




 <p>SOLUTIONS D'AUTOMATISME INDUSTRIEL</p>	<p><b>SOUS SYSTEME TRIAGE DE PIECES</b></p>	<p>TRIAGE DE PIECES PAR IA MODELE PEDAGOGIQUE VIRTUEL</p>
---	---	---

## Structure matérielle

Détection			Commande du process	Process virtuel
				

## Cahier des charges

### Contexte

La partie opérative est simulée avec le logiciel FACTORY I/O (FIO).

Cette simulation permet de générer et de trier automatiquement des pièces sur un convoyeur principal à partir de la détection de couleurs via une caméra.

La chaîne de détection repose sur l'utilisation d'une webcam chargée de capturer l'image de l'objet à analyser. La reconnaissance de la couleur de la pièce est réalisée grâce à une approche Deep Learning mise en œuvre à l'aide de la plateforme Teachable Machine. Celle-ci permet de classer les objets détectés selon deux catégories : pièces bleues et pièces roses.

La prise de décision, basée sur la classe détectée ainsi que sur le niveau de confiance associé, est assurée par un programme développé en langage Python. Ce programme gère également la communication avec la partie commande via une interface MODBUS.

Enfin, la partie commande du process est confiée à un automate industriel virtuel M221 de Schneider Electric, chargé de piloter le convoyeur et les dispositifs de tri dans l'environnement FACTORY I/O.


### Entrées / Sorties

#### Entrées

- Webcam : détection de la couleur de l'objet
  - Couleur détectée = rose
  - Couleur détectée = bleu
- FIO : Capteurs de couleur des pièces sur le convoyeur principal (vert / bleu)

#### Sorties

- FIO : Génération d'une pièce verte
- FIO : Génération d'une pièce bleue
- FIO : Actionneur de déviation vers :
  - Convoyeur pièces vertes
  - Convoyeur pièces bleues

 <p>SOLUTIONS D'AUTOMATISME INDUSTRIEL</p>	<p><b>SOUS SYSTEME TRIAGE DE PIECES</b></p>	<p>TRIAGE DE PIECES PAR IA MODELE PEDAGOGIQUE VIRTUEL</p>
---	---	---

## Règles de fonctionnement

- Si la webcam détecte un objet rose → génération d'une pièce verte
- Si la webcam détecte un objet bleu → génération d'une pièce bleue
- Chaque pièce est transportée sur le convoyeur principal
- Dans la zone de tri :
  - Les pièces vertes sont dirigées vers le convoyeur vert
  - Les pièces bleues sont dirigées vers le convoyeur bleu

Lire la vidéo en suivant ce lien : <https://www.youtube.com/watch?v=J24TMIg7nQc>

## Exploitation d'un modèle Teachable Machine avec Python

---

### Étape 1 : Entraînement du modèle

- ☐ Créer un projet sur Teachable Machine
- ☐ Définir les classes « pièce bleue » et « pièce rose »
- ☐ Capturer un nombre suffisant d'images pour chaque classe
- ☐ Lancer l'entraînement du modèle
- ☐ Vérifier la qualité de la reconnaissance (tests en temps réel)

### Étape 2 : Exportation du modèle

- ☐ Cliquer sur Exporter le modèle
- ☐ Sélectionner le format TensorFlow / Keras
- ☐ Télécharger l'archive générée par Teachable Machine
- ☐ Extraire les fichiers :
  - keras\_model.h5
  - labels.txt

### Étape 3 : Préparation de l'environnement Python


- ☐ Ouvrir l'IDE Python (PyCharm, par exemple)
- ☐ Vérifier que l'environnement virtuel (.venv) est activé
- ☐ Identifier le répertoire de travail du projet

### Étape 4 : Installation du modèle

- ☐ Copier les fichiers keras\_model.h5 et labels.txt
- ☐ Les coller dans le répertoire de travail du projet Python (ex. dossier .venv)

### Étape 5 : Intégration du code Python

- ☐ Créer le script Python
- ☐ Copier le code fourni pour :
  - l'acquisition vidéo (webcam)
  - le chargement du modèle Keras

 <p>SOLUTIONS D'AUTOMATISME INDUSTRIEL</p>	<p><b>SOUS SYSTEME TRIAGE DE PIECES</b></p>	<p>TRIAGE DE PIECES PAR IA MODELE PEDAGOGIQUE VIRTUEL</p>
---	---	---

- la classification des images
- l'exploitation du taux de confiance

### Étape 6 : Test de la reconnaissance

- ☐ Lancer le programme Python
- ☐ Tester avec des objets bleus et roses
- ☐ Vérifier la classe détectée et le niveau de confiance

### Étape 7 : Mise en œuvre de la communication MODBUS

- ☐ Identifier le rôle des équipements :
  - Python : client 1 MODBUS
  - FIO : client 2 MODBUS
  - Automate M221 : serveur MODBUS
- ☐ Configurer les **adresses IP** dans le même masque de sous réseau et le **port MODBUS (502)**.
  - `host = '127.0.0.1' # Adresse IP du serveur Modbus`
  - `port = 502 # Port Modbus (502 par défaut)`
- ☐ Importer et installer la bibliothèque MODBUS Python (ex. pymodbus)
- ☐ Programmer dans Python :
  - l'écriture des données de classification vers l'automate
  - la gestion des erreurs de communication

### Mise en œuvre de l'automate programmable virtuel M221


---

- ☐ Utiliser le logiciel EcoStruxure Machine Expert - Basic
- ☐ Configurer l'automate M221 :
  - association des variables MODBUS aux variables internes
- ☐ Saisir le programme pour la gestion du process

### Mise en œuvre de Factory I/O

---

- ☐ Créer ou importer la scène
- ☐ Configurer FIO :
  - activation du client MODBUS
  - association des variables internes aux variables MODBUS

	<b>SOUS SYSTEME TRIAGE DE PIECES</b>	<b>TRIAGE DE PIECES PAR IA MODELE PEDAGOGIQUE VIRTUEL</b>
---	--	---

## Validation

---

- ☐ EcoStruxure Machine Expert – Basic :
  - Lancer le simulateur
  - Exécuter le programme
  
- ☐ Factory I/O
  - Ouvrir la scène
  - Connecter le pilote MODBUS
  - Lancer la simulation
  
- ☐ Python
  - Exécuter le programme
  
- ☐ Résultat attendu
  - Lire la vidéo en suivant ce lien : <https://www.youtube.com/watch?v=J24TMlg7nQc>