C6 - Lecture du N° de boite par CAMÉRA M120 et Dialogue AVEC L’API

Caméra Matrix 120 de chez Datalogic Code-barres EAN8 (Seuls les 4 chiffres de droite sont significatifs pour notre activité)



STATIONs utilisÉEs : Tool1 ou 2 ou 3 avec une balance et CAMÉRA M120, Tool4 et un HMI.

# Consignes gÉnÉrales

Chaque étudiant doit faire son application sur **son** PC.

Vérifiez la mise en route (bouton 0/I du panneau de commande) du Tool ou des Tools que vous utilisez. Repérez bien les adresses IP indiquées sur les matériels.

## Objectif

Le but de ce TP est de :

* D’installer la CPU 1512C avec sa carte de communication PtP, de configurer la M120 et d’acquérir les caractères du code-barres lu par la Matrix 120 ;
* De créer une FC pour extraire le numéro de boite et l’afficher sur l’HMI ;
* Puis de configurer d’un deuxième API (Tool4) pour qu’il reçoive le numéro et le poids de la boite lus au Tool1.

# Niveau 1

## Objectifs de ce niveau

D’installer la CPU 1512C avec sa carte de communication PtP, de configurer la M120 et d’acquérir les caractères intéressants du code-barres lu par la Matrix 120.

*Nota : Pour la configuration vous pouvez utiliser la notice sur le poste.*

## Installation de la CPU + CM PtP et adressage E/S

Dans un **nouveau projet**:

**Installez** une CPU 1512C et nommez la ToolX (X est le n° du Tool)

**Dotez-la** **d’un sous-réseau** et de son adresse IP (10.96.9.3X) **indiquée sur l’API**.

**Installez** la carte de communication PtP (attention de bien vérifier la référence sur le poste : HF ou BA) et **configurez** la carte avec les paramètres de connexion RS232 indiqués sur la balance.

I**ndiquez** les adresses pour les cartes trois cartes E/S (1000 ; 0 ; 2 respectivement pour les entrées et pour les sorties).

## VÉrification du fonctionnement de l’API

**Ouvrez** l’OB1 et programmez un réseau pour allumer la lampe verte (%Q2.5) lorsque vous appuierez sur le bouton Start (%I0.0).

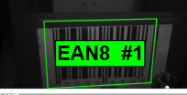
Par un clic droit, renommez %I0.0 en Tn-BP\_Start et %Q2.5 en Tn-L Verte (n = n° de Tool).

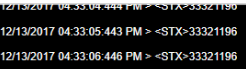
## VÉrification des rÉglages de la M120 par DL Code

**Lancez** DLCode (icône sur le bureau) et **faites la recherche** des M120 dans le réseau .

La caméra est configurée pour acquérir le code toutes les secondes (cela peut être vérifié en observant le flash de la caméra).

**Passez** en affichage Moniteur pour obtenir l’image vue par la M120. Le code EAN8 doit être reconnu. Cela vous permet de vérifier la position et l’orientation de la caméra M120 :



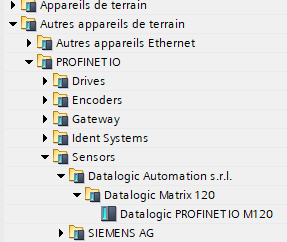
**Affichez** la console pour faire apparaitre le code lu toutes les secondes. 

**Ne faites aucune modification sur la configuration de la M120**.

## Installation du driver de la M120

Sous TIA :

Dans Appareil et réseaux, **installez** le driver nommé Datalogic Matrix 120 pour la caméra Matrix 120 de Datalogic : Catalogue du matériel/Autres appareils de terrain/en Profinet I0.



## Configuration de l’adresse IP et du nom de la M120

|  |  |
| --- | --- |
| **Un clic sur Non affectés permet de la relier à votre Tool**. Dans la copie d’écran ci-contre, deux API sont présents dans l’application. Cette action permet de la relier sur le réseau. **Renommez**-**la** DLA-M120\_N avec N : numéro du Tool.  Ex. : DLA-M120\_1, pour le Tool1. |  |

|  |  |
| --- | --- |
| **Indiquez** l’adresse IP 10.96.9.6X (61 s’il s’agit de la caméra du Tool1) pour la caméra M120 de Datalogic.  Nota : Pour voir les adresses IP cliquez sur l’icone .  **Attention : Bien respecter le nom de la DLA-M120\_X** (le X indique le numéro du Tool).  Un clic sur l’appareil permet d’ouvrir la configuration de ce matériel. |  |

## Adresses pour le dialogue avec la M120

|  |  |
| --- | --- |
| **Notez** les adresses des octets Input et Output par défaut sur le compte rendu. (Il n’est pas utile de les changer). |  |

Les Input et Output sont liés à l’API. Ce dernier tableau indique que la M120 peut fournir 64 octets (BYTE) à l’API qui correspondent aux chiffres du code-barres, et recevoir 8 octets de l’API (demande de prise d’image lorsque c’est l’API qui prend l’image).

## Identification des octets significatifs

|  |  |
| --- | --- |
| **Placer une boite dont vous avez repéré les 4 derniers chiffres du code-barres sous la caméra M120.**  Dans une table de visualisation, **affichez les octets** %IB126 à %IB150 (voire plus si nécessaire). On recherche les octets stockant les 4 derniers caractères du code-barres.  Avertissement : ces adresses sont liées à l’API et peuvent donc être différentes selon le Tool (et sa caméra) sur lequel vous êtes. C’est pourquoi on scrute une plage d’adresse assez grande pour récupérer les 4 derniers chiffres du code-barres.  Dans ce cas, le format le mieux adapté est l’affichage en caractères.  **Notez les adresses qui correspondent aux quatre derniers chiffres du code à barres du numéro de la boite**. |  |
| Dans la table des variables, créez les variables liées aux octets des caractères du numéro de boite :  T1-CarMilliersNoBoite  T1-CarCentainesNoBoite  T1-CarDizainesNoBoite  T1-CarUnitésNoBoite  (T1- identifie ici une variable du Tool1) |  |

## VÉrification du fonctionnement (À faire valider par l’enseignant) :

Les caractères, du code-barres de la boite sous la caméra M120, doit être visibles dans une table de visualisation. Les bytes (%IBn) significatifs, doivent être nommés.

## Sur le compte rendu

Notez les adresses des octets d’input et d’output du module M120

Notez les adresses des octets %IB qui correspondent aux chiffres du numéro de la boite, le chiffre des milliers, des centaines, des dizaines et des unités.

# Niveau 2

## Objectifs du niveau 2

L’objectif de ce niveau est de créer une FC pour indiquer la présence de la boite sous le lecteur M120 et de donner le numéro de la boite en fonction des quatre derniers chiffres du code-barres de la boite lu par la caméra M120.

## Insertion d’un bloc FC

|  |  |
| --- | --- |
| **Ajoutez** un nouveau bloc FC nommé **NoBoiteParM120** en langage **SCL**.  **Déclare**z :  Quatre variable Input, nommée CarMilliers, CarCentaines, CarDizaines et CarUnités en BYTE. |  |

|  |  |
| --- | --- |
| La variable Output en Bool, PresenceBoite, pour indiquer la présence de la boite.  La variable InOut en Int, NoBoite pour donner le numéro de boite calculé.  La variable Temp en Bool, FrontPres, utilisé pour le front montant. |  |

## Insertion et renseignement de FC dans l’OB1

|  |  |
| --- | --- |
| **Insérez** l’appel de FC dans le réseau de l’OB1.  **Renseignez** les quatre entrées avec les quatre octets des chiffres du numéro de la boite lue sur le code-barres.  **Renseignez** la InOut par le mot %MW100 que l’on renommera par T1-NumBoite.  **Renseignez** la variable Output PrésenceBoite, par le bit M16.7.  **Chargez l’API**.  **Vérifiez** (icone lunettes) dans le réseau de l’OB1 que les valeurs des caractères correspondent à ceux du numéro de la boite. |  |

## Visualisation des octets significatifs

|  |  |
| --- | --- |
| * Dans la table de visualisation, **demandez l’affichage des octets des chiffres du numéro de la boite en Caractère et en DEC**. * **Constatez** que le chiffre ‘0’ en ASCII correspond à la valeur 48, le chiffre ‘1’ à 49, etc. |  |

## Bit de prÉsence de la boite

**Programmez** les lignes SCL de la FC16 permettant de de forcer la variable PrésenceBoite de sortie Bool indiquant la présence de la boite si le caractère des unités est significatif (supérieur à B#16#00).

Nota : si Toto est un Bool et Titi un Int, il est possible d’écrire : Toto := Titi <567 ; Dans ce cas, Toto est vrai (à l’état 1) tant que Titi est inférieur à 567.

## Extraction du numÉro de la boÎte

La FC16 doit permettre d’obtenir le numéro de la boite en fonction des 4 derniers caractères du code-barres. Il faut prendre en compte les différentes conversions :

### La conversion d’un TYPE CHAR en TYPE dÉcimal

Pour obtenir la valeur en décimal d’un chiffre exprimé en CHAR, il suffit de retrancher la valeur 48 à la valeur du caractère. En effet, si le chiffre est ‘4’, sa valeur ASCII est 52, en retranchant 48 on obtient bien la valeur 4.

### La conversion des Bytes en Int

Les variables d’entrée (recevant les %IBn) sont des types BYTE, il faut les convertir en type INT, la fonction **BYTE\_TO\_INT** permet cette conversion. (Il n’est pas possible de retrancher 48 à une variable en Byte). Par exemple pour convertir la variable en byte en une variable en Int :

Variable en Int := BYTE\_TO\_INT(IN := Variable en Byte) ;

## Programmation d’un front montant

Le front montant permet de ne faire le calcul que si la boite est présente, ce qui évite de faire le calcul avec des caractères qui ne sont pas des chiffres. De plus, lorsque la boite n’est plus sous le lecteur, on souhaite que le dernier numéro lu reste dans la variable de retour.

**Déclarez** un bit  FrontPres dans les variable Temp, et au front montant de PrésenceBoite.

**Rédigez** le programme permettant de calculer le numéro de la boite en fonction des variables d’entrée. Il suffit pour cela d’additionner les milliers, les centaines, les dizaines et les unités, pondérées par 1000, 100, 10 et 1.

## Sur le compte rendu

Ecrivez les lignes de FC16 et le réseau de l’OB1

## Vérification du fonctionnement (à faire valider par l’enseignant) :

Vérifiez (cliquez sur dans l’OB1) que le bit associé à la sortie PrésenceBoite passe à 1 lorsqu’une boite est sous la M120.

Vérifiez que la variable (%MW100) prend le numéro inscrit sur le code-barres de la boite.

Vérifiez que le bit PrésenceBoite passe à 0 lorsqu’il n’y a plus de boite sous la M120 et que son numéro est rémanent (garde sa valeur alors qu’aucune boite n’est sous le lecteur).

# Niveau 3

## Objectifs de ce niveau

* Affichage d’une variable en type REAL, %MD200, que l’on nommera "T1-PoidsBoite",
* Affichage du numéro de boite lu sur le code-barres "T1-NumBoite", %MW100 (calculée au niveau précédent)
* Allumer un voyant selon l’état de "PrésenceBoite", %M16.7 (du niveau précédent)
* Transférer un INT et un REAL du Tool1 vers le Tool4.

## Insertion d’un HMI

|  |  |
| --- | --- |
| Dans Appareils & Réseaux, à partir du catalogue matériel, **insérez**, dans Appareils et réseau, un HMI TP700 confort doté de l’adresse indiquée sur l’HMI. |  |

## CrÉation de champs E/S

|  |  |
| --- | --- |
| **Insérez** une vue dans laquelle placez deux champs E/S, en mode sortie, pour y faire afficher le poids de la boite et son numéro.  **Indiquez** le format d’affichage du nombre :  Pour le poids : s999.99 (avec signe)  Pour le numéro : 9999 (sans signe, sans décimale). |  |

Pour les essais, une boite sera placée sous le lecteur, le poids sera introduit sur le HMI.

## Voyants

|  |  |
| --- | --- |
| Dans une vue de l’HMI, **placez un voyant** (un cercle) qui devra être vert quand la boite est sous la M120 (PrésenceBoite %M16.7) et gris dans le cas contraire.  **Transférez et vérifiez** que le voyant s’allume lors de la présence de la boite. |  |

## Insertion d’un autre API (Tool4)

Dans les Appareils et réseaux, **insérez** un deuxième API CPU1512C nommé Tool4. **Reliez-le au réseau** et **indiquez** **l’adresse IP** notée sur la CPU. **Modifiez** les adresses E/S (1000, 0 et 2). (Le Tool4 ne possède pas de carte de communication).

## Chargement dans le Tool4 et vÉRIFICATION DU FONCTIONNEMENT

**Vérifiez** que le Tool4 passe bien en RUN après le chargement. Et, dans une table de visualisation, faites allumer et éteindre les lampes. (Si elles ne s’allument pas, vérifiez que les adresses des cartes E/S sont correctes).

## DÉclaration du mot ÉvÉnement

**Déclarez** dans les variables du Tool1 le mot : "T1-Event" (%MW80) **en type WORD**. On utilisera le bit de poids 0 de ce mot qui se note : "T1-Event".%X0.

## Bit pour demander l’enregistrement en DB

|  |  |
| --- | --- |
| Dans l’OB1 du Tool1, **programmez** le réseau pour forcer le bit "T1-Event".%X0 pendant 1 seconde. |  |
| Dans l’OB1 du Tool1**, faites allumer** la lampe verte si le bit "T1-Event".%X0 est à 1 |  |

**Transférez et vérifiez** que lors d’une impulsion sur le bouton Start, la lampe verte s’allume pendant 1s (et que le bit "T1-Event".%X0 passe à 1).

## DÉclaration des variables du Tool4

**Déclarez** : T4-L Verte, T4-Orange, et T4-Rouge aux adresses %Q1.5, %Q1.6 et %Q1.7

**Déclarez** un mot %MD200 en Real, nommé T4-PoidsRecu et un autre %MW100 en Int, nommé T4-NoBoiteRecu.

## Envoi des donnÉes du Tool1 vers le Tool4

|  |  |
| --- | --- |
| Dans la rubrique Alarmes HMI, **ajoutez** un événement qui, à l’apparition du bit 0 de T1-Event, ("T1-Event".%X0), force la mise à 1 de la lampe orange du Tool4. |  |

**Faites éteindre** la lampe quand le bit "T1-Event".%X0 disparait.

**Chargez, et vérifiez** que lorsque vous appuyez sur Start du Tool1, la lampe verte du Tool1 et l’orange du Tool4 s’allument pendant 1 seconde. On remarquera qu’il y a un temps de réponse entre les deux mises à 1. Ce temps peut être réduit à 100ms en configurant le temps de réponse dans les variables HMI.

Lors de l’apparition du même événement ("T1-Event".%X0), **ajoutez** une opération ce calcul pour forcer %MW100 et %MD200 du Tool4 à la valeur du numéro de boite et au poids au Tool1.

## VÉrification du fonctionnement (À faire valider par l’enseignant) :

Lorsque le bouton Start du Tool1 est enfoncé, la lampe orange du Tool4 doit s’allumer et les mots %MW100 et %MD200 (du Tool4) doivent prendre la valeur du numéro de boite et du poids introduit sur le HMI du tool1.

# Bonus A

## Objectifs de ce niveau

On propose d’enregistrer tous les poids envoyés par le Tool1 dans une base de données (DB) pour chaque numéro de boite.

## CrÉation du DB

|  |  |
| --- | --- |
| Dans la rubrique Ajouter un bloc, demandez la création du DB nommé DB Tool4 dans lequel définissez une variable tableau nommé PoidsReçu en Real de 0…1999 |  |

## CrÉation de la FC pour enregistrer les poids reçus

|  |  |
| --- | --- |
| Créez dans la même rubrique une fonction FC nommée EnregEnDB et déclarez les variables suivantes :  Input  NoBoiteReçu : Int ; pour y stocker le No de boite reçu  Poids Reçu : Real ; pour y stocker le poids reçu |  |

## Programme de la FC

**Rédigez** le programme permettant d’enregistrer la variable Input Poidsreçu dans la variable tableau à la position donnée par le numéro de boite reçu.

## VÉrification du fonctionnement (À faire valider par l’enseignant) :

**Placez une boite sous la M120**, **puis sur la balance**, et **appuyez** sur Start, **répétez** cette opération pour 4 boites chargées de façon inégale (placez 1 ou 2 ou 3 ou 4 écrous à l’intérieur). **Ouvrez une VAT** et faites-y afficher les valeurs stockées en DB Tool4 et **vérifiez** que les poids sont cohérents avec les numéros de boite.

# Bonus B

Pour le Tool4, **insérez** le driver de la M120, **nommez-le** DLA-M120\_4, **reliez-le** au tool4 et donner lui son adresse IP (64), **repérez** les adresses **%I** et **%Q**, copiez le bloc de lecture de la boite du Tool1 dans les blocs du Tool4, insérez l’appel dans l’OB1 du Tool4 pour obtenir le numéro de la boite lue au Tool4 sur le mot %MW102.

**Nota** : **Il n’est pas possible de relier le DLA M120 à l’API s’il est en ligne**.

Chargez et vérifiez que MW102 prend le numéro de la boite lu au Tool4.

|  |  |
| --- | --- |
| Ajoutez une variable Real en Output de la FC16 du Tool4 nommée PoidsBoiteSousM120.  Ajoutez la ligne dans la FC16 :  #PoidsBoiteSousM120 := "DB Tool4".PoidsRecu[#NoBoite];  Dans l’exemple ci-contre, la boite N°1431 pesée au Tool1, est lue au Tool4, le poids affiché est celui qui a été enregistré pour ce No de boite. |  |

## VÉrification du fonctionnement (À faire valider par l’enseignant) :

**Faites plusieurs enregistrements** **et vérifiez** que le poids donné par FC16, pour cette boite, est bien celui pesé au Tool1 pour cette boite.