

# Dossier de conception détaillé

## Projet IEC61499 – **Universal Automation**

*Formation Informatique et Systèmes  
Intelligents Embarqués*  
Année 2025 – 2026

## PRI 5A

### Membres de l'équipe :

Damien LORIGEON – Chef de projet/Dev IEC61131 & IEC61499

### Client :

Jean Paul CHEMLA – Professeur Polytech

Arthur OUSSOUNKIRI ELIEZER GAMBO – Doctorant Université de Reims

Bernard RIERA – Professeur Université de Reims

Stéphane LECASSE – Professeur Université de Reims

Auteur : Damien LORIGEON

Version 2 - 18/12/2025

# Objectifs

Ce document présente la conception détaillé du projet *IEC61499 – Universal Automation*.

Il définit l'architecture fonctionnelle et logicielle globale du système de tri de pièces simulé sous *Factory I/O*, ainsi que les principes de conception retenus pour son implémentation sous *EcoStruxure Control Expert* (IEC 61131-3) et *EcoStruxure Automation Expert* (IEC 61499).

Le dossier de conception général s'inscrit dans la démarche du cycle en V adoptée pour le projet.

Il définit l'architecture globale et les interactions entre les différents modules du système, servant ainsi de référence aux tests d'intégration réalisés ultérieurement.

## Référence

### 1. Internes

Référence :	Titre	Lien
PRI Polytech Tours	Projet IEC61499 – Universal Automation	Dépôt GIT : <a href="https://github.com/ElDLOR/PRI-IEC61499-UniversalAutomation.git">https://github.com/ElDLOR/PRI-IEC61499-UniversalAutomation.git</a> Équipe Teams : <a href="#">Général   Projet - Universal Automation - IEC 61499   Microsoft Teams</a>

### 2. Externes

Référence :	Titre	Lien
UniversalAutomation.org	Ressources IEC 61499 / EAE	<a href="https://universalautomation.org">https://universalautomation.org</a>

## Définition

- **IEC 61131** : norme historique de programmation des automates (POU, Grafcet, ST, etc.).
- **IEC 61499 (EAE)** : norme orientée événements, blocs fonctionnels distribués.
- **Factory IO** : outil de simulation 3D de systèmes industriels.
- **EAE** : EcoStruxure Automation Expert.
- **ECE** : EcoStruxure Control Expert.

# Dossier de conception général

## pour le projet

« Projet IEC61499 – Universal Automation »

Descriptions		
Projet :	Projet IEC61499 – Universal Automation	
Clients	Jean Paul CHEMLA	jean-paul.chemla@univ-tours.fr
Auteurs	Damien LORIGEON	damien.lorigeon@univ-tours.fr
Date d'émission :	26/01/2026	

Validation			
Nom	Date	Valide (O/N)	Commentaires
LORIGEON	26/01/2026	O	
CHEMLA			
RIERA			
GAMBO			

Suivis des versions		
Version	Date	Description de la modifications
1	17/11/2026	Première version
2	18/12/2025	Nouvelle structure IEC61499

## Sommaires

1	Introduction .....	7
1.1	Objectif du document ` .....	7
1.2	Position dans le cycle en V.....	7
1.3	Périmètre du document.....	8
1.4	Diagramme fonctionnels – Grafcet de référence .....	9
2	Liste des variables – Capteurs et Actionneurs .....	12
3	Conception détaillée – IEC61131-3 .....	15
3.1	Organisation générale .....	15
3.2	SFC principal – Grafcet général .....	16
3.3	Action d'aiguillage – E11, E21, E31 .....	17
3.4	Transition ST et LD .....	17
3.5	Sous-GRAFCET de tri .....	18
3.6	Gestion des compteurs .....	19
3.7	Tests unitaires.....	19
4	Conception détaillée – IEC61499 .....	21
4.1	Organisation générale de l'application IEC61499 .....	21
4.2	Configuration matérielle – SoftDpac et communication Modbus TCP.24	
4.2.1	Blocs Modbus utilisés dans le SoftDpac	
	24	
4.2.2	Configuration du bloc MODBUSRCOIL8	
	24	
4.3	Gestion des variables – SymlinkMultiVar.....	25
4.4	Structure fonctionnelle – CAT_EntryConvoyeur.....	27
4.5	Structure fonctionnelle – CAT_VisionDetection.....	30
4.5	Structure fonctionnelle – CAT_TriSequence .....	32

## 1 Introduction

### 1.1 Objectif du document

Le présent document constitue le Dossier de Conception Détaillé (DCD) du projet *IEC61499 – Universal Automation*.

Il décrit de manière précise et internalisée le fonctionnement de chaque module logiciel nécessaire à la réalisation du système de tri de pièces simulé sous *Factory I/O*.

L'objectif du DCD est de fournir une description détaillée de l'architecture interne des éléments développés dans les deux environnements cibles :

- IEC 61131-3 (EcoStruxure Control Expert),
- IEC 61499 (EcoStruxure Automation Expert).

Ce document sert de référence pour la création et la validation des tests unitaires, en garantissant que chaque composant logiciel peut être vérifié indépendamment avant l'intégration globale du système.

### 1.2 Position dans le cycle en V

Dans la démarche du cycle en V, ce document intervient après la conception générale (DCG) et avant les phases d'implémentation et de tests unitaires.

Alors que le DCG décrit l'architecture fonctionnelle globale du système et les interactions entre les modules, le DCD se concentre sur :

- La structure interne des blocs,
- Leurs interfaces (événements, entrées, sorties),
- Leurs variables internes,
- La logique détaillée permettant leur exécution.

Ces informations sont indispensables pour établir des scénarios de tests unitaires complets.

### 1.3 Périmètre du document

Le DCD couvre la conception détaillée des éléments suivants :

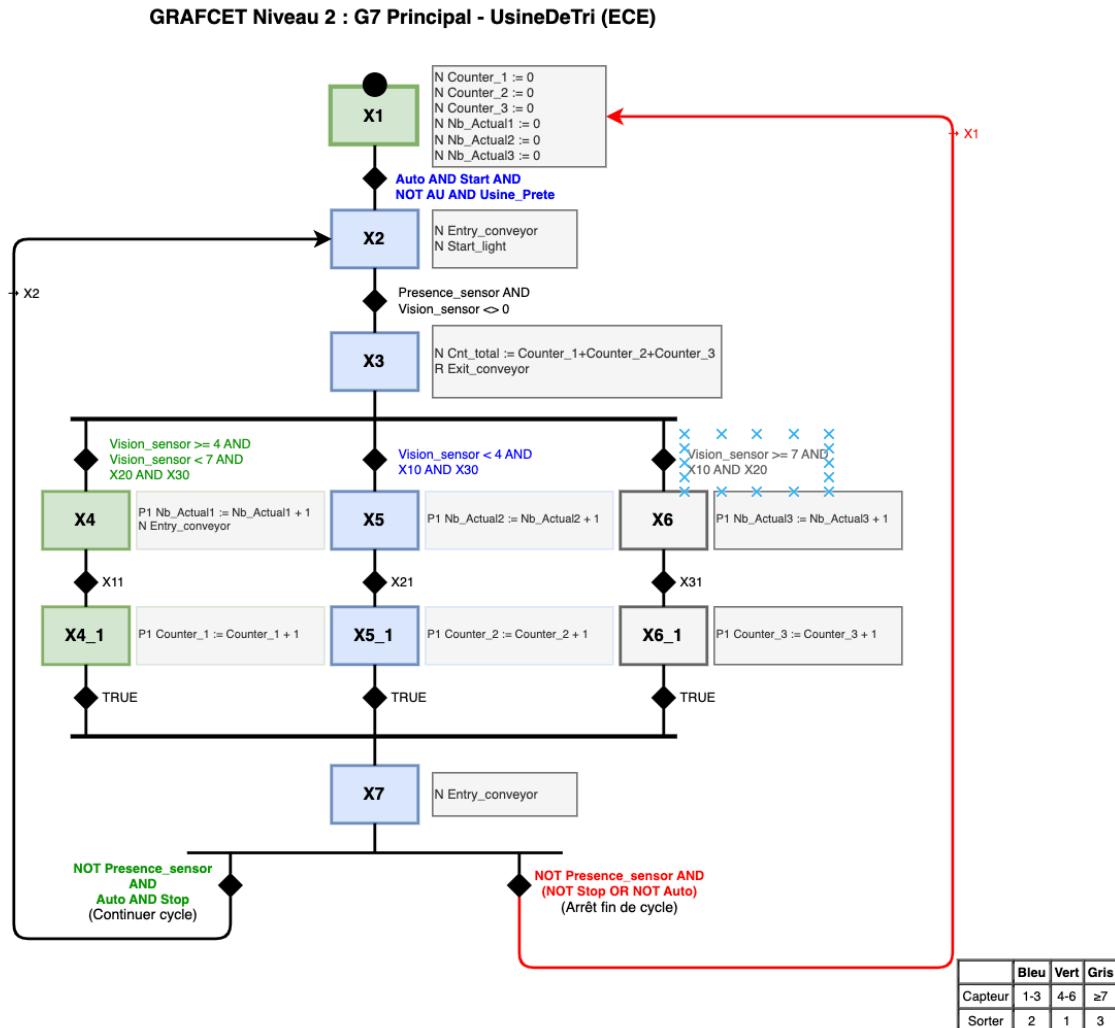
- Les POUs, actions et sous-GRAFCETs développés en IEC 61131-3 ;
- Les FB basiques, composites et blocs de supervision réalisés en IEC 61499 ;
- La structuration des composites fonctionnels (Entrée, Détection, Tri) ;
- L'architecture interne du bloc de communication Modbus TCP ;
- Les interfaces événementielles et données échangées entre les modules ;
- Les algorithmes internes supportant le fonctionnement du GRAFCET.

Sont exclus :

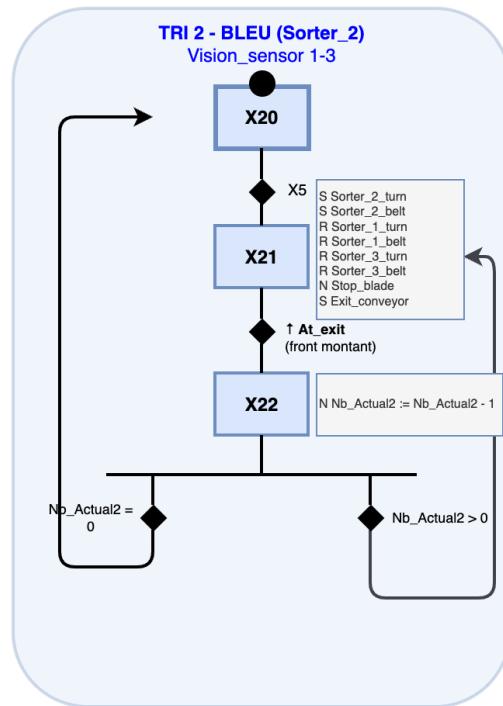
- La description du procédé industriel dans Factory I/O (déjà couverte par le DCG),
- Les aspects de validation globale (test d'intégration),
- Les considérations comparatives entre IEC 61131-3 et IEC 61499 (réservées aux documents de synthèse).

## 1.4 Diagramme fonctionnels – Grafcet de référence

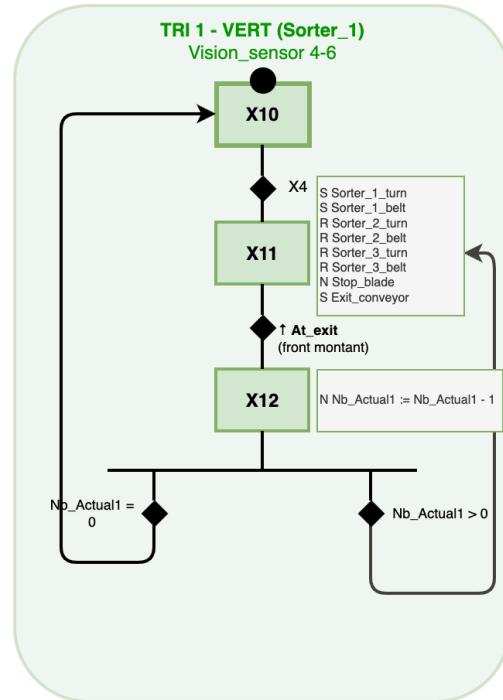
Grafcet général :



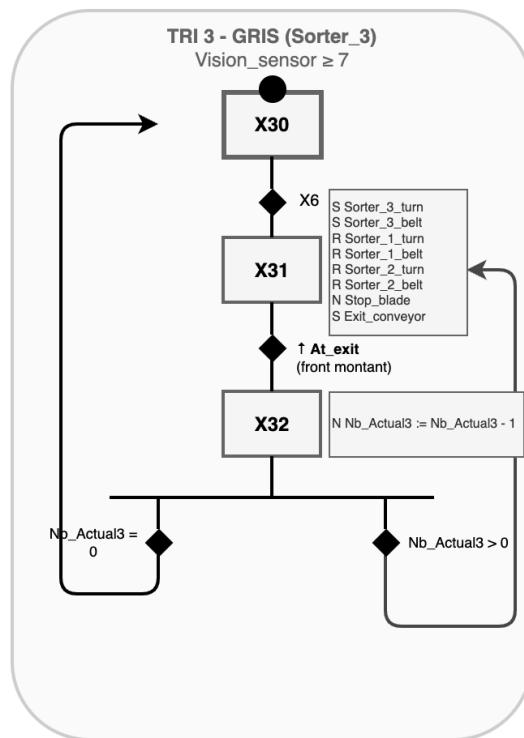
Grafcet de tri - Couleur BLEU :



Grafcet de tri - Couleur VERT :



Grafcet de tri - Couleur GRIS :



## 2 Liste des variables – Capteurs et Actionneurs

Cette section présente l'ensemble des variables utilisées dans l'application, correspondant aux entrées et sorties du procédé simulé sous *Factory I/O*.

Les adresses Modbus associées à chaque variable sont celles configurées dans EcoStruxure Simulation (Soft PLC) pour IEC 61131-3, et dans le SoftDpac pour IEC 61499.

Les variables internes sont également listées afin de documenter les données nécessaires au fonctionnement du GRAFCET et des séquences de tri.

Liste des capteurs :

Nom de variables	Type	Fonctionnement	Adresse Modbus
At_exit	EBOOL	Indique qu'une pièce est en position de sortie.	%M0
AU	EBOOL	Arrêt d'urgence activé.	%M4
Auto	EBOOL	Indique que le système est en mode automatique.	%M5
FACTORY I/O (Running)	EBOOL	Simulation en cours d'exécution.	%M6
Presence_sensor	EBOOL	Détecte la présence d'une pièce sur la ligne.	%M7
Reset	EBOOL	Bouton de remise à zéro appuyé.	%M2
Start	EBOOL	Bouton de démarrage appuyé.	%M1
Stop	EBOOL	Bouton d'arrêt appuyé.	%M3
Vision_sensor	WORD	Capteur de vision identifiant le type de pièce (bleu[1..3], verte[4..7], gris[>7]).	%MW0

Liste des actionneurs :

Nom de variables	Type	Fonctionnement	Adresse Modbus
Counter_1	UINT	Incrémente le compteur de pièces pour la sortie 1.	%MW8
Counter_2	UINT	Incrémente le compteur de pièces pour la sortie 2.	%MW9
Counter_3	UINT	Incrémente le compteur de pièces pour la sortie 3.	%MW10
Entry_conveyor	EBOOL	Commande le tapis d'entrée des pièces.	%M8
Exit_conveyor	EBOOL	Commande le tapis de sortie générale.	%M10
Reset_light	EBOOL	Allume la lampe signalant un reset.	%M18
Sorter_1_belt	EBOOL	Active le convoyeur de tri n°1.	%M12
Sorter_1_turn	EBOOL	Oriente le système de tri vers la sortie 1.	%M11
Sorter_2_belt	EBOOL	Active le convoyeur de tri n°2.	%M14
Sorter_2_turn	EBOOL	Oriente le système de tri vers la sortie 2.	%M13
Sorter_3_belt	EBOOL	Active le convoyeur de tri n°3.	%M16
Sorter_3_turn	EBOOL	Oriente le système de tri vers la sortie 3.	%M15
Start_light	EBOOL	Allume la lampe indiquant un cycle en cours.	%M17
Stop_blade	EBOOL	Stoppe mécaniquement le flux de pièces (barrière).	%M9
Stop_light	EBOOL	Allume la lampe d'arrêt d'urgence.	%M19

Liste des variables internes :

Nom de variables	Type	Fonctionnement	Adresse Modbus
Nb_actual_1	UINT	Variables interne représente le nombre de pièces VERTES en cours de traitement dans le système.	/
Nb_actual_2	UINT	Variables interne représente le nombre de pièces BLEUS en cours de traitement dans le système.	/
Nb_actual_3	UINT	Variables interne représente le nombre de pièces GRISES en cours de traitement dans le système.	/
Cnt_total	UINT	Variable interne permettant d'avoir le nombre total de pièce triées.	/

### 3 Conception détaillée – IEC61131-3

Cette section décrit la structure interne de l'application développée sous EcoStruxure Control Expert, conforme à la norme IEC 61131-3.

L'ensemble de la logique est implémenté principalement en SFC (Sequential Function Chart), complété par des actions d'étape, des transitions ST (Structured Text) et quelques éléments en LD (Ladder Diagram).

L'objectif est de détailler les éléments nécessaires à la réalisation des tests unitaires, en expliquant le comportement interne de chaque composant logiciel.

#### 3.1 Organisation générale

L'application IEC 61131-3 est structurée autour de trois niveaux :

1. Un SFC principal (MAIN)
  - Représentation complète du GRAFCET général du système.
2. Trois sous-GRAFCETs de tri (SFC secondaires)
  - Un pour chaque couleur : vert, bleu, gris.
3. Des actions ST et LD
  - Utilisées pour gérer les transitions, la logique interne, les incrémentations et la synchronisation avec Factory I/O.

Les entrées/sorties Factory I/O ont été intégrées en variables globales, respectant exactement la table d'adressage Modbus TCP du Soft PLC.

### 3.2 SFC principal – Grafcet général

Le SFC principal reproduit le GRAFCET général utilisé pour décrire le fonctionnement du système.

Il comporte les étapes suivantes :

- E1 — Initialisation

Remise à zéro des compteurs, préparation du cycle.

- E2 — Mise en marche du convoyeur d'entrée

Activation du convoyeur d'entrée et allumage du voyant de mise en marche.

- E3 — Détection et addition des compteurs

Détection d'une pièce, mise à jour des compteurs globaux, arrêt du convoyeur d'entrée si nécessaire.

- E4 / E5 / E6 — Sélection du tri

Aiguillage vers le grafcet de tri correspondant (vert / bleu / gris) selon la couleur détectée.

- E7 — Reprise du convoyeur d'entrée

Réactivation du convoyeur pour amener la pièce suivante.

- E10 / E20 / E30 — Traitement des pièces

Étape initiale des grafcets de tris.

- E11 / E21 / E31 — Transport horizontal et validation de sortie

Activation des convoyeurs horizontaux, barrière de tri et validation de la sortie.

- E12 / E22 / E32 — Décrémentation des compteurs

Mise à jour des nombres de pièces restantes à traiter pour chaque couleur.

Chaque étape active des actions en qualification N, S, P ou R, conformément au fonctionnement normal d'un SFC IEC 61131-3.

Qualification	Nom complet
N	Normal
S	SET
R	RESET
P	Pulse

### 3.3 Action d'aiguillage – E11, E21, E31

Les étapes E11, E21 et E31 jouent un rôle essentiel dans la logique d'accès aux sous-GRAFCETs de tri.

Contrairement à des transitions LD, ces étapes contiennent des actions internes utilisant les qualificatifs :

- S (Set) → activation du grafcet de tri correspondant
- R (Reset) → fin du grafcet de tri
- N (Normal) → action normale, continue

Ces actions pilotent des variables logiques internes qui activent :

- Le grafcet de tri vert (E4.x),
- Le grafcet de tri bleu (E5.x),
- Le grafcet de tri gris (E6.x).

Ce mécanisme garantit une séparation claire entre :

- La logique de sélection de la séquence de tri,
- Et la logique opérationnelle du tri elle-même.

### 3.4 Transition ST et LD

#### **Transitions implémentées en LD**

Le LD est utilisé pour gérer uniquement les fronts montants, car il offre une représentation claire et sans ambiguïté.

Sont implémentés en LD :

- At\_Exit (détection du passage en sortie),
- Fronts nécessaires à la synchronisation des transitions du SFC.

## Transitions implémentées en ST

La majorité des transitions sont en ST afin de bénéficier d'une logique plus lisible et modulaire.

Exemples de transitions ST :

- Démarrer : condition de lancement du cycle
- PiecePresente : validation d'une pièce détectée
- PieceVerte / PieceBleue / PieceGrise : aiguillage selon la couleur
- Fini : condition de retour à l'état initial
- PieceSuiv : confirmation d'une pièce suivante à traiter
- E11.x / E21.x / E31.x : conditions logiques complémentaires pour aiguillage

Ces transitions permettent au SFC de suivre exactement le comportement attendu du GRAFCET.

### 3.5 Sous-GRAFCET de tri

Chaque couleur possède son propre grafcet de tri :

- Tri Vert (E4.x → E11 → E12)
- Tri Bleu (E5.x → E21 → E22)
- Tri Gris (E6.x → E31 → E32)

Ces sous-grafcets assurent :

- La commande du convoyeur vertical,
- Le placement de la barrière de séparation,
- L'activation du convoyeur horizontal,
- La validation du capteur de sortie,
- La décrémentation du compteur couleur (Nb\_ActualX).

Ces blocs sont essentiels pour les tests unitaires, car ils peuvent être isolés et vérifiés indépendamment.

### 3.6 Gestion des compteurs

Les compteurs utilisés dans le système sont :

- Counter\_1, Counter\_2, Counter\_3 → Nombre de pièces triées par couleur
- Nb\_Actual1, Nb\_Actual2, Nb\_Actual3 → Nombre de pièces présentes en cours de traitement
- Cnt\_total → Total des pièces détectées

Les incrémentations sont réalisées via des actions en qualification P, permettant un comportement impulsif (une seule incrémentation par front).

La décrémentation ( $Nb\_ActualX := Nb\_ActualX - 1$ ) est effectuée dans les étapes E12 / E22 / E32.

### 3.7 Tests unitaires

Les tests unitaires porteront notamment sur :

Le SFC principal

- Vérification des transitions (Démarrer, Fini, PiecePresente...)
- Activation correcte des étapes
- Cohérence des actions associées

Les sous-GRAFCETs de tri

- Activation/désactivation correcte des convoyeurs
- Détection correcte des fins de course
- Décrémentation correcte des compteurs

### Les transitions ST/LD

- Fronts montants détectés correctement en LD
- Conditions ST évaluées correctement

### Les compteurs

- Initialisation correcte
- Incrémentation impulsionale correcte
- Décrémentation correcte

### Les variables internes

- Mise à jour cohérente
- Aiguillage correct vers les grafcets de tri

## 4 Conception détaillée – IEC61499

Cette section décrit l'architecture interne de l'application développée dans EcoStruxure Automation Expert (EAE), conforme à la norme IEC 61499, basée sur une approche événementielle et orientée composants fonctionnels distribués.

L'objectif est de détailler la structure interne des blocs fonctionnels (FB), des composites (CAT) et du module de communication, afin de produire les éléments nécessaires à la réalisation des tests unitaires.

### 4.1 Organisation générale de l'application IEC61499

L'application IEC 61499 est structurée autour de cinq éléments principaux :

#### 1. Un SoftDpac

Assure la communication Modbus TCP avec Factory I/O en mode Client. → Contient les blocs de lecture (MODBUSRCOIL8, MODBUSRREG1) et d'écriture (MODBUSW8COIL, MODBUSW4REG) des variables. → Factory I/O est configuré en Serveur Modbus TCP.

#### 2. Des blocs SymlinkMultiVar (lecture/écriture)

Assurent le lien entre les variables Modbus et les variables internes de l'application.  
**ReadInCoil** : Lecture des 8 entrées TOR (capteurs) + 1 registre (Vision\_sensor).

**WriteInCoil** : Écriture des 12 sorties TOR (actionneurs) + 3 registres (compteurs).

#### 3. Un bloc E\_CYCLE

Génère un événement cyclique (période 200 ms) déclenchant la lecture/écriture Modbus et l'exécution des CAT. Assure la synchronisation temporelle de l'ensemble de l'application.

#### 4. Cinq modules fonctionnels EQUIP (Composite Function Blocks)

Chacun encapsule un FB basique implémentant la logique métier via un ECC (Execution Control Chart) :

CAT	FB interne	Rôle
EQUIP_EntryConveyor	FB_EntryLogic	Gestion du convoyeur d'entrée, de la Stop blade, des voyants IHM (Start_light, Reset_light, Stop_light), de l'AU et du blocage lors des changements de couleur. Émet PIECE_PRESENTE, AU_OUT, RESET_OUT
EQUIP_VisionSensor	FB_VisionSensor	Lecture du capteur Vision_Sensor (valeurs 1-9), conversion en code couleur (1=Bleu, 2=Vert, 3=Gris), détection des changements via variable Changement. Communication via Socket adapter ICouleur
EQUIP_TRIManager	FB_TriLogic	Gestion de la logique de tri via Plug adapter ICouleur, émission BLOCK/UNBLOCK, compteurs Counter_1/2/3. EQUIP_Sorter (x3) contrôle chaque trieur (Turn+Belt). EQUIP_ExitConveyor détecte front descendant At_exit via E_F_TRIGGER et émet PIECE_SORTIE

#### 5. Des FB basiques (Basic Function Blocks)

Implémentent la logique interne de chaque CAT via un ECC (diagramme d'états-transitions). Chaque état de l'ECC est associé à un algorithme (code ST) et à un événement de sortie. Les transitions sont conditionnées par des expressions logiques sur les entrées.

##### Flux événementiel entre les CAT

Événement	Source	Destination	Description
PIECE_PRESENTE	EQUIP_EntryConveyor	EQUIP_VisionSensor	Pièce détectée par Presence_sensor
PIECE_BLEUE	EQUIP_VisionSensor	EQUIP_TRIManager	Pièce bleue identifiée (Vision 1-3)
PIECE_VERTE	EQUIP_VisionSensor	EQUIP_TRIManager	Pièce verte identifiée (Vision 4-7)

PIECE_GRISE	EQUIP_VisionSensor	EQUIP_TRIManager	Pièce grise identifiée (Vision >7)
PIECE_SORTIE	EQUIP_TRIManager	EQUIP_EntryConveyor	Front montant At_exit détecté

### Données partagées entre les CAT

Variable	Source	Destination	Type	Description
Changement_Couleur	EQUIP_VisionSensor	EQUIP_EntryConveyor	BOOL	Indique un changement de couleur
Tri_En_Cours	EQUIP_TRIManager	EQUIP_EntryConveyor	BOOL	Indique qu'un tri est en cours

Cette structure modulaire permet une séparation claire des responsabilités et facilite la création de tests unitaires indépendants pour chaque CAT.

## 4.2 Configuration matérielle – SoftDpac et communication Modbus TCP

La communication avec Factory I/O repose sur un SoftDpac, configuré comme Client Modbus TCP.

### Configuration Modbus TCP

Paramètre	Valeur
Mode	Client
Adresse IP serveur	?
Port	502
Serveur Modbus	Factory I/O

**Note importante :** Contrairement à la configuration IEC 61131-3 où l'automate simulé (Soft PLC) est serveur, dans EAE le SoftDpac fonctionne en mode **Client**. Factory I/O doit donc être configuré en **Serveur Modbus TCP**.

### 4.2.1 Blocs Modbus utilisés dans le SoftDpac

Bloc	Type	Fonction	Plage d'adresses
MODBUSRCOIL8	Lecture Coils	Lecture des 8 entrées TOR	0-7 (Coils 0x)
MODBURREG1	Lecture Registre	Lecture du Vision_sensor	0 (Holding Register)
MODBUSW8COIL	Écriture Coils	Écriture des 12 sorties TOR	8-19 (Coils 0x)
MODBUSW4REG	Écriture Registres	Écriture des 3 compteurs	8-10 (Holding Registers)

### 4.2.2 Configuration du bloc MODBUSRCOIL8

Paramètre	Valeur	Description
Type de données	inonly	Lecture Coils (plage 0x : 00001-09999)

Paramètre	Valeur	Description
Adresse de départ	0	Premier Coil à lire
Nombre de Coils	8	At_exit, Start, Reset, Stop, AU, Auto, FactoryIO_Run, Presence_sensor

**Point critique :** Le type de données doit être "inonly" et non "input". Le type "input" correspond aux Discrete Inputs (plage 1x), ce qui provoque une erreur de lecture.

#### 4.3 Gestion des variables – SymlinkMultiVar

L'application utilise deux blocs **SymlinkMultiVar** permettant de faire le lien entre les variables Modbus du SoftDpac et les variables internes des CAT.

##### Bloc ReadInCoil (Lecture des entrées)

Variable	Nom Factory I/O	Adresse Modbus	Type
NAME1 / VALUE1	At_exit	%M0 (Coil 0)	BOOL
NAME2 / VALUE2	Start	%M1 (Coil 1)	BOOL
NAME3 / VALUE3	Reset	%M2 (Coil 2)	BOOL
NAME4 / VALUE4	Stop	%M3 (Coil 3)	BOOL
NAME5 / VALUE5	AU	%M4 (Coil 4)	BOOL
NAME6 / VALUE6	Auto	%M5 (Coil 5)	BOOL
NAME7 / VALUE7	FactoryIO_Run	%M6 (Coil 6)	BOOL
NAME8 / VALUE8	Presence_sensor	%M7 (Coil 7)	BOOL
NAME9 / VALUE9	Vision_sensor	%MW0 (Reg 0)	INT

##### Bloc WriteInCoil (Écriture des sorties)

Variable	Nom Factory I/O	Adresse Modbus	Type
NAME1 / VALUE1	Entry_conveyor	%M8 (Coil 8)	BOOL
NAME2 / VALUE2	Stop_blade	%M9 (Coil 9)	BOOL

Variable	Nom Factory I/O	Adresse Modbus	Type
NAME3 / VALUE3	Exit_conveyor	%M10 (Coil 10)	BOOL
NAME4 / VALUE4	Sorter_1_turn	%M11 (Coil 11)	BOOL
NAME5 / VALUE5	Sorter_1_belt	%M12 (Coil 12)	BOOL
NAME6 / VALUE6	Sorter_2_turn	%M13 (Coil 13)	BOOL
NAME7 / VALUE7	Sorter_2_belt	%M14 (Coil 14)	BOOL
NAME8 / VALUE8	Sorter_3_turn	%M15 (Coil 15)	BOOL
NAME9 / VALUE9	Sorter_3_belt	%M16 (Coil 16)	BOOL
NAME10 / VALUE10	Start_light	%M17 (Coil 17)	BOOL
NAME11 / VALUE11	Reset_light	%M18 (Coil 18)	BOOL
NAME12 / VALUE12	Stop_light	%M19 (Coil 19)	BOOL
NAME13 / VALUE13	Counter_1	%MW8 (Reg 8)	INT
NAME14 / VALUE14	Counter_2	%MW9 (Reg 9)	INT
NAME15 / VALUE15	Counter_3	%MW10 (Reg 10)	INT

### Mécanisme de synchronisation

Le bloc **E\_CYCLE** (période 200 ms) déclenche la chaîne d'événements :

E\_CYCLE.EO → ReadInCoil.REQ → CAT.REQ → WriteInCoil.REQ

## 4.4 Structure fonctionnelle – EQUIP\_EntryConveyor

### Rôle du CAT

Le EQUIP\_EntryConveyor gère le convoyeur d'entrée, la barrière (Stop\_blade), les voyants et le blocage lors des changements de couleur.

### Interface – EventInputs

Nom	With	Description
INIT	QI	Initialisation du CAT
REQ	QI, Start, Stop, Reset, Auto, AU, Presence_sensor, Changement_Couleur, Tri_En_Cours	Requête cyclique
PIECE_SORTIE	-	Front montant At_exit depuis EQUIP_TRIManager

### Interface – EventOutputs

Nom	With	Description
INITO	QO	Initialisation terminée
CNF	QO, Entry_conveyor, Stop_blade, Start_light, Reset_light, Stop_light	Confirmation avec sorties
PIECE_PRESENTE	-	Pièce détectée, vers EQUIP_VisionSensor

### Interface – InputVars

Nom	Type	Description
QI	BOOL	Quality Input
Start	BOOL	Bouton démarrage
Stop	BOOL	Bouton arrêt (NF : TRUE = relâché)
Reset	BOOL	Bouton remise à zéro
Auto	BOOL	Mode automatique
AU	BOOL	Arrêt d'urgence
Presence_sensor	BOOL	Capteur présence pièce
Changement_Couleur	BOOL	Depuis EQUIP_VisionSensor

Nom	Type	Description
Tri_En_Cours	BOOL	Depuis EQUIP_TRIManager

### Interface – OutputVars

Nom	Type	Description
QO	BOOL	Quality Output
Entry_conveyor	BOOL	Commande convoyeur entrée
Stop_blade	BOOL	Commande barrière
Start_light	BOOL	Voyant démarrage
Reset_light	BOOL	Voyant reset
Stop_light	BOOL	Voyant arrêt

### ECC – États et transitions

État	Action	Description	Event Output
START	-	État initial	-
INIT	ACT_INIT	Initialisation des sorties	INITO
IS_RUN	ACT_RUN	Convoyeur en marche	CNF
PIECE_DETECTÉ	ACT_DETECT	Pièce détectée	CNF,PIECE_PRESENT
CHECK_BLOCK	ACT_CHECK	Vérification blocage	CNF
BLOCKED	ACT_BLOCKED	Bloqué, attente pièce sortie	CNF
UNBLOCK	ACT_UNBLOCK	Déblocage	CNF

#### Algorithme ACT\_INIT

QO := TRUE; Entry\_conveyor := FALSE; Stop\_blade := FALSE; Start\_light := FALSE; Reset\_light := TRUE;  
 Stop\_light := FALSE; Bloque := FALSE;

#### Algorithme ACT\_RUN

Entry\_conveyor := TRUE; Stop\_blade := FALSE; Start\_light := TRUE; Reset\_light := FALSE; Stop\_light := FALSE;

#### Algorithme ACT\_DETECT

Entry\_conveyor := TRUE; Stop\_blade := FALSE; Start\_light := TRUE; Bloque := Changement\_Couleur AND  
 Tri\_En\_Cours;

### Algorithme ACT\_BLOCKED

Entry\_conveyor := FALSE; Stop blade := TRUE; Start\_light := TRUE;

#### Tests unitaires associés

ID	Test	Résultat attendu
TU-E01	Initialisation	QO = TRUE, Reset_light = TRUE
TU-E02	Démarrage (Auto AND Start AND Stop AND NOT AU)	Entry_conveyor = TRUE
TU-E03	Détection pièce	PIECE_PRESENTE émis
TU-E04	Blocage changement couleur	Stop blade = TRUE
TU-E05	Déblocage (PIECE_SORTIE)	Entry_conveyor = TRUE

## 4.5 Structure fonctionnelle – EQUIP\_VisionSensor

### Rôle du CAT

Le EQUIP\_VisionSensor gère la lecture du capteur Vision\_sensor, l'identification de la couleur (Bleu/Vert/Gris) et la détection des changements de couleur.

### Interface – EventInputs

Nom	With	Description
INIT	QI	Initialisation
REQ	QI, Vision_sensor, Presence_sensor	Requête cyclique
PIECE_PRESENTE	-	Depuis EQUIP_EntryConveyor

### Interface – EventOutputs

Nom	With	Description
INITO	QO	Initialisation terminée
CNF	QO, Couleur_Actuelle, Changement_Couleur	Confirmation avec données
PIECE_BLEUE	-	Pièce bleue détectée
PIECE_VERTE	-	Pièce verte détectée
PIECE_GRISE	-	Pièce grise détectée

### Interface – Variables

Nom	Type	I/O	Description
QI	BOOL	Input	Quality Input
Vision_sensor	INT	Input	Valeur du capteur couleur
Presence_sensor	BOOL	Input	Capteur présence
QO	BOOL	Output	Quality Output
Couleur_Actuelle	INT	Output	1=Bleu, 2=Vert, 3=Gris
Changement_Couleur	BOOL	Output	TRUE si couleur différente
Couleur_Precedente	INT	Internal	Mémorisation couleur précédente

### Correspondance Vision\_sensor / Couleur

Valeur Vision_sensor	Couleur	Événement émis
1, 2, 3	Bleu	PIECE_BLEUE
4, 5, 6, 7	Vert	PIECE_VERT
> 7	Gris (métal)	PIECE_GRISE

### ECC – États

État	Action	Event Output
START	-	-
INIT	ACT_INIT	INITO
DETECT	ACT_DETECT	CNF
BLEU	ACT_BLEU	PIECE_BLEUE
VERT	ACT_VERT	PIECE_VERT
GRIS	ACT_GRIS	PIECE_GRISE

### Algorithme ACT\_BLEU / ACT\_VERT / ACT\_GRIS

```

Couleur_Precedente := Couleur_Actuelle; Couleur_Actuelle := X; (* 1, 2 ou 3 *) IF Couleur_Precedente <> 0
AND Couleur_Precedente <> Couleur_Actuelle THEN           Changement_Couleur := TRUE; ELSE
Changement_Couleur := FALSE; END_IF;
  
```

### Tests unitaires associés

ID	Test	Résultat attendu
TU-V01	Détection bleu (Vision=2)	PIECE_BLEUE émis, Couleur_Actuelle=1
TU-V02	Détection vert (Vision=5)	PIECE_VERT émis, Couleur_Actuelle=2
TU-V03	Détection gris (Vision=9)	PIECE_GRISE émis, Couleur_Actuelle=3
TU-V04	Changement Bleu→Vert	Changement_Couleur = TRUE
TU-V05	Même couleur	Changement_Couleur = FALSE

## 4.5 Structure fonctionnelle – EQUIP\_TRIManager

### Rôle du CAT

Le EQUIP\_TRIManager gère les 3 sorters, le convoyeur de sortie, les compteurs et la détection du front montant At\_exit via E\_R\_TRIG.

### Interface – EventInputs

Nom	With	Description
INIT	QI	Initialisation
REQ	QI, At_exit	Requête cyclique
PIECE_BLEUE	-	Depuis EQUIP_VisionSensor
PIECE_VERTE	-	Depuis EQUIP_VisionSensor
PIECE_GRISE	-	Depuis EQUIP_VisionSensor

### Interface – EventOutputs

Nom	With	Description
INITO	QO	Initialisation terminée
CNF	QO, Exit_conveyor, Sorter_X_turn/belt, Counter_X, Tri_En_Cours	Confirmation
PIECE_SORTIE	-	Front montant At_exit détecté

### Interface – OutputVars

Nom	Type	Description
Exit_conveyor	BOOL	Convoyeur de sortie
Sorter_1_turn / Sorter_1_belt	BOOL	Sorter 1 (Bleu)
Sorter_2_turn / Sorter_2_belt	BOOL	Sorter 2 (Vert)
Sorter_3_turn / Sorter_3_belt	BOOL	Sorter 3 (Gris)
Counter_1 / Counter_2 / Counter_3	INT	Compteurs pièces triées
Tri_En_Cours	BOOL	Indique qu'un tri est actif

### Bloc E\_R\_TRIG (Rising Trigger)

Le bloc E\_R\_TRIG est utilisé pour détecter un front montant sur At\_exit :

REQ → [E\_R\_TRIG] → AT\_EXIT\_RISING At\_exit → (QI)

### ECC – États

État	Action	Event Output
START	-	-
INIT	ACT_INIT	INITO
TRI_BLEU	ACT_TRI_BLEU	CNF
TRI_VERT	ACT_TRI_VERT	CNF
TRI_GRIS	ACT_TRI_GRIS	CNF
CNT_BLEU	ACT_CNT_BLEU	CNF, PIECE_SORTIE
CNT_VERT	ACT_CNT_VERT	CNF, PIECE_SORTIE
CNT_GRIS	ACT_CNT_GRIS	CNF, PIECE_SORTIE

### Algorithme ACT\_TRI\_BLEU

```
Exit_conveyor := TRUE; Sorter_1_turn := TRUE; Sorter_1_belt := TRUE; Sorter_2_turn := FALSE; Sorter_2_belt
:= FALSE; Sorter_3_turn := FALSE; Sorter_3_belt := FALSE; Tri_En_Cours := TRUE;
```

### Algorithme ACT\_CNT\_BLEU

```
Counter_1 := Counter_1 + 1; Tri_En_Cours := FALSE;
```

### Tests unitaires associés

ID	Test	Résultat attendu
TU-T01	Tri bleu (PIECE_BLEUE)	Sorter_1 activé, Exit_conveyor = TRUE
TU-T02	Comptage (AT_EXIT_RISING)	Counter_X++, PIECE_SORTIE émis
TU-T03	Tri_En_Cours actif	TRUE en état TRI_X
TU-T04	Changement sortir	Transition TRI_X → TRI_Y