

Envoi de Données

Deboxeur robotisé



mAGIC DRAW UMl

En premier lieu, on va utiliser la fonction dans **corresp.modèle** l’outil **TrainPatMaxPattern** afin de pouvoir repérer notre pièce par la caméra.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

On remplit comme paramètre la Région du modèle ou on place un cercle dans le centre de notre pièce.

Une image contenant cercle, jaune, Caractère coloré, capture d’écran

Description générée automatiquement

Et comme origine du modèle le centre de notre pièce.

Une image contenant cercle, jaune

Description générée automatiquement

On peut voir que notre fonction est bien définie.



Ensuite on va utiliser l’outil **FindPatMaxPatterns**.

Qui va permettre de chercher et trouver la pièce dans un zone choisi.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

On va en premier définir la région de recherche de notre outil

Une image contenant cercle, motif, jaune

Description générée automatiquement

Il est recommandé de choisir une région assez grande afin de ne pas rencontrer de problème lors du déclenchement de la pièce.

On choisit notre modèle qu’on a défini juste avant afin de définir quel gabarit on utilise.



Et on peut voir que nos coordonnées sont définies (en pixel).

Maintenant qu’on définit notre valeur en Pixel, il faut les valeurs réelles donc il faut tout d’abord définir le calibrage de la caméra.

Dans le mode Tableur de In-Sight Explorer, on vient rechercher dans la palette sur la droite et choisir **CalibrateGrid** et on glisse la fonction dans une cellule du tableur.



Un onglet s’affiche **configuration**, on choisit l’espacement entre les carrées, personnellement j’ai choisi 10mm.

**Attention ! Imprimez la grille de calibration si ça n’est pas fait.**

**Une image contenant texte, capture d’écran, ordinateur, logiciel

Description générée automatiquement**

Une image contenant motif, habits, Rectangle, tissu

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquementEnsuite dans la rubrique **Insérer**, notre grille est prise en compte par le logiciel car chaque carré est dans le tableau. Enfin on clique sur **calibrer**.

Pour finir avec notre calibration, dans **Résultat** on voit les nombres de points de la grille, les erreurs moyenne et maximal de la calibration et la marge d’erreur et on clique sur **OK**.

**Attention ! Si la marge d’erreur est trop élevée, il faut recommencer.**

Une image contenant texte, Appareils électroniques, capture d’écran, affichage

Description générée automatiquement

Une fois notre calibrage effectué, on va utiliser l’outil **TransPatternsToWorld** afin d’avoir les coordonnées en unité métrique.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquementEnsuite, en double cliquant sur **Calib**, le tableur s’ouvre et on sélectionne le calibre que l’on a effectué juste avant, ensuite dans **Structure Patterns**, on choisit le pattern que l’on a fait pour notre reconnaissance de forme.

On voit que dans le tableur, des valeurs se sont insérer dans des cellules.



Une fois tous ces éléments effectués, on va venir marquer dans une cellule **‘Point (0,0)**



Et on insère en dessous la commande cette fois ci, **TransPixelToWorld**

Une image contenant cercle, roue, jaune

Description générée automatiquementDans Calib, on prend le calibre que l’on a créée avant et dans l’onglet **Point**, on va choisir notre point (0,0). Mon point 0,0 sera situé en haut à droite de la caméra (zoomez pour être le plus précis possible).

Ce que ça doit ressembler après avoir tout remplie.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

L’aperçu dans les cellules :



Ensuite pour faire la conversion du pixel au réel il faut soustraire notre **Valeur X** de notre **Point 0** et la **Ligne** de la fonction **TransPatternsToWorld**

Et cela nous donne se résultat

**Attention ! Parfois des valeurs négatives peuvent apparaitre.**

Pour palier à ce problème on va utiliser la commande if()



Cette commande permettre d’avoir que des valeurs positives.

Explication de la commande :

Si notre **valeur X** du point 0 moins la valeur de **Ligne** dans **TransPatternsToWorld** est inférieure à 0 alors on fait la valeur de **Ligne** dans **TransPatternsToWorld** moins la **valeur X** du point 0 sinon on fait le calcul de base.

Et avec cette commande on va avoir que des valeurs positives.

On fait pareil pour Y mais on utile la **valeur Y** et la valeur de **Colonne** dans **TransPatternsToWorld.**



On trouve donc la valeur réelle de X et Y de notre pièce, la valeur réelle est en millimètres.

Pour faire une conversion pour avoir la valeur la plus proches de celle données de base.

Exemple :

Si on a une valeur de 72,324 en valeur réelle, il va donner la valeur de 73. Donc pour avoir la valeur la plus proche et donc être plus précis dans nos envoie de coordonnées à Automation Studio, j’ai fait cette ligne de commande.



Explication :

On soustrait la valeur réelle X à la valeur réelle X en entier (Trunc permet de convertir un nombre a virgule en entier) est supérieur ou égal à 0,500, alors on arrondi à l’entier supérieur sinon on laisse la valeur de base

On applique cette commande aussi à la valeur Y.

Envoie des coordonnées à Automation Studio

Pour que le programme puisse être envoyer à Automation Studio on va utiliser la commande **FormatOutputBuffer.** A l’intérieur de cette formule on va choisir le programme qu’on à effectuer juste auparavant.

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement

On choisit comme type de données un Nombre entier non signé de 16 bits

Ensuite on va utiliser la commande WriteEIPBuffer

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

On laisse l’évènement par défaut et on choisit le Tampon qu’on a créée juste avant.

On peut voir que c’est OK en regardant dans Automation Studio

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement