

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

- 05 - Unity Pro: présentation des données

1. PRESENTATION GENERALE DES DONNEES	3
2. LES FAMILLES DE TYPES DE DONNEES	4
<input type="checkbox"/> INTRODUCTION	
<input type="checkbox"/> LES VARIABLES	
<input type="checkbox"/> LES FONCTIONS	
<input type="checkbox"/> LES FONCTIONS ELEMENTAIRES (EF)	
<input type="checkbox"/> LES BLOCS FONCTIONS ELEMENTAIRES (EFB)	
<input type="checkbox"/> LES BLOCS FONCTIONS DERIVES (DFB)	
3. PRÉSENTATION DES INSTANCES DE DONNÉES	10
<input type="checkbox"/> INTRODUCTION	
<input type="checkbox"/> LES ADRESSES NON LOCALISEES	
<input type="checkbox"/> LES ADRESSES LOCALISEES	
<input type="checkbox"/> EXEMPLES	
<input type="checkbox"/> ALLOCATION MÉMOIRE DES INSTANCES	
4. PRÉSENTATION DES RÉFÉRENCES DE DONNÉES	12
<input type="checkbox"/> INTRODUCTION	
<input type="checkbox"/> RÉFÉRENCES DE DONNÉES POSSIBLES SELON LE TYPE DE DONNÉES	

1. PRESENTATION GENERALE DES DONNEES

□ GENERALITES

- ✓ Une donnée désigne un d'objet qui peut être instancié tel que:
 - une variable,
 - un bloc fonction.
- ✓ Les données sont définies en **trois phases** qui sont :
 - la **phase TYPES de DONNÉES**, dans laquelle est précisée :
 - ❖ sa **catégorie**,
 - ❖ son **format**.
 - la **phase INSTANCES de DONNÉES**, dans laquelle est défini son emplacement mémoire et sa propriété qui est :
 - ❖ **localisée** ou
 - ❖ **non localisée**.
 - la **phase RÉFÉRENCES de DONNÉES**, dans laquelle est défini son moyen d'accès :
 - ❖ par **valeur immédiate**,
 - ❖ par **nom**,
 - ❖ par **adresse**.

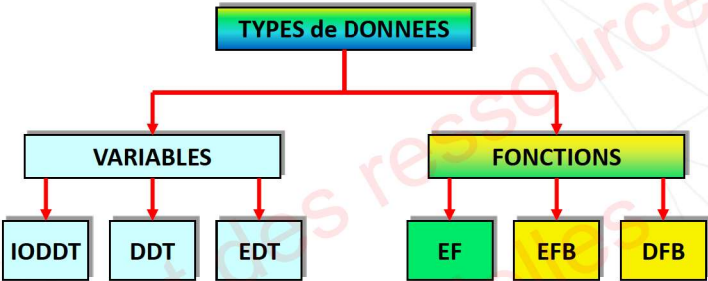


- ✓ **Instancier** une donnée consiste à partir de son type à lui allouer un emplacement mémoire.
- ✓ **Référencier** une donnée consiste à lui définir une référence (nom, adresse, etc...) permettant de l'atteindre en mémoire.

2. LES FAMILLES DE TYPES DE DONNEES

INTRODUCTION

- ✓ Un **type de donnée** est une information logique qui spécifie pour une donnée :
 - sa **structure**,
 - son **format**,
 - la liste de ses **attributs**,
 - son **comportement**.
- ✓ Ces propriétés sont partagées par toutes les instances du type de donnée.
- ✓ Les familles de types de données sont classées dans différentes catégories:



LES VARIABLES

- ✓ Une variable permet d'identifier des objets de données.
- ✓ Son contenu est défini par :
 - **1 NOM** (32 caractères maximum),
 - **1 TYPE**

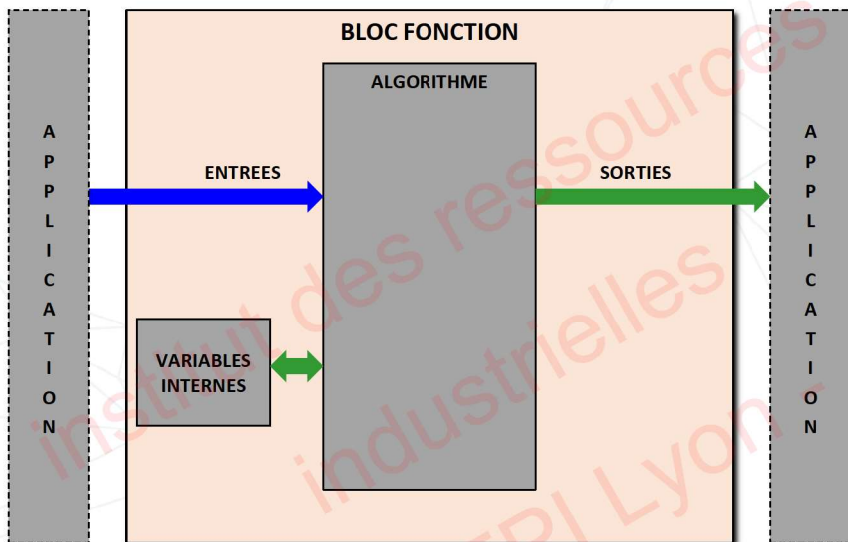
TYPE		DESIGNATION	EXEMPLE
IODDT	IO Derived Data Type	Données d'Entrées/Sorties dérivées Variable associée à plusieurs éléments d'un module d'entrées/sorties	Éléments nécessaires au diagnostic d'une voie analogique
DDT	Derived Data Type	Donnée Dérivée ou personnalisée pour une variable associée à plusieurs éléments	Tableau de données
EDT	Elementary Data Type	Donnée élémentaire	Temporisation, compteur, variable interne

- **1 ADRESSE** ou **1 NOM** (non obligatoire)
- **1 VALEUR INITIALE** (non obligatoire)
- **1 COMMENTAIRE** (non obligatoire)

2. LES FAMILLES DE TYPES DE DONNEES

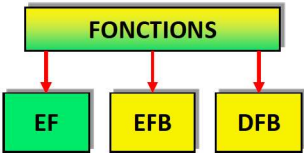
❑ LES FONCTIONS

- ✓ Les blocs fonctions sont des **entités contenant**:
 - des **variables d'entrées** et de **sorties** servant d'interface avec l'application,
 - un **algorithme de traitement** exploitant les variables d'entrées et renseignant les variables de sorties,
 - des **variables internes privées et publiques** exploitées par l'algorithme de traitement.



2. LES FAMILLES DE TYPES DE DONNEES

LES FONCTIONS



TYPE		DESIGNATION	EXEMPLES
EF	Elementary Function 	Fonction Élémentaire Fonction logique prédéfinie qui n'a pas d'information d'état interne L'algorithme est écrit par le constructeur	Comparaison, fonctions logiques
EFB	Elementary Function Blocks 	Blocs Fonctions Élémentaires Fonction logique prédéfinie qui possède des états et des paramètres internes L'algorithme est écrit par le constructeur	Compteurs, temporisations
DFB	Derived Function Blocks 	Blocs Fonctions Dérivés Fonction logique prédéfinie qui possède des états et des paramètres internes L'algorithme est écrit en langage d'automatismes (IL, LD, ST, FBD)	Alarmes, diagnostic


2. LES FAMILLES DE TYPES DE DONNEES

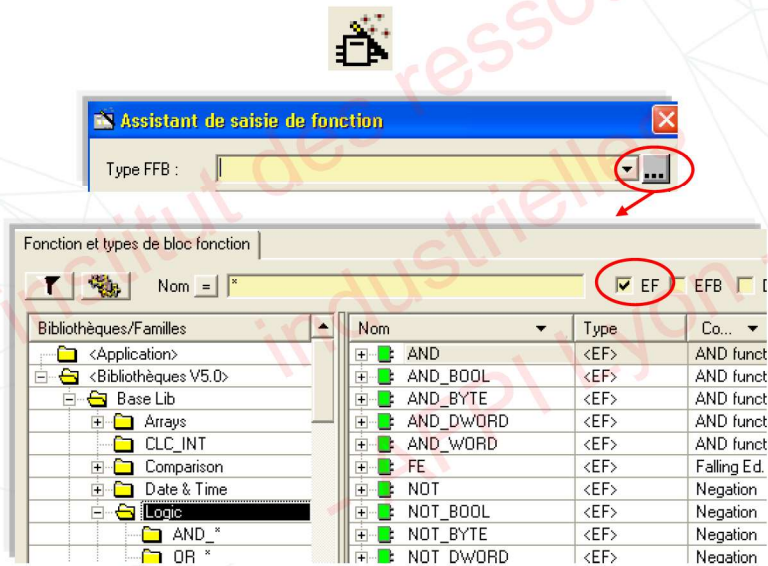
□ LES FONCTIONS ELEMENTAIRES (EF)

✓ DEFINITION

- **Elément**, utilisé dans un programme, qui **réalise** une **fonction logicielle prédéfinie**.
- Une fonction élémentaire EF ne dispose **pas d'information d'état interne**. Plusieurs appels de la même fonction à l'aide des mêmes paramètres d'entrées fournissent toujours les mêmes valeurs de sortie.

✓ UTILISATION

- Icône associé aux Fonctions Élémentaires dans les bibliothèques: 
- Les fonctions élémentaires sont accessibles depuis la bibliothèques et sont classés selon différentes familles:



- Les fonctions élémentaires **ne sont pas instanciables**.


2. LES FAMILLES DE TYPES DE DONNEES

❑ LES BLOCS FONCTIONS ELEMENTAIRES (EFB)

✓ DEFINITION

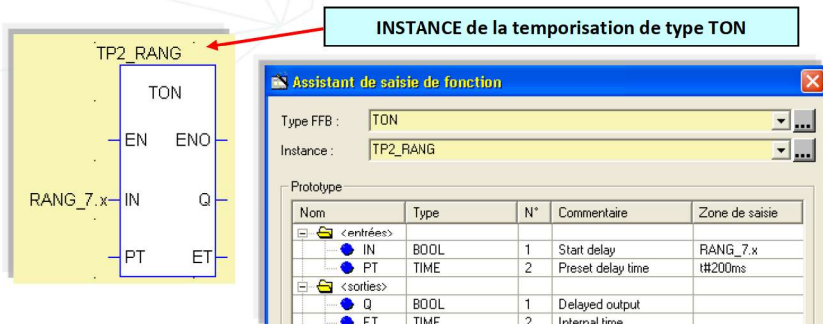
- **Bloc**, utilisé dans un programme, qui **réalise une fonction logicielle prédéfinie**.
- Un bloc fonction élémentaire EFB possède des états et des paramètres internes.

✓ UTILISATION

- Icône associée aux Blocs Fonctions Elémentaires dans les bibliothèques: 
- Les blocs fonctions élémentaires sont accessibles depuis la bibliothèques et sont classés selon différentes familles:



- Les EFB sont **obligatoirement instanciés**. Un espace en mémoire est alloué au bloc dont la taille varie suivant le type de bloc.
- **Exemple:**




2. LES FAMILLES DE TYPES DE DONNEES

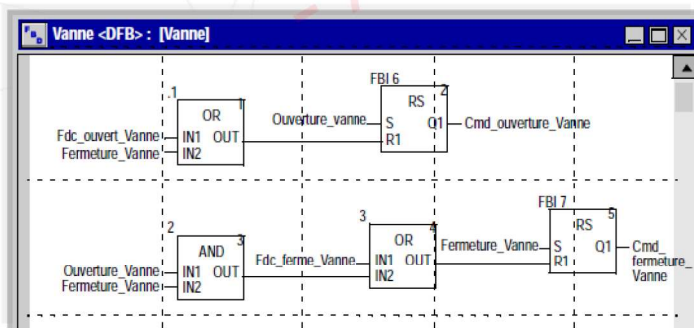
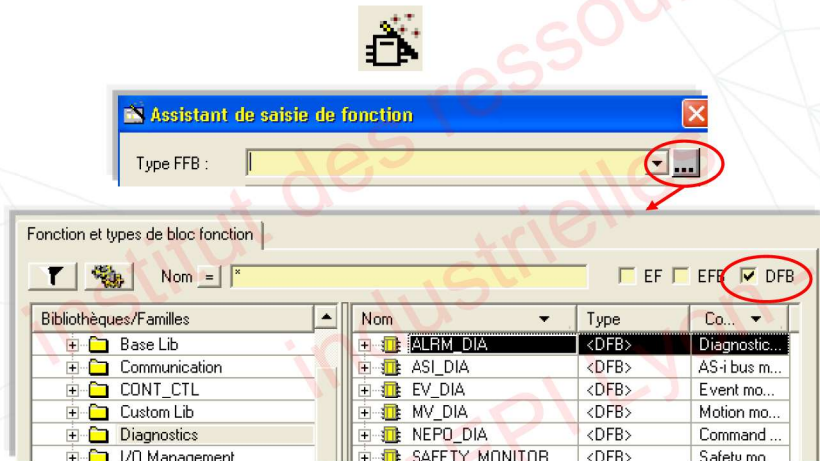
❑ LES BLOCS FONCTIONS DERIVES (DFB)

✓ DEFINITION

- **Bloc**, développé en langage d'automatisme par l'utilisateur, qui **répond** à une **spécificité** de l'application.
- Les blocs fonctions dérivés **DFB** permettent de **structurer et d'optimiser l'application**.

✓ UTILISATION

- Icône associée aux Blocs Fonctions Dérivés dans les bibliothèques: 
- Les blocs fonctions dérivés sont accessibles depuis la bibliothèques et sont classés selon différentes familles:



3. PRÉSENTATION DES INSTANCES DE DONNÉES

❑ INTRODUCTION

- ✓ Une instance de données est une entité fonctionnelle individuelle, qui possède toutes les caractéristiques du type de données duquel elle dépend.
- ✓ Une ou plusieurs instances peuvent être rattachées à un type de données.
- ✓ L'instance de données peut avoir une allocation mémoire :
 - non localisée ou
 - Localisée

❑ LES ADRESSES NON LOCALISEES

- ✓ **L'emplacement mémoire de l'instance est alloué automatiquement par le système**, il peut changer à chaque génération de l'application.
- ✓ La variable est définie par :
 - **1 NOM** choisi par l'utilisateur
 - **1 TYPE**
 - **1 VALEUR INITIALE**
 - **1 COMMENTAIRE**

❑ LES ADRESSES LOCALISEES

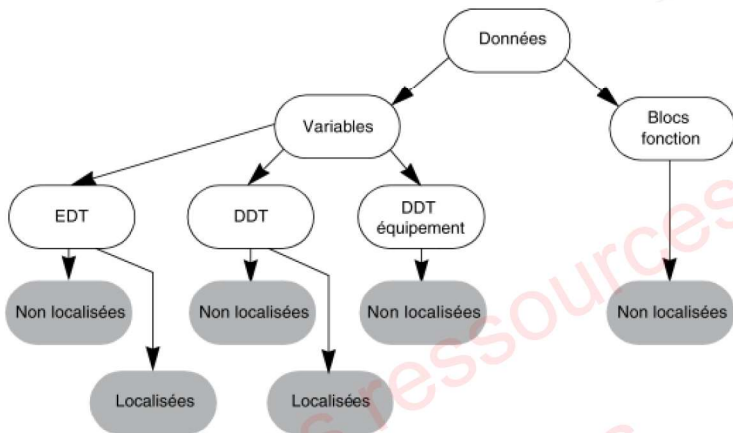
- ✓ **L'emplacement mémoire de l'instance est fixe**, il est prédéfini et ne change jamais.
- ✓ La variable est définie par :
 - **1 NOM** choisi par l'utilisateur
 - **1 ADRESSE** définie par le constructeur
 - **1 TYPE**
 - **1 VALEUR INITIALE**
 - **1 COMMENTAIRE**

❑ EXEMPLES

Nom	Type	Adresse	Valeur	Commentaire
M_MANU	WORD	%MW1		Commandes manuelles
M_PRISE	BOOL			mémoire "prise de la pièce"
M_TEST	BOOL	%MW1.12		Commande de tous les voyants
Mag_vide	BOOL			Magasin vide
Manu	EBOOL	%I0.2.8		Sélection "marche manuelle"

3. PRÉSENTATION DES INSTANCES DE DONNÉES

❑ ALLOCATION MÉMOIRE DES INSTANCES



4. PRÉSENTATION DES RÉFÉRENCES DE DONNÉES

❑ INTRODUCTION

- ✓ Une référence de donnée permet à l'utilisateur d'**accéder** à l'**instance de cette donnée** :
 - par **valeur immédiate**, uniquement pour les données de type EDT,
 - par **adressage**, uniquement pour les données de type EDT,
 - par **nom (symbole)**, pour tous les types de données EDT, DDT, EFB, DFB ainsi que les objets SFC.

❑ RÉFÉRENCES DE DONNÉES POSSIBLES SELON LE TYPE DE DONNÉES

