1ère partie : modélisation

2ème partie : microcontrôleurs

3ème partie : API

Mise en œuvre des MAE en langage ST

1 – CONCEPTS GÉNÉRAUX

2 – MISE EN ŒUVRE DE L'ÉVOLUTION

1 – Concepts généraux

2 – MISE EN ŒUVRE DE L'ÉVOLUTION

Principe général

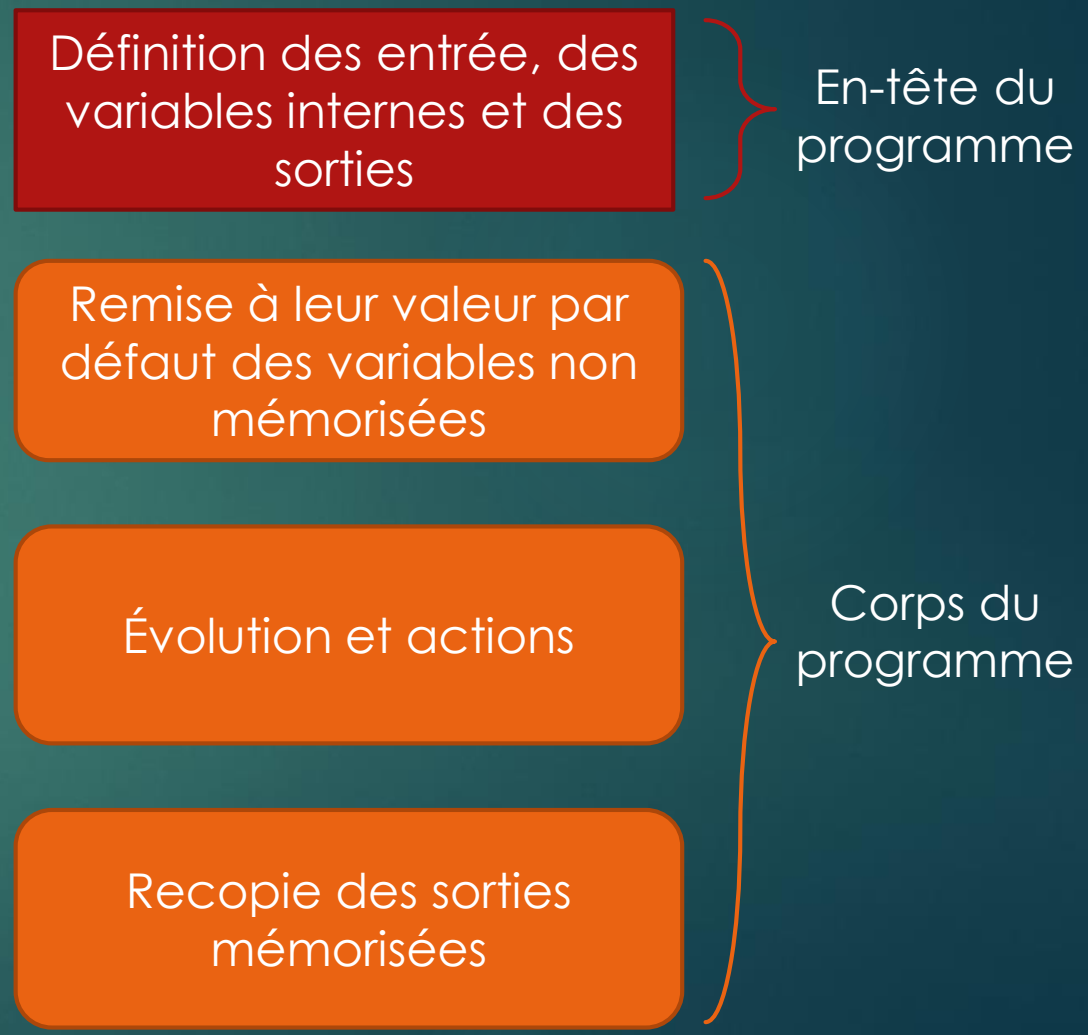
- Le langage ST est, comme son nom l'indique, un langage structuré
- Les concepts sont proches du langage C
  - Il est donc possible d'utiliser une approche similaire à celle que l'on a vue pour le langage C
- Les types de variables sont ceux déjà vus sur API
  - Les déclaration de variables utiliseront une structure similaire aux MAE en LD

Différences avec le langage C

- Il n'est pas possible de définir d'énumération
- On gèrera donc l'état :
  - Soit par un vecteur de bit : encodage one-hot comme en LD
  - Soit par une variable de type entier
    - Contrairement au LD, le travail sur les entiers est simple en langage ST
- En C, on avait une structure pour gérer les sorties
  - Cette structure, statique, permettait de réaliser des sorties mémorisées
  - Toutefois, dans les faits, il s'agissait d'une variable locale à la fonction de commande
  - On utilisera ici des variables internes (statiques) pour les sorties mémorisées, comme on l'a fait en LD

Structure du programme

- Sur API, les entrées et sorties sont gérées automatiquement
  - Pas de nécessité de fonctions « lecture des entrées » ni « écriture des sorties »
  - On aura donc un seul bloc fonctionnel, qui gèrera l'évolution et les actions
- La structure de ce bloc sera la même qu'en C :
  - On commence par remettre toutes les variables non mémorisées à leur valeur par défaut
  - Puis on gère dans un seul bloc l'évolution et les actions
  - Seule différence : les actions mémorisées étant stockées dans des variables internes au bloc, on rajoutera simplement à la fin du programme une recopie de leur valeur sur les sorties



1 – CONCEPTS GÉNÉRAUX

2 – Mise en œuvre de l'évolution

Stockage de l'état

- Pour utiliser une approche similaire au C, à base de switch, l'état doit être porté par une seule variable
  - On utilisera donc une variable statique de type entier pour enregistrer l'état courant

Évolution de la MAE

- L'équivalent du **switch** en ST est…
  - Le **case**
- Il faut donc gérer l'évolution et les actions dans ce bloc

```
CASE #"état courant" OF
    1:
       // Gestion de l'état 1
    2:
       // Gestion de l'état 2
       // etc.
   ELSE
       // Gestion du default
END_CASE;
```

Évolution de la MAE

- L'équivalent du **switch** en ST est…
  - Le **case**
- Il faut donc gérer l'évolution et les actions dans ce bloc
- Le principe reste le même qu'en C
  - Voir le cours sur la MAE en C, en adaptant la syntaxe au langage ST

```
CASE #"état courant" OF
    0:
        IF <condition> THEN
            #"action sur transition" := nouvelle valeur;
            #"état courant" := 1;
        END_IF;
    1:
        #"action sur état" := nouvelle valeur;
        IF <condition> THEN
            #"état courant" := 0;
        EN_IF;
    ELSE
        #"état courant" := 0;
END_CASE;
```

Et…

- C’est tout !

Exercice

Exercice

Réaliser le programme ST correspondant à cette MAE