



Intégrer un automate M340 ou TSX Premium dans un système Schneider Electric (Unity Pro)

- 07 - Unity Pro: le traitement du programme

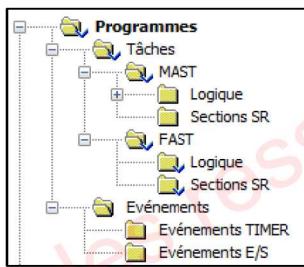
Unity Pro: le traitement du programme

1. Description des fonctions disponibles	3
2. Description des tâches et des traitements	4
□ La tâche maître (MAST)	
□ La tâche rapide (FAST)	
□ Les tâches auxiliaires (AUX)	
□ Le Traitement Evènementiel (EVT)	
3. Description des sections et des sous-programmes	9
□ Description des sections	
□ Description des sous-programmes	
4. Fonctionnement d'une application monotâche	11
□ Description du cycle de la tâche maître (MAST)	
□ Exécution cyclique d'une application monotâche	
□ Exécution périodique d'une application monotâche	
□ Contrôle de la durée du cycle (chien de garde)	
5. Fonctionnement d'une application multitâche	15
□ Structure logicielle multitâche	
□ Les niveaux de priorité	
□ Séquencement des tâches dans une structure multitâche	
□ Contrôle des tâches	
□ Gestion des Traitements Evènementiels	
□ Exécution du traitement événementiel de type TIMER	
□ Echanges d'entrées/de sorties dans les traitements événementiels	
6. Modes de fonctionnement des automates Modicon M340	23
□ Traitement en cas de DÉMARRAGE à FROID des automates Modicon M340	
□ Traitement sur REPRISE à CHAUD pour les automates Modicon M340	

Unity Pro: le traitement du programme

1. Description des fonctions disponibles

- ✓ Le programme application est constitué de plusieurs tâches:
 - la tâche maître cyclique ou périodique constituée de sections et de sous-programmes,
 - la tâche rapide périodique,
 - les tâches auxiliaires périodique,
 - le traitement des événements de type E/S,
 - Le traitement des événements de type Temporisateur (TIMER),
 - le traitement des événements de type E/S et Temporisateur.



Unity Pro: le traitement du programme

2. Description des tâches et des traitements

La tâche maître (MAST)

- ✓ La tâche maître représente la **principale tâche** du programme de l'application. Elle est **obligatoire** et créée par défaut.
- ✓ **Structure de la tâche maître**
 - La tâche maître (MAST) est constituée de sections et de sous-programmes.
 - Chaque section de la tâche maître est programmée dans les langages suivants : LD, FBD, IL, ST ou SFC.
 - Le SFC ne peut être utilisé que dans les sections de tâche maître. Le nombre de sections programmées en SFC est illimité
 - Les sous-programmes sont programmés en LD, FBD, IL ou ST et sont appelés dans les sections de tâche.
- ✓ **Exécution de la tâche maître**
 - L'exécution de la tâche maître peut être choisie :
 - ❖ cyclique (choix par défaut)
 - ❖ ou périodique (1 à 255ms)
- ✓ **Contrôle de la tâche maître**
 - La tâche maître peut être contrôlée par programme, par des bits et des mots système.

Objets système	Signification
%SW0	Période de tâche
%S30	Activation de la tâche maître
%S11	Erreur du chien de garde
%S19	Dépassement de période
%SW27	Nombre de ms passées dans le système pendant le dernier cycle Mast.
%SW28	Temps système le plus long (en ms).
%SW29	Temps système le plus court (en ms).
%SW30	Temps d'exécution (en ms) du dernier cycle.
%SW31	Temps d'exécution (en ms) du cycle le plus long.
%SW32	Temps d'exécution (en ms) du cycle le plus court.

Unity Pro: le traitement du programme

2. Description des tâches et des traitements

- **La tâche rapide (FAST)**
 - ✓ La tâche rapide est destinée aux **traitements de courte durée et périodiques**.
 - ✓ **Structure de la tâche rapide**
 - La tâche rapide (FAST) est constituée de sections, et de sous-programmes.
 - Chaque section de la tâche rapide se programme dans un des langages : LD, FBD, IL ou ST.
 - Le langage SFC n'est pas utilisable dans les sections d'une tâche rapide.
 - Les sous-programmes se programmant en langage LD, FBD, IL ou ST, ils sont appelés dans les sections de la tâche.
 - ✓ **Exécution de la tâche rapide**
 - L'exécution de la tâche rapide est périodique.
 - Elle est **plus prioritaire** que la tâche maître.
 - La période de la tâche rapide (FAST) est fixée par configuration, de 1 à 255 ms.
 - Le programme exécuté doit cependant rester court pour éviter le débordement des tâches moins prioritaires.
 - ✓ **Contrôle de la tâche rapide**
 - La tâche rapide peut être contrôlée par programme par bits et mots système.

Objets système	Signification
%SW1	Période de tâche.
%S31	Activation de la tâche rapide.
%S11	Erreur du chien de garde
%S19	Dépassement période.
%SW33	Temps d'exécution (en ms) du dernier cycle.
%SW34	Temps d'exécution (en ms) du cycle le plus long.
%SW35	Temps d'exécution (en ms) du cycle le plus court.

Unity Pro: le traitement du programme

2. Description des tâches et des traitements

Les tâches auxiliaires (AUX)

- ✓ Les tâches auxiliaires sont conçues pour les **tâches de traitement lentes**. Il s'agit des tâches de plus faible priorité.
- ✓ Il est possible de programmer jusqu'à quatre tâches auxiliaires (AUX0, AUX1, AUX2 ou AUX3) sur les automates Premium TSX P57 5•• et Quantum 140 CPU 6••••.
- ✓ Il est possible de programmer jusqu'à deux tâches auxiliaires (AUX0, AUX1) sur les automates Modicon M580 BME P58 ••••.
- ✓ Les tâches auxiliaires ne sont pas disponibles pour les automates Modicon M340.
- ✓ **Structure des tâches auxiliaires**
 - Les tâches auxiliaires (AUX) sont composées de sections et de sous-programmes.
 - Chaque section de la tâche auxiliaire est programmée dans l'un des langages suivants : LD, FBD, IL ou ST.
 - Le langage SFC n'est pas utilisable dans les sections d'une tâche auxiliaire.
 - Un maximum de 64 sous-programmes peuvent être programmés en langage LD, FBD, IL ou ST. Ils sont appelés dans les sections des tâches.
- ✓ **Exécution des tâches auxiliaires**
 - L'exécution des tâches auxiliaires est périodique.
 - Elles ont la priorité la plus faible.
 - La période d'une tâche auxiliaire peut être fixée entre 10ms et 2,55s.
- ✓ **Contrôle de la tâche rapide**
 - Les tâches auxiliaires peuvent être contrôlées par programme, par des bits et mots système.

Objets système	Signification
%SW2	Période de la tâche auxiliaire 0
%SW3	Période de la tâche auxiliaire 1
%SW4	Période de la tâche auxiliaire 2
%SW5	Période de la tâche auxiliaire 3
%S32	Activation de la tâche auxiliaire 0
%S33	Activation de la tâche auxiliaire 1
%S34	Activation de la tâche auxiliaire 2
%S35	Activation de la tâche auxiliaire 3
%S11	Erreur du chien de garde
%S19	Dépassement de période
%SW36 à %SW47	Temps d'exécution (en ms) des cycles le plus court et le plus long de chaque tâche auxiliaire

Unity Pro: le traitement du programme

2. Description des tâches et des traitements

Les tâches auxiliaires (AUX)

- ✓ Les tâches auxiliaires sont conçues pour les **tâches de traitement lentes**. Il s'agit des tâches de plus faible priorité.
- ✓ Il est possible de programmer jusqu'à quatre tâches auxiliaires (AUX0, AUX1, AUX2 ou AUX3) sur les automates Premium TSX P57 5•• et Quantum 140 CPU 6••••.
- ✓ Il est possible de programmer jusqu'à deux tâches auxiliaires (AUX0, AUX1) sur les automates Modicon M580 BME P58 ••••.
- ✓ Les tâches auxiliaires ne sont pas disponibles pour les automates Modicon M340.
- ✓ **Structure des tâches auxiliaires**
 - Les tâches auxiliaires (AUX) sont composées de sections et de sous-programmes.
 - Chaque section de la tâche auxiliaire est programmée dans l'un des langages suivants : LD, FBD, IL ou ST.
 - Le langage SFC n'est pas utilisable dans les sections d'une tâche auxiliaire.
 - Un maximum de 64 sous-programmes peuvent être programmés en langage LD, FBD, IL ou ST. Ils sont appelés dans les sections des tâches.
- ✓ **Exécution des tâches auxiliaires**
 - L'exécution des tâches auxiliaires est périodique.
 - Elles ont la priorité la plus faible.
 - La période d'une tâche auxiliaire peut être fixée entre 10ms et 2,55s.
- ✓ **Contrôle de la tâche rapide**
 - Les tâches auxiliaires peuvent être contrôlées par programme, par des bits et mots système.

Objets système	Signification
%SW2	Période de la tâche auxiliaire 0
%SW3	Période de la tâche auxiliaire 1
%SW4	Période de la tâche auxiliaire 2
%SW5	Période de la tâche auxiliaire 3
%S32	Activation de la tâche auxiliaire 0
%S33	Activation de la tâche auxiliaire 1
%S34	Activation de la tâche auxiliaire 2
%S35	Activation de la tâche auxiliaire 3
%S11	Erreur du chien de garde
%S19	Dépassement de période
%SW36 à %SW47	Temps d'exécution (en ms) des cycles le plus court et le plus long de chaque tâche auxiliaire

Unity Pro: le traitement du programme

2. Description des tâches et des traitements

Le Traitement Évènementiel (EVT)

- ✓ Le traitement évènementiel est utilisé pour réduire le temps de réponse des programmes d'application aux événements :
 - provenant de modules d'entrée/sortie,
 - de temporiseurs d'évènements.
- ✓ Ces tâches de traitement ont la **priorité sur toute autre tâche** pour leur exécution. Elles sont donc adaptées aux tâches de traitement nécessitant un temps de réponse très court en rapport avec l'évènement.
- ✓ Le nombre de tâches évènementielles pouvant être programmées dépend du type de processeur.
- ✓ **Structure d'une tâche évènementielle**
 - Une tâche évènementielle est monosection, et n'est constituée que d'une seule section (non conditionnée).
 - Elle est programmée au choix en langage LD, FBD, IL ou ST.
 - Deux types d'évènements sont disponibles :
 - ❖ **Évènement E/S** : pour les évènements issus de modules d'entrée/sortie
 - ❖ **Évènements TIMER** : pour les évènements issus de temporiseurs d'évènements.
- ✓ **Exécution d'une tâche évènementielle**
 - L'exécution d'une tâche évènementielle est asynchrone.
 - La survenue d'un évènement redirige le programme d'application vers la tâche de traitement associée à la voie d'entrée/sortie ou au temporisateur d'évènement qui a produit l'évènement
- ✓ **Contrôle d'une tâche évènementielle**
 - Les bits et mots système suivants peuvent être utilisés pour contrôler les tâches évènementielles au cours de l'exécution d'un programme.

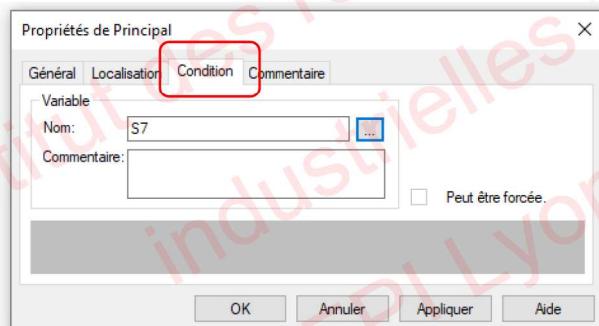
Objets système	Signification
%S38	Activation du traitement évènementiel.
%S39	Saturation de la pile de gestion d'appel d'évènements.
%SW48	Nombre de tâches de traitement d'évènements E/S et de télégrammes Exécutées (Les TELEGRAMMES ne sont disponibles que pour les PREMIUM).
%SW75	Nombre d'évènements de type Timer dans la file d'attente.

Unity Pro: le traitement du programme

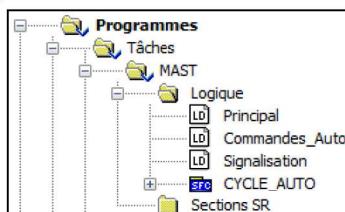
3. Description des sections et des sous-programmes

□ Description des sections

- ✓ Les sections sont des entités autonomes de programmation.
- ✓ Les étiquettes de repérage des lignes d'instructions, des réseaux de contacts ... sont propres à la section (pas de saut de programme possible vers une autre section).
- ✓ Elles se programment soit en :
 - langage à contacts LD,
 - langage en blocs fonctionnels FBD,
 - Liste d'instructions IL,
 - littéral structuré ST,
 - diagramme fonctionnel en séquence SFC, sous réserve que le langage soit accepté dans la tâche.
- ✓ Les sections sont exécutées dans leur ordre de programmation dans la fenêtre du navigateur (vue structurelle).
- ✓ Une **condition d'exécution** peut être associée à une ou plusieurs sections dans les tâches maître, rapides et auxiliaires, mais pas dans les traitements événementiels.



- ✓ Elles sont reliées à une tâche. Une même section ne peut pas appartenir à plusieurs tâches en même temps.



Unity Pro: le traitement du programme

3. Description des sections et des sous-programmes

Description des sous-programmes

- ✓ Les sous-programmes sont programmés sous forme d'entités distinctes, dans l'un des langages suivants :
 - Schéma à contacts (LD),
 - Langage en blocs fonction (FBD)
 - Liste d'instructions (IL)
 - Langage littéral structuré (ST)
- ✓ Un sous-programme est appelé depuis les sections ou un autre sous-programme.
- ✓ Un sous-programme ne peut pas s'appeler lui-même (non récursif)
- ✓ Les sous-programmes sont également liés à une tâche. Le même sous-programme ne peut pas être appelé à partir de différentes tâches.

Caractéristiques d'un sous-programme:

Objets système	Signification
Nom	32 caractères maximum (les accents sont autorisés, mais les espaces sont interdits).
Langage	LD, FBD, IL ou ST.
Tâche	maître (MAST) rapide (FAST) ou auxiliaire (AUX)
Commentaire	512 caractères maximum

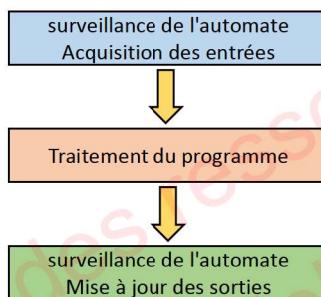
Unity Pro: le traitement du programme

4. Fonctionnement d'une application monotâche

□ Description du cycle de la tâche maître (MAST)

- ✓ Le programme d'une application monotâche est associé à une seule tâche utilisateur, la tâche maître.
- ✓ L'exécution de la tâche maître peut être choisie :
 - cyclique
 - Périodique

✓ Le cycle de scrutation lorsque l'automate est en RUN



➤ Acquisition des entrées

- ❖ Ecriture en mémoire de l'état des informations présentes sur les entrées des modules TOR et métier associés à la tâche.
- ❖ Ces valeurs peuvent être modifiées par les valeurs de forçage.

➤ Traitement du programme

- ❖ Exécution du programme d'application, écrit par l'utilisateur

➤ Mise à jour des sorties

- ❖ Ecriture des bits ou des mots de sortie sur les modules TOR et métier associés à la tâche, selon l'état défini par l'application.
- ❖ Comme pour les entrées, l'écriture des sorties peut être modifiée par des valeurs de forçage.

➤ Surveillance de l'automate

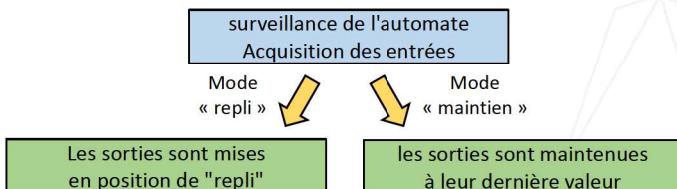
- ❖ Durant les phases d'acquisition des entrées et de mise à jour des sorties, le système réalise aussi implicitement la surveillance de l'automate (gestion des bits et mots système, mise à jour des valeurs courantes de l'horodateur, mise à jour des voyants d'état (DEL), détection des passages RUN/STOP, ...) et le traitement des requêtes en provenance du terminal (modifications et animation).

Unity Pro: le traitement du programme

4. Fonctionnement d'une application monotâche

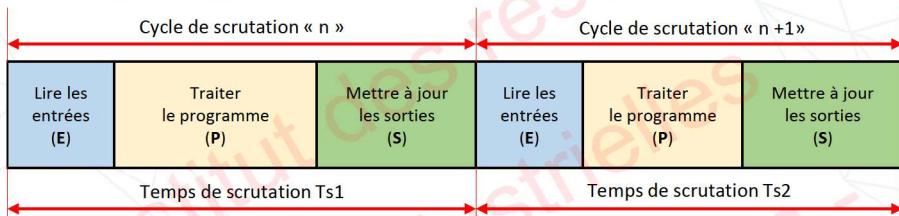
□ Description du cycle de la tâche maître (MAST)

- ✓ Le cycle de scrutation lorsque l'automate est en STOP



□ Exécution cyclique d'une application monotâche

- ✓ Fonctionnement:



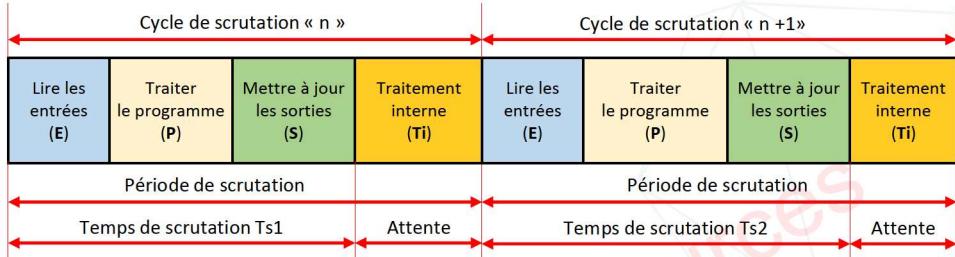
- Ce type de fonctionnement consiste à enchaîner les cycles de tâche les uns après les autres.
- Après la mise à jour des sorties, le système effectue un propre traitement et commence un autre cycle de tâche, sans s'arrêter.
- La durée d'un cycle est contrôlée par l'automate (chien de garde) et ne doit pas excéder la valeur définie dans la configuration Tmax (1500 ms par défaut, 1,5 s maximum).

Unity Pro: le traitement du programme

4. Fonctionnement d'une application monotâche

❑ Exécution périodique d'une application monotâche

✓ Fonctionnement:



- Dans ce mode de marche, l'acquisition des entrées, le traitement du programme d'application et la mise à jour des sorties s'effectuent de façon périodique, selon une fréquence définie comprise entre 1 et 255 ms.
- En début de cycle automatique, une température dont la valeur courante est initialisée sur la fréquence définie, démarre (compte à rebours).
- Le cycle automatique doit se terminer avant l'expiration de cette température qui relance un nouveau cycle.

✓ Mode de marche:

- Le processeur effectue dans l'ordre le traitement interne, l'acquisition des entrées, le traitement du programme d'application et la mise à jour des sorties.
 - ❖ Si la période n'est pas encore terminée, le processeur complète son cycle de fonctionnement jusqu'à la fin de la période par du traitement interne.
 - ❖ Si le temps de fonctionnement devient supérieur à celui affecté à la période, l'automate signale un dépassement de période par la mise à 1 du bit système %S19 de la tâche. Le traitement se poursuit et est exécuté dans sa totalité (il ne doit pas dépasser néanmoins le temps limite du chien de garde). Le cycle suivant est enchaîné après l'écriture implicite des sorties du cycle en cours.

Unity Pro: le traitement du programme

4. Fonctionnement d'une application monotâche

Contrôle de la durée du cycle (chien de garde)

✓ Généralités:

- La durée de l'exécution de la tâche maître, en fonctionnement cyclique ou périodique, est contrôlée par l'automate (chien de garde) et ne doit pas excéder la valeur définie dans la configuration **Tmax** (1500 ms par défaut, 1,5 s maximum).

✓ Chien de garde logiciel (fonctionnement périodique ou cyclique)

- Si le dépassement du chien de garde se produit, l'application est déclarée en erreur, ce qui entraîne l'arrêt immédiat de l'automate (état **HALT**).
- Le bit **%S11** indique que le chien de garde est dépassé. Il est mis à l'état 1 par le système lorsque la durée de cycle est supérieure au chien de garde.
- Le mot **%SW11** contient la valeur du chien de garde en ms. Cette valeur n'est pas modifiable par le programme.
- La réactivation de la tâche requiert la connexion du terminal afin d'analyser la cause de l'erreur, la corriger, réinitialiser l'automate et le faire passer sur RUN.
- Il n'est pas possible de quitter HALT en basculant vers STOP. Pour ce faire, vous devez réinitialiser l'application pour vérifier la cohérence des données.

✓ Contrôle en fonctionnement périodique

- En fonctionnement périodique, un contrôle supplémentaire permet la détection d'un dépassement de la période. Un dépassement de la période n'entraîne pas l'arrêt de l'automate tant qu'il reste inférieur à la valeur du chien de garde.
- Le bit **%S19** indique que la période est dépassée. Il est mis à l'état 1 par le système lorsque la durée de cycle est supérieure à la période de la tâche.
- Le mot **%SW0** contient la valeur de la période (en ms). Il est initialisé lors d'un redémarrage à froid par la valeur définie. Il peut être modifié par l'utilisateur.

✓ Exploitation des temps d'exécution de tâche maître

- Les mots système suivants peuvent être utilisés pour obtenir des informations sur la durée de cycle:
 - ❖ **%SW30** contient le temps d'exécution du dernier cycle
 - ❖ **%SW31** contient le temps d'exécution du cycle le plus long
 - ❖ **%SW32** contient le temps d'exécution du cycle le plus court

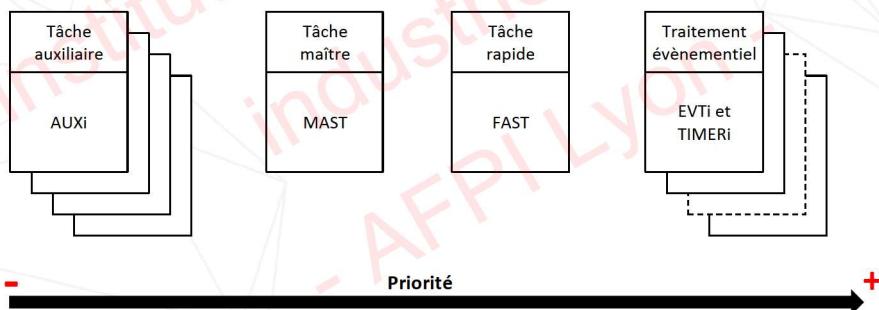
Unity Pro: le traitement du programme

5. Fonctionnement d'une application multitâche

Structure logicielle multitâche

Tâches ou traitements	Désignation	Description
Maître	MAST	<ul style="list-style-type: none">Toujours présent, peut être défini sur cyclique ou périodique.
Rapide	FAST	<ul style="list-style-type: none">Facultatif, toujours périodique.
Auxiliaires	AUX 0 à 3	<ul style="list-style-type: none">Facultatif et toujours périodique.
Evénements	EVTi - TIMERi	<ul style="list-style-type: none">Appelé par le système lorsqu'un événement se produit sur un module d'entrées/sorties ou est déclenché par l'événement TIMER.Ces types de traitements sont facultatifs et peuvent être utilisés par des applications devant agir sur les entrées et sorties dans un délai très court.

Les niveaux de priorité



- ✓ La tâche maître (MAST) est toujours l'application de base. Les autres tâches varient en fonction du type d'automate.
- ✓ Des niveaux de priorité sont déterminés pour chaque tâche de façon à établir un ordre d'importance pour les types de traitements.
- ✓ Le traitement événementiel peut être activé de façon asynchrone par rapport aux tâches périodiques, dans un ordre généré par des événements externes. Celui-ci est traité en priorité et nécessite l'interruption de tout autre traitement en cours.

Unity Pro: le traitement du programme

5. Fonctionnement d'une application multitâche

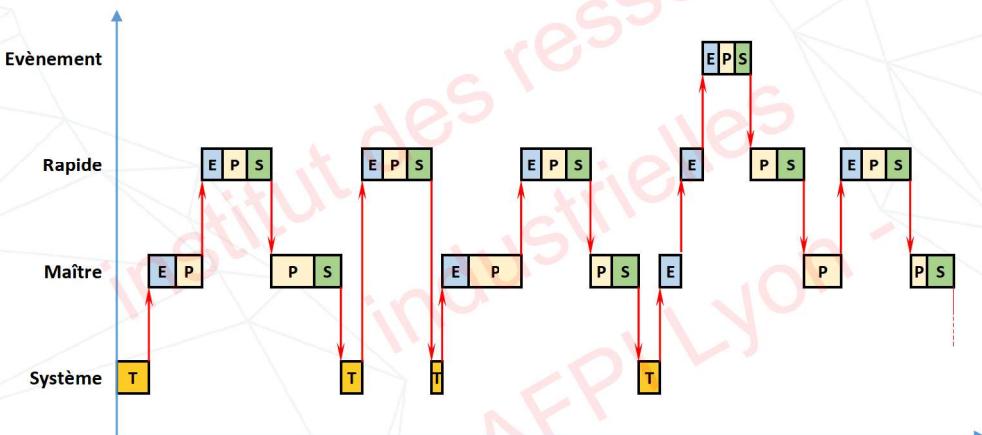
□ Séquencement des tâches dans une structure multitâche

✓ Fonctionnement

- La tâche maître est active par défaut.
- Les tâches rapide et auxiliaires sont actives par défaut si elles sont programmées.
- Le traitement événementiel est activé lors d'apparition de l'événement qui lui a été associé.

✓ Description du séquencement des tâches

- Exemple: séquencement des tâches d'un traitement multitâche comportant:
 - ❖ une tâche maître cyclique,
 - ❖ une tâche rapide de période 20 ms,
 - ❖ un traitement événementiel.



- ❖ T: traitement interne
- ❖ E: acquisition des entrées
- ❖ P: traitement du programme
- ❖ S: mise à jour des sorties

Unity Pro: le traitement du programme

5. Fonctionnement d'une application multitâche

Contrôle des tâches

Fonctionnement cyclique ou périodique

- En fonctionnement multitâche, la tâche avec la priorité la plus élevée devra être utilisée en mode périodique pour permettre aux tâches avec la priorité la plus basse de s'exécuter.
- Pour cette raison, seule la tâche avec la priorité la plus faible devrait être utilisée en mode cyclique. Par conséquent, choisir le mode de fonctionnement cyclique pour la tâche maître exclut l'utilisation de tâches auxiliaires.

Mesure des durées des tâches

- La durée des tâches est mesurée en continu. Cette mesure représente la durée entre le démarrage et la fin de l'exécution de la tâche. Elle inclut le temps mis par les tâches de priorité de niveau le plus élevé qui peuvent interrompre l'exécution de la tâche mesurée.
- Les mots système suivants donnent les durées des cycles actuels, maximum et minimum pour chaque tâche (valeur en ms)

Mesure des durées	MAST	FAST	AUX0	AUX1	AUX2	AUX3
Courant	%SW30	%SW33	%SW36	%SW39	%SW42	%SW45
Maximum	%SW31	%SW34	%SW37	%SW40	%SW43	%SW46
Minimum	%SW32	%SW35	%SW38	%SW41	%SW44	%SW47

- les durées maximum et minimum sont prises à partir des durées mesurées depuis le dernier redémarrage à froid.

Périodes de tâche

- Les périodes sont définies dans les propriétés de la tâche. Elles peuvent être modifiées par les mots système suivants.

Mots système	Tâche	Valeurs	Valeurs par défaut	Remarques
%SW0	MAST	0-255 ms	Cyclique	0 = en fonctionnement cyclique
%SW1	FAST	1-255 ms	5 ms	-
%SW2	AUX0	10 ms-2,55 s	100 ms	Les valeurs de la période sont exprimées en 10 ms.
%SW3	AUX1	10 ms-2,55 s	200 ms	
%SW4	AUX2	10 ms-2,55 s	300 ms	
%SW5	AUX3	10 ms-2,55 s	400 ms	

Unity Pro: le traitement du programme

5. Fonctionnement d'une application multitâche

Contrôle des tâches

Périodes de tâche

- Lorsque la durée de cycle de la tâche dépasse la période, le système met à l'état 1 le bit système **%S19** et continue avec le cycle suivant.
- les valeurs des périodes ne dépendent pas de la priorité des tâches. Il est possible de définir la période d'une tâche rapide qui est plus importante que la tâche maître.

Chien de garde

- L'exécution de chaque tâche est contrôlée par un chien de garde configurable, à l'aide des propriétés de la tâche.
- Le tableau suivant offre la place des valeurs du chien de garde pour chacune des tâches.

Tâche	Valeurs du chien de garde (min...max)	Valeur du chien de garde par défaut	Mot système associé
MAST	10..1500 ms	250 ms	%SW11
FAST	10..500 ms	100 ms	-
AUX0	100..5000 ms	2000 ms	-
AUX1	100..5000 ms	2000 ms	-
AUX2	100..5000 ms	2000 ms	-
AUX3	100..5000 ms	2000 ms	-

- Si le dépassement du chien de garde se produit, l'application est déclarée en erreur, ce qui entraîne l'arrêt immédiat de l'automate (état HALT).
- Le mot **%SW11** contient la valeur du chien de garde de la tâche maître en ms. Cette valeur n'est pas modifiable par le programme.
- Le bit **%S11** indique que le chien de garde est dépassé. Il est mis à l'état 1 par le système lorsque la durée de cycle est supérieure au chien de garde.
- Il n'est pas possible de quitter HALT en basculant vers STOP. Pour ce faire, vous devez réinitialiser l'application pour vérifier la cohérence des données.

Unity Pro: le traitement du programme

5. Fonctionnement d'une application multitâche

Contrôle des tâches

Inhibition d'une tâche

- Lorsque le programme d'application s'exécute, il est possible d'activer ou d'inhiber une tâche à l'aide des bits système suivants :

Tâche	Bits système
MAST	%S30
FAST	%S31
AUX0	%S32
AUX1	%S33
AUX2	%S34
AUX3	%S35

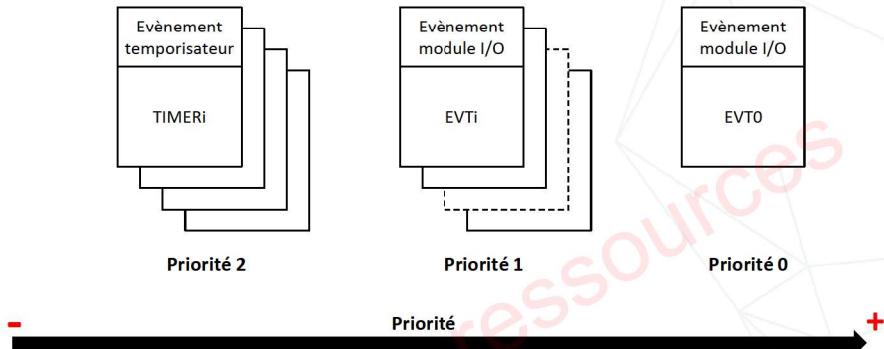
- La tâche est active lorsque le bit système associé est à l'état 1. Ces bits sont testés par le système à la fin de la tâche maître.
- Lorsqu'une tâche est inhibée, les entrées continuent à être lues et les sorties à être écrites.
- Au démarrage du programme d'application, uniquement au premier cycle d'exécution, la tâche maître est active. A la fin du premier cycle, les autres tâches sont automatiquement activées sauf si l'une des tâches est inhibée (associée au bit système réglé sur 0) par le programme.

Unity Pro: le traitement du programme

5. Fonctionnement d'une application multitâche

□ Gestion des Traitements Évènementiels

- ✓ Le traitement évènementiel a la priorité sur les tâches.
- ✓ Il existe 3 niveaux de priorité:



✓ Gestion des Priorités

- Le traitement évènementiel EVT0 est le traitement le plus prioritaire. Il peut interrompre lui-même d'autres types de traitement évènementiel.
- Le traitement évènementiel EVT_i déclenché par des modules d'entrée/sortie (priorité 1) a la priorité sur le traitement évènementiel TIMER_i déclenché par des temporiseurs (priorité 2).
- Sur automates Modicon M580, M340, Premium et Atrium : les types de traitement évènementiel de priorité 1 sont enregistrés et traités dans l'ordre.
- Le traitement évènementiel déclenché par un temporisateur reçoit un niveau de priorité 2. La priorité de traitement est déterminée par le numéro de temporisateur le plus bas.

✓ Contrôle

- Le programme d'application peut globalement valider ou bloquer les divers types de traitement évènementiel en utilisant le bit système %S38. Si un ou plusieurs événements surviennent alors qu'ils sont bloqués, le traitement associé est perdu.
- Deux fonctions élémentaires du langage, **MASKEVT()** et **UNMASKEVT()**, utilisés dans le programme d'application, peuvent être employés pour masquer ou démasquer le traitement évènementiel.
- Si un ou plusieurs évènements surviennent alors qu'ils sont masqués, le système les enregistre et le traitement associé est exécuté après démasquage.

Unity Pro: le traitement du programme

5. Fonctionnement d'une application multitâche

Exécution du traitement événementiel de type TIMER

- ✓ Le traitement événementiel de type TIMER est un processus déclenché par la fonction ITCNTRL (voir Unity Pro, Système, Bibliothèque de blocs).
- ✓ Cette fonction de temporisation active le traitement événementiel, dès que la valeur de présélection est atteinte.

Echanges d'entrées/de sorties dans les traitements événementiels

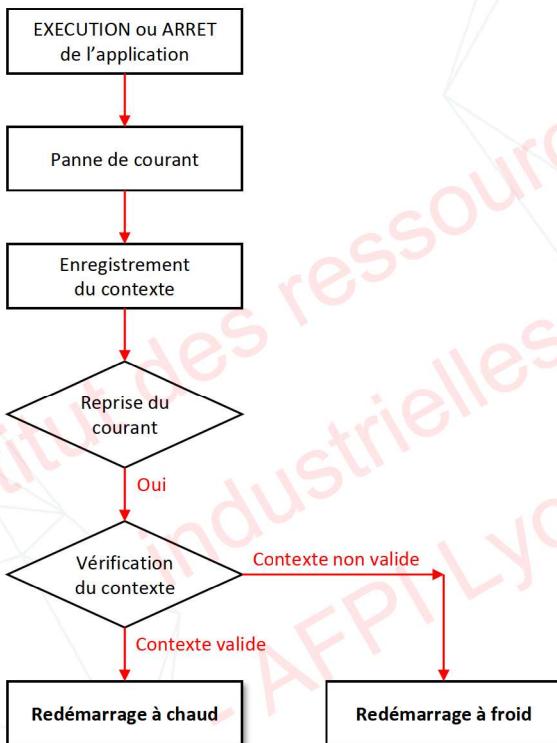
- ✓ Il est possible d'utiliser à chaque traitement événementiel des voies d'entrées/sorties autres que celle relative à l'événement.
- ✓ Comme pour les tâches, les échanges sont alors réalisés implicitement par le système avant (%I) et après (%Q) le traitement applicatif.
 - 1) L'apparition d'un événement dérouté le programme application vers le traitement qui est associé à la voie d'entrée/sortie qui a provoqué l'événement.
 - 2) Toutes les entrées associées au traitement événementiel sont acquises automatiquement.
 - 3) Le traitement événementiel est exécuté. Il doit être le plus court possible.
 - 4) Toutes les sorties associées au traitement événementiel sont mises à jour.

Unity Pro: le traitement du programme

6. Modes de fonctionnement des automates Modicon M340

□ Traitement en cas de COUPURE de COURANT et restauration des automates Modicon M340

- ✓ Si la durée de la coupure est inférieure au temps de filtrage de l'alimentation, il n'y a aucune incidence sur le programme qui s'exécute normalement. Dans le cas inverse, il y a interruption du programme et activation de la restauration de l'alimentation.
- ✓ Les phases du cycle d'alimentation:



- En cas de coupure de courant, le système enregistre le contexte applicatif, les valeurs des variables d'application et l'état du système dans la mémoire Flash interne.
- Le système configure toutes les sorties à l'état de repli (état défini par configuration).
- Dès l'alimentation restaurée, certaines actions et vérifications sont effectuées pour vérifier si le redémarrage à chaud est disponible :
 - ❖ restauration du contexte applicatif à partir de la mémoire Flash interne,
 - ❖ vérification avec la carte mémoire (présence, disponibilité de l'application),
 - ❖ vérification du contexte applicatif pour voir s'il est identique à celui de la carte mémoire.

Unity Pro: le traitement du programme

6. Modes de fonctionnement des automates Modicon M340

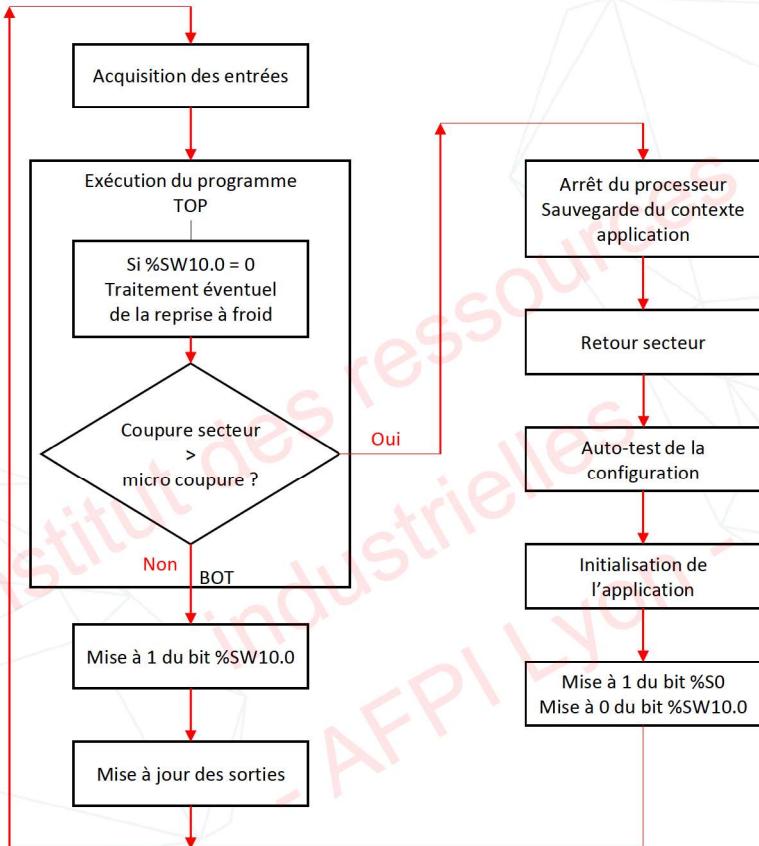
- Traitement en cas de DÉMARRAGE à FROID des automates Modicon M340
 - ✓ Les causes d'un démarrage à froid

Causes	Caractéristiques du démarrage
Chargement d'une application	Démarrage à froid forcé en mode STOP
Rétablissement de l'application située dans la carte mémoire lorsqu'elle est différente de celle dans la RAM interne	Démarrage à froid forcé en mode STOP ou RUN selon la configuration
Rétablissement de l'application située dans la carte mémoire, à l'aide des commandes Automate → Sauvegarde du projet →	Démarrage à froid forcé en mode STOP Le démarrage en mode RUN défini dans la configuration n'est pas pris en compte
Actionnement du bouton RESET de l'alimentation	Démarrage à froid forcé en mode STOP ou RUN selon la configuration
Actionnement du bouton RESET de l'alimentation moins de 500 ms après une mise hors tension	Démarrage à froid forcé en mode STOP ou RUN selon la configuration
Actionnement du bouton RESET de l'alimentation après une erreur de processeur, sauf dans le cas d'une erreur de chien de garde	Démarrage à froid forcé en mode STOP Le démarrage en mode RUN défini dans la configuration n'est pas pris en compte
Initialisation depuis Unity Pro Forçage à 1 du bit système %S0	Démarrage en mode STOP ou RUN (conservation du mode de fonctionnement en cours lors de l'arrêt), initialisation de l'application uniquement
Reprise après une coupure de courant avec perte du contexte	Démarrage à froid forcé en mode STOP ou RUN selon la configuration

Unity Pro: le traitement du programme

6. Modes de fonctionnement des automates Modicon M340

- Traitement en cas de DÉMARRAGE à FROID des automates Modicon M340
 - ✓ Le déroulement d'un redémarrage à froid



Unity Pro: le traitement du programme

6. Modes de fonctionnement des automates Modicon M340

Traitement en cas de DÉMARRAGE à FROID des automates Modicon M340

les phases de reprise de l'exécution du programme lors d'un redémarrage à froid

- Le démarrage est effectué en mode **RUN** ou **STOP**, selon l'état du paramètre « Démarrage automatique en RUN » défini dans la configuration ou, le cas échéant, selon l'état de l'entrée RUN/STOP.
- L'exécution du programme reprend en début de cycle.
- Le système effectue les actions suivantes :
 - ❖ la désactivation des tâches, autres que la tâche maître, jusqu'à la fin du premier cycle de la tâche maître ;
 - ❖ l'initialisation de données (bits, image des E/S, mots...) par les valeurs initiales définies dans l'éditeur de données (valeur réglée sur 0 si aucune valeur initiale n'est définie). Les valeurs des mots %MW peuvent être récupérées en cas de redémarrage à froid si les deux conditions suivantes sont remplies :
 - L'option « Initialiser %MW au démarrage à froid » est décochée dans l'écran de configuration du processeur.
 - La mémoire flash interne a une sauvegarde valide (**%SW96**)
 - ❖ l'initialisation des blocs fonction élémentaires à partir des données initiales,
 - ❖ l'initialisation des données déclarées dans les blocs fonction dérivés (DFB) : 0 ou valeur initiale déclarée dans le type de DFB,
 - ❖ l'initialisation des bits et mots système,
 - ❖ le positionnement des graphes sur les étapes initiales,
 - ❖ l'annulation de tout forçage,
 - ❖ l'initialisation des files de messages et d'événements,
 - ❖ l'envoi des paramètres de configuration à tous les modules d'E/S TOR et spécialisés.
- Pour ce premier cycle de reprise, le système effectue les opérations suivantes :
 - ❖ Il relance la tâche maître avec les bits **%SO** (redémarrage à froid) et **%S13** (premier cycle en mode RUN) mis à 1, et le mot **%SW10** (détectio

Traitement par programme:

- Il est conseillé de tester le bit **%SW10.0** pour détecter un démarrage à froid et lancer un traitement spécifique à ce démarrage à froid.

Modification des sorties:

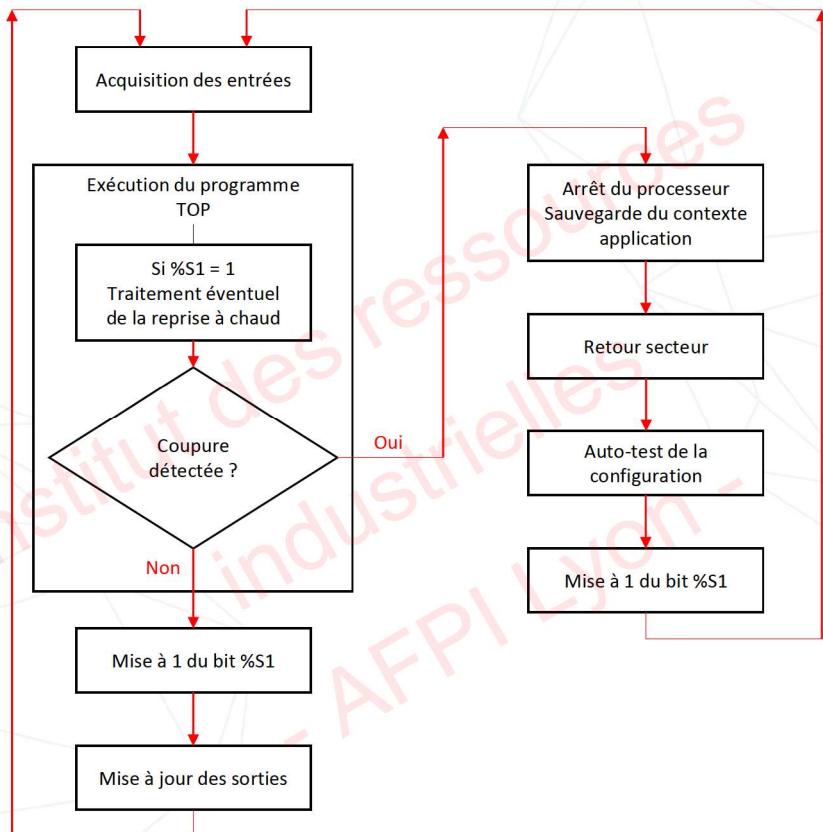
- Dès la détection de la coupure de courant, les sorties prennent la valeur de repli ou conservent leur valeur courante suivant la configuration.
- Au rétablissement de l'alimentation, les sorties restent à 0 jusqu'à ce qu'elles soient mises à jour par la tâche.

Unity Pro: le traitement du programme

6. Modes de fonctionnement des automates Modicon M340

□ Traitement sur REPRISE à CHAUD pour les automates Modicon M340

- ✓ Une reprise à chaud peut être provoquée par un rétablissement du courant sans perte du contexte.
- ✓ le déroulement d'une reprise à chaud:



Unity Pro: le traitement du programme

6. Modes de fonctionnement des automates Modicon M340

- **Traitement sur REPRISE à CHAUD pour les automates Modicon M340**
 - ✓ **Les phases de reprise de l'exécution du programme sur reprise à chaud :**
 - L'exécution du programme ne reprend pas à partir de l'élément où a eu lieu la coupure de courant. Le programme restant est ignoré pendant le démarrage à chaud. Chaque tâche redémarre depuis le début.
 - A la fin du cycle de reprise, le système effectue :
 - ❖ la restitution de la valeur variable de l'application,
 - ❖ la mise du bit %S1 à l'état 1,
 - ❖ l'initialisation des files de messages et d'événements,
 - ❖ l'envoi des paramètres de configuration à tous les modules d'entrées/sorties TOR et métiers,
 - ❖ la désactivation de la tâche rapide et des traitements événementiels (jusqu'à la fin du premier cycle de la tâche maître).
 - Le système effectue un cycle de reprise dans lequel il :
 - ❖ relance la tâche maître depuis le début du cycle,
 - ❖ remet à l'état 0 le bit %S1 à la fin de ce premier cycle de la tâche maître,
 - ❖ réactive la tâche rapide et les traitements événementiels à la fin de ce premier cycle de la tâche maître.
 - ✓ **Traitement par programme de la reprise à chaud :**
 - En cas d'une reprise à chaud, si vous souhaitez un traitement particulier de l'application, vous devez écrire le programme correspondant pour vérifier que %S1 est à l'état 1 en début de programme de la tâche maître.
 - ✓ **Modification des sorties:**
 - Dès la détection de la coupure de courant, les sorties prennent la valeur de repli ou conservent leur valeur courante suivant la configuration.
 - Au rétablissement du courant, les sorties restent en mode sécurité (état 0) jusqu'à leur mise à jour par une tâche d'exécution.