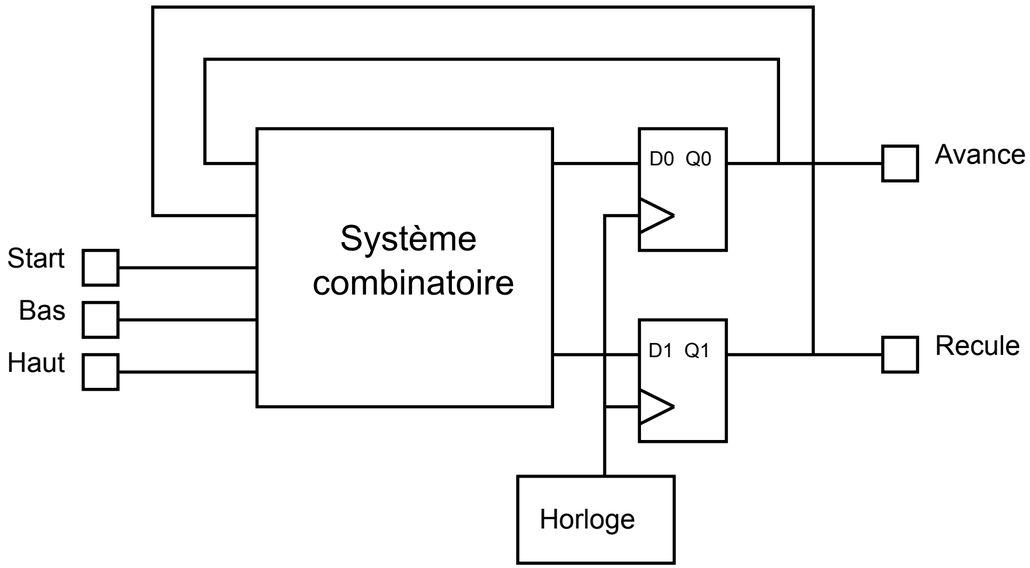
Rappel de l'exemple : la perceuse





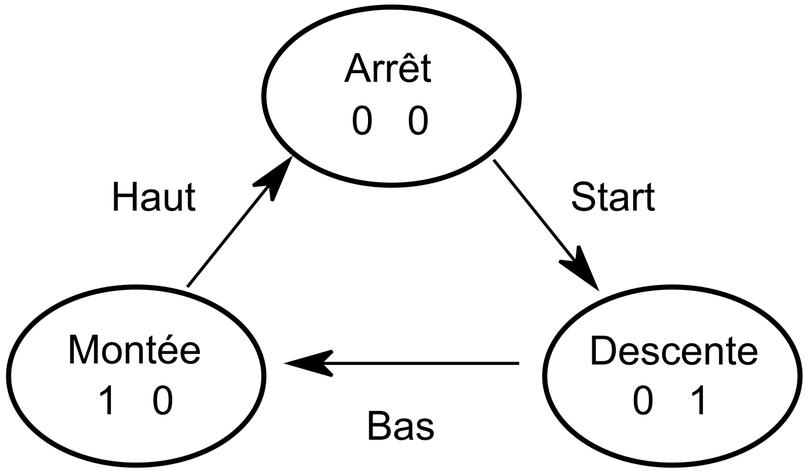


Un programme inspiré des systèmes logiques



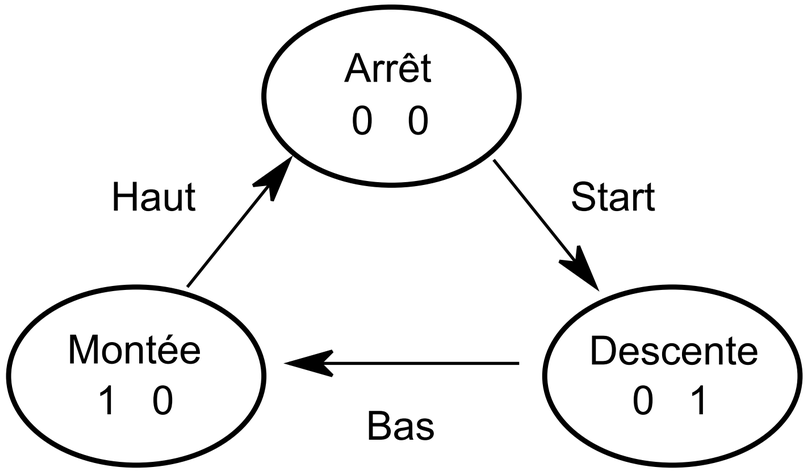

```
void loop(){
  // Lit les valeurs des entrées :
  Start = digitalRead(StartPin);
  Bas = digitalRead(BasPin);
  Haut = digitalRead(HautPin);

  // Calcule le nouvel état :
  AvancePlus = (Start && !Recule && !Avance) ||
               (!Bas && !Recule && Avance);
  ReculePlus = (Bas && !Recule && Avance) ||
               (!Haut && Recule && !Avance);

  // Enregistre le nouvel état :
  Avance = AvancePlus;
  Recule = ReculePlus;

  // Active les sorties :
  digitalWrite(AvancePin, Avance);
  digitalWrite(ReculePin, Recule);
}
```

Comment trouver l'organigramme ?

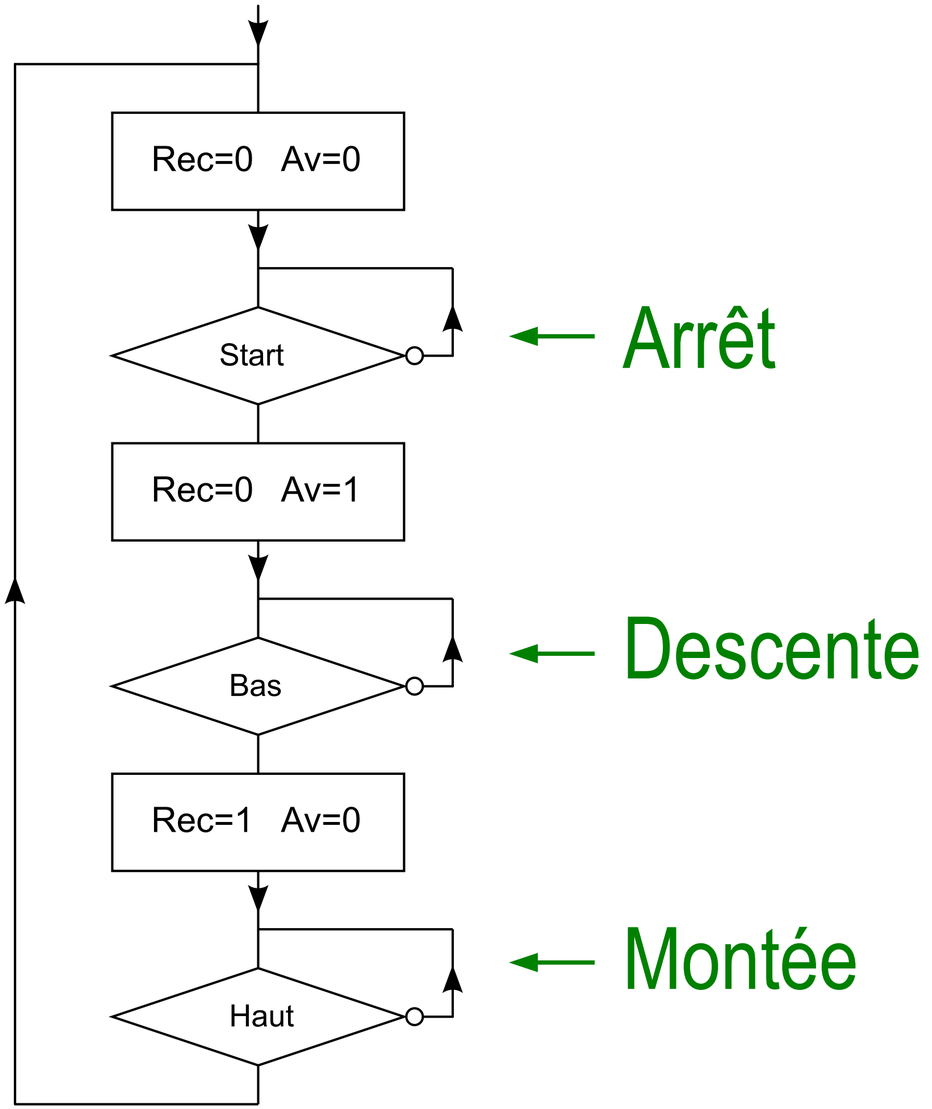


Graphe d'état ⟶ Organigramme ?

L'organigramme

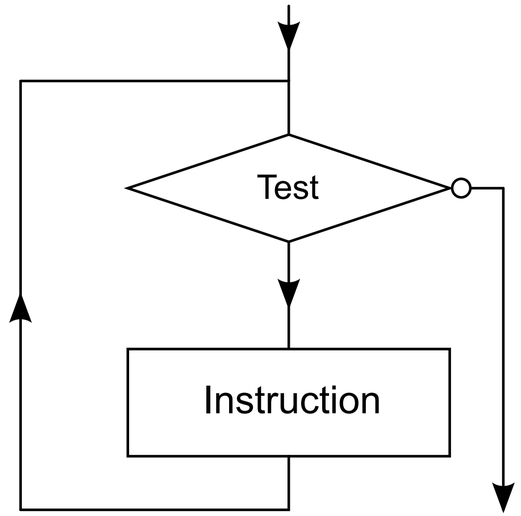


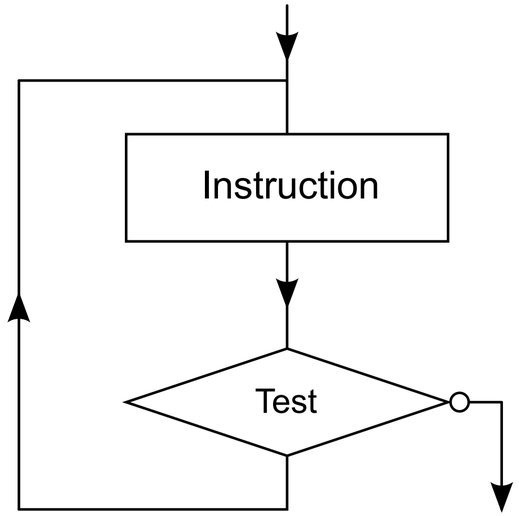
Graphe d'état

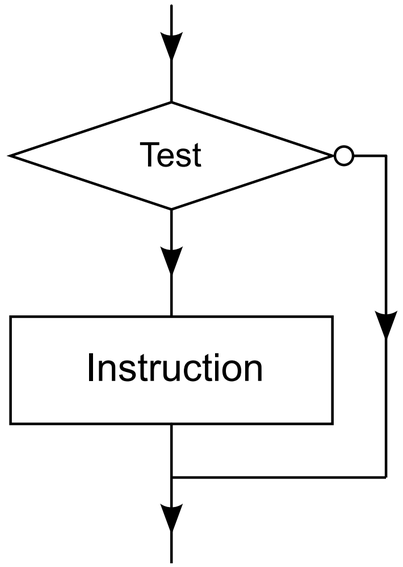


Écrire un programme en C

Structures disponibles :

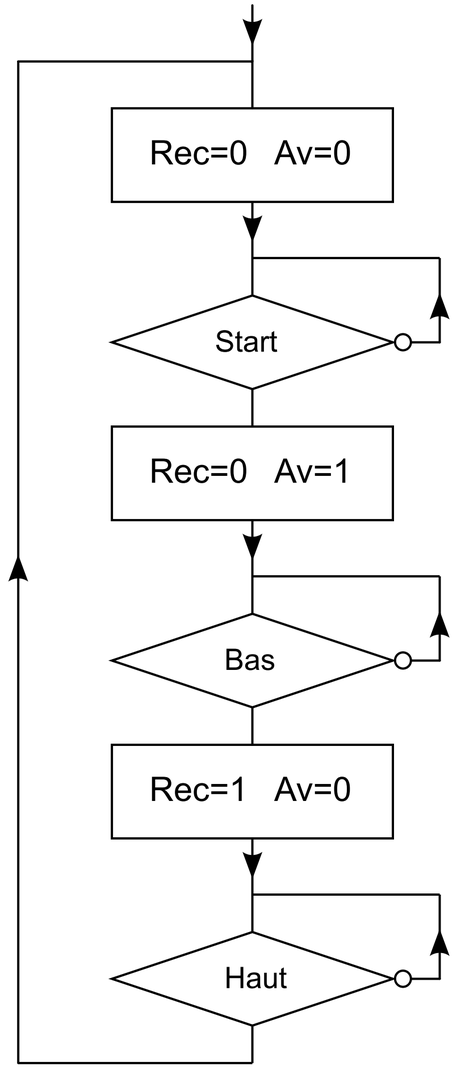






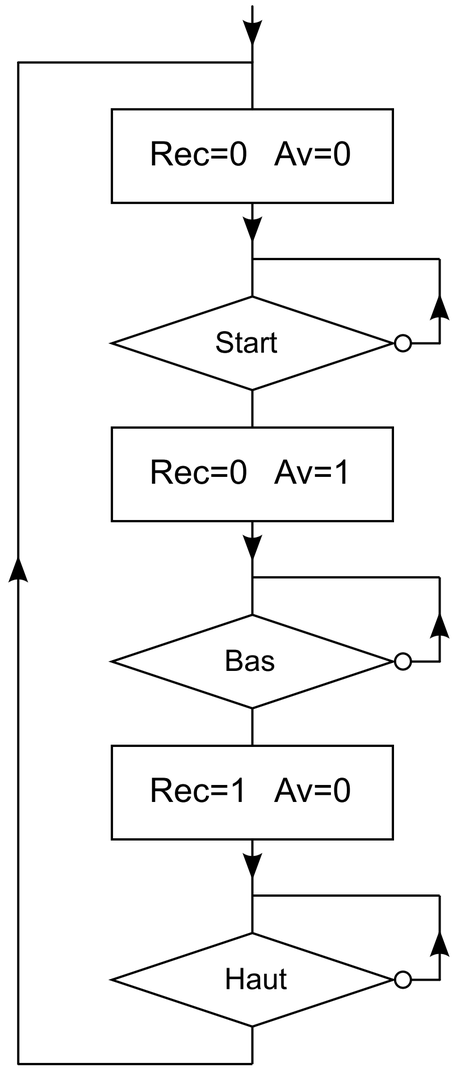
Boucles : while do...while Test : if

Passer de l'organigramme au programme



Programme en C ?

Passer de l'organigramme au programme




```
while (1) {
        Av = 0; Rec = 0;
        while  (Start) {
        }
        Av = 1; Rec = 0;
        while  (Bas) {
        }
        Av = 0; Rec = 1;
        while  (Haut) {
        }
}
```

Un programme C correct

```
while (1) {
        Av = 0; Rec = 0;
        while  (Start) {
        }
        Av = 1; Rec = 0;
        while  (Bas) {
        }
        Av = 0; Rec = 1;
        while  (Haut) {
        }
}
```

```
while (1) {
        Av(0); Rec(0);
        while  (!Start()) {
        }
        Av(1); Rec(0);
        while  (!Bas()) {
        }
        Av(0); Rec(1);
        while  (!Haut()) {
        }
}
```

Les procédures correspondantes

```
while (1) {
        Av(0); Rec(0);
        while  (!Start()) {
        }
        Av(1); Rec(0);
        while  (!Bas()) {
        }
        Av(0); Rec(1);
        while  (!Haut()) {
        }
}
```

```
#define PinAv P1_0
#define PinStart P1_3

void Av(int val) {
        digitalWrite(PinAv, val) ;
}

int Start() {
        return !digitalRead (PinStart) ;
}
```

Programme fonctionnel sous Arduino (ici Energia)

```
#define PinAv P1_0
#define PinRec P1_6
#define PinStart P1_3
#define PinBas P1_4
#define PinHaut P1_5

void Av(int val) {digitalWrite(PinAv, val);}
void Rec(int val) {digitalWrite(PinRec, val);}

int Start() {return !digitalRead(PinStart);}
int Bas() {return !digitalRead(PinBas);}
int Haut() {return !digitalRead(PinHaut);}
```

```
int main () {
  WDTCTL = WDTPW | WDTHOLD;
  pinMode(PinAv, OUTPUT);
  pinMode(PinRec, OUTPUT);
  pinMode(PinStart, INPUT_PULLUP);
  pinMode(PinBas, INPUT_PULLUP);
  pinMode(PinHaut, INPUT_PULLUP);

  while (1) {
    Av(0); Rec(0);
    while  (!Start()) {
    }
    Av(1); Rec(0);
    while  (!Bas()) {
    }
    Av(0); Rec(1);
    while  (!Haut()) {
    }
  }
}
```

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

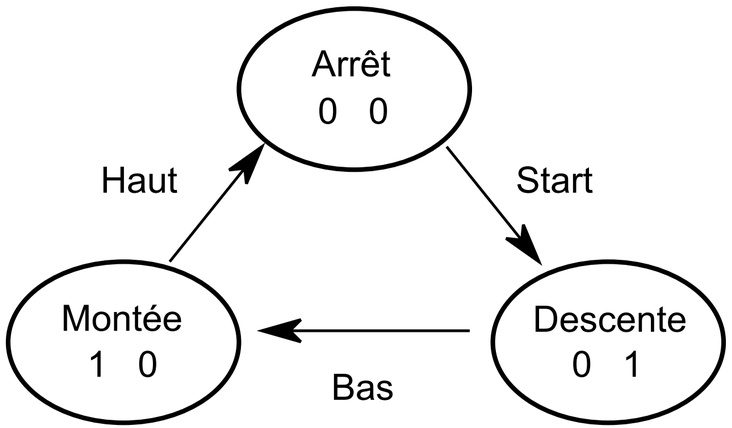
C'est difficile !

- *Pour un graphe d'état compliqué, le passage à un algorithme, puis à un programme, est fastidieux.*

- **Utilisation d'une variable d'état** :

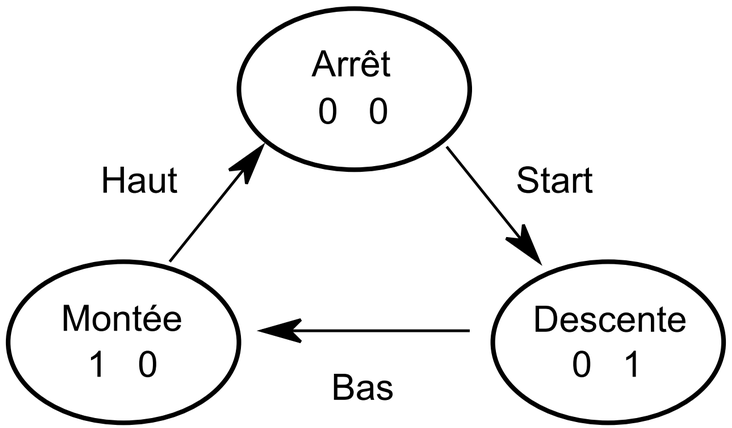
  imiter le graphe d'état !

Passer directement du graphe d'état au programme



Graphe d'état ⟶ Programme

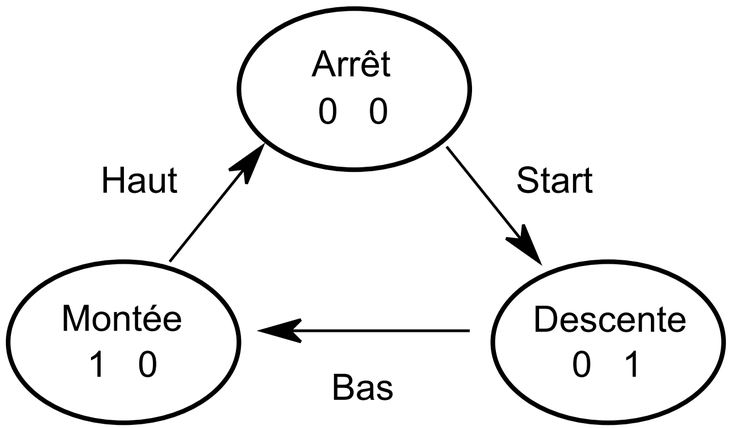
Utiliser une variable d'état



Graphe d'état

```
enum {arret, descente, montee};
...
int main() {
  ...
  int etat = arret;
  while (1) {
    if (etat == arret) ...

    if (etat == descente) ...

    if (etat == montee) ...

  }
}
```

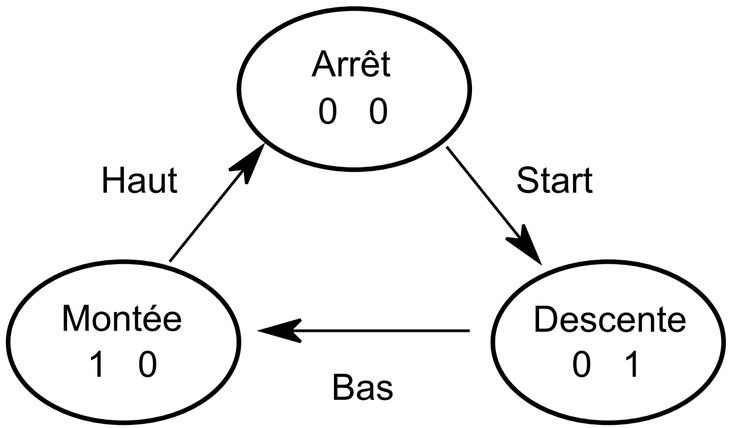
Gérer les sorties selon l'état



Graphe d'état

```
enum {arret, descente, montee};
...
int main() {
  ...
  int etat = arret;
  while (1) {
    ...
    if (etat == descente) {
      Av(0); Rec(1);


    ...
  }
}
```

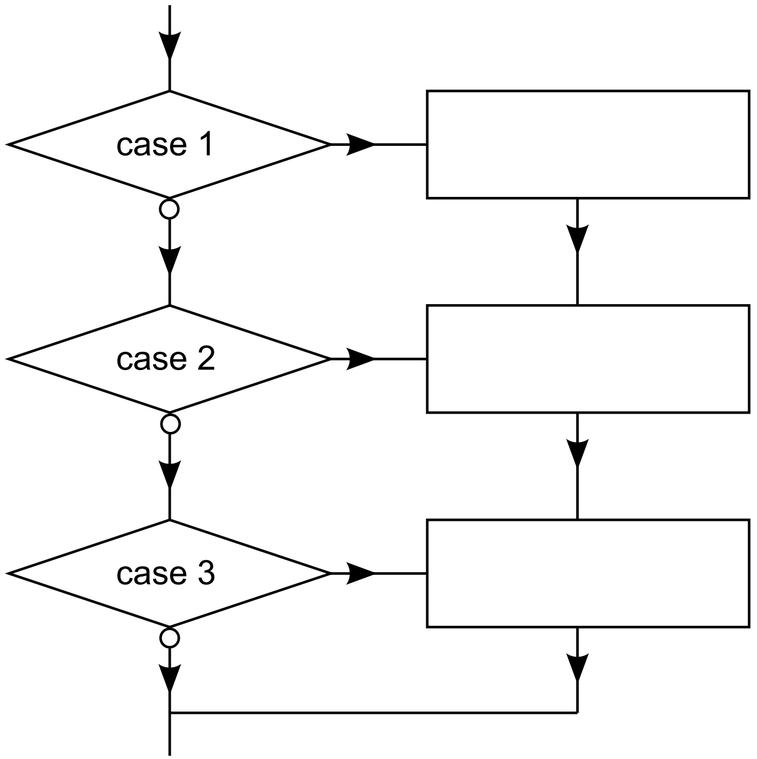
S'occuper des transitions selon l'origine de la flèche



Graphe d'état

```
enum {arret, descente, montee};
...
int main() {
  ...
  int etat = arret;
  while (1) {
   ...
    if (etat == descente) {
      Av(0) ; Rec(1) ;
      if (Bas()) {
        etat = montee ;
      } ...
  }
}
```

Utilisation de la structure switch...case




```
while (1) {
  switch (etat) {
    case arret : Av(0); Rec(0);
      if (Start()) { etat=descente; }
      break;
    case descente : Av(1); Rec(0);
      if (Bas()) { etat=montee; }
      break;
    case montee : Av(0); Rec(1);
      if (Haut()) { etat=arret; }
      break;
  }
```

break permet de sortir d'une structure

Programme fonctionnel avec Arduino (ici Energia)

```
void Av(int val) {digitalWrite(PinAv, val);}
void Rec(int val) {digitalWrite(PinRec, val);}

int Start() {return !digitalRead(PinStart);}
int Bas() {return !digitalRead(PinBas);}
int Haut() {return !digitalRead(PinHaut);}

enum {arret, descente, montee};
int main () {
  WDTCTL = WDTPW | WDTHOLD;
  pinMode(PinAv, OUTPUT);
  pinMode(PinRec, OUTPUT);
  pinMode(PinStart, INPUT_PULLUP);
  pinMode(PinBas, INPUT_PULLUP);
  pinMode(PinHaut, INPUT_PULLUP);

  int etat = arret;
  while (1) {
    switch (etat) {
      case arret : Av(0); Rec(0);
        if(Start()) { etat=descente; }
        break;
      case descente : Av(1); Rec(0);
        if(Bas()) { etat=montee; }
        break;
      case montee : Av(0); Rec(1);
        if(Haut()) { etat=arret; }
        break;
    }
  }
}
```

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Programmation d'une machine d'état

- Recherche d'un algorithme :

  pas facile...

- Utilisation d'une variable d'état :

  méthode générale