

Projet 23



R6 - AVRIL 2008



Etude d'une caméra intelligente



HÉLÈNE CUNRATH
FRANCIS MARTINEZ

BÉNÉDICTE DUPUIS
NICOLAS CHOISEL

Encadrant

Pierre Graebing

Sommaire



Introduction

Application

Cahier des charges

Planning

Budget

Présentation de la caméra

Connexion au système

Implémentation d'un programme

Traitement d'images

Notion de temps réel

Conclusion

Introduction

Présentation du client



Contrôle qualité

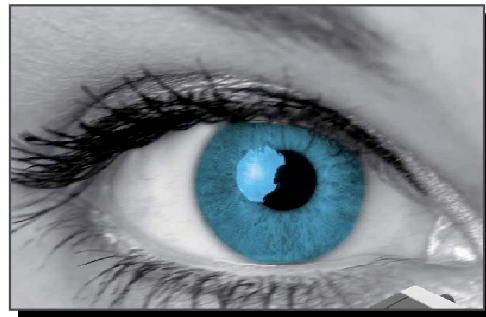
Détecter les défauts d'un
composant dans une chaîne de
production et intervenir
(éventuellement en temps réel)



Introduction

Problématique

Objectif : Remplacer l'opérateur humain par un système automatisé basé sur la vision par ordinateur

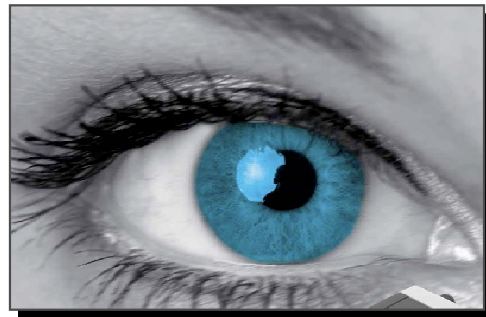


Système automatisé

Introduction

Problématique

Objectif : Remplacer l'opérateur humain par un système automatisé basé sur la vision par ordinateur



Système automatisé

Introduction

Problématique

Objectif : Remplacer l'opérateur humain par un système automatisé basé sur la vision par ordinateur



Système automatisé

Solution : Utilisation de caméras intelligentes?

Application



Exemple d'application:

contrôle qualité dans une
chaîne de montage



Surtout pour l'industrie



Permet de surveiller l'automatisation



Application

Détection de composants défectueux



Exemple d'application: détection de formes



Cahier des charges

Objectif



Cahier des charges

- ✦ Permettre la connexion au système de vision intelligente via Internet
- ✦ Mise en place d'une procédure d'implémentation de programmes sur le système
- ✦ Implémentation sur le système d'une bibliothèque simple de traitement d'images

Cahier des charges

- ✦ Optimisation du système pour une application en temps réel
- ✦ Etude de la possibilité d'utilisation des FPGA intégrés au système pour les applications en temps réel

A réaliser : Manuel de prise en main de la caméra

→ Formation du client et des futurs utilisateurs de la caméra

Cahier des charges

Liens avec les applications



Cahier des charges

Liens avec les applications

<i>Accès à distance</i>	Suivi du déroulement des opérations, récupération des données...
<i>Traitement d'images</i>	Détection d'anomalies
<i>Temps réel</i>	Intervention immédiate en cas de problème
<i>FPGA</i>	Contrôle qualité en temps réel

Présentation de la caméra

Principales caractéristiques

Plateforme
Linux intégrée

Système de
capture et de
stockage d'images



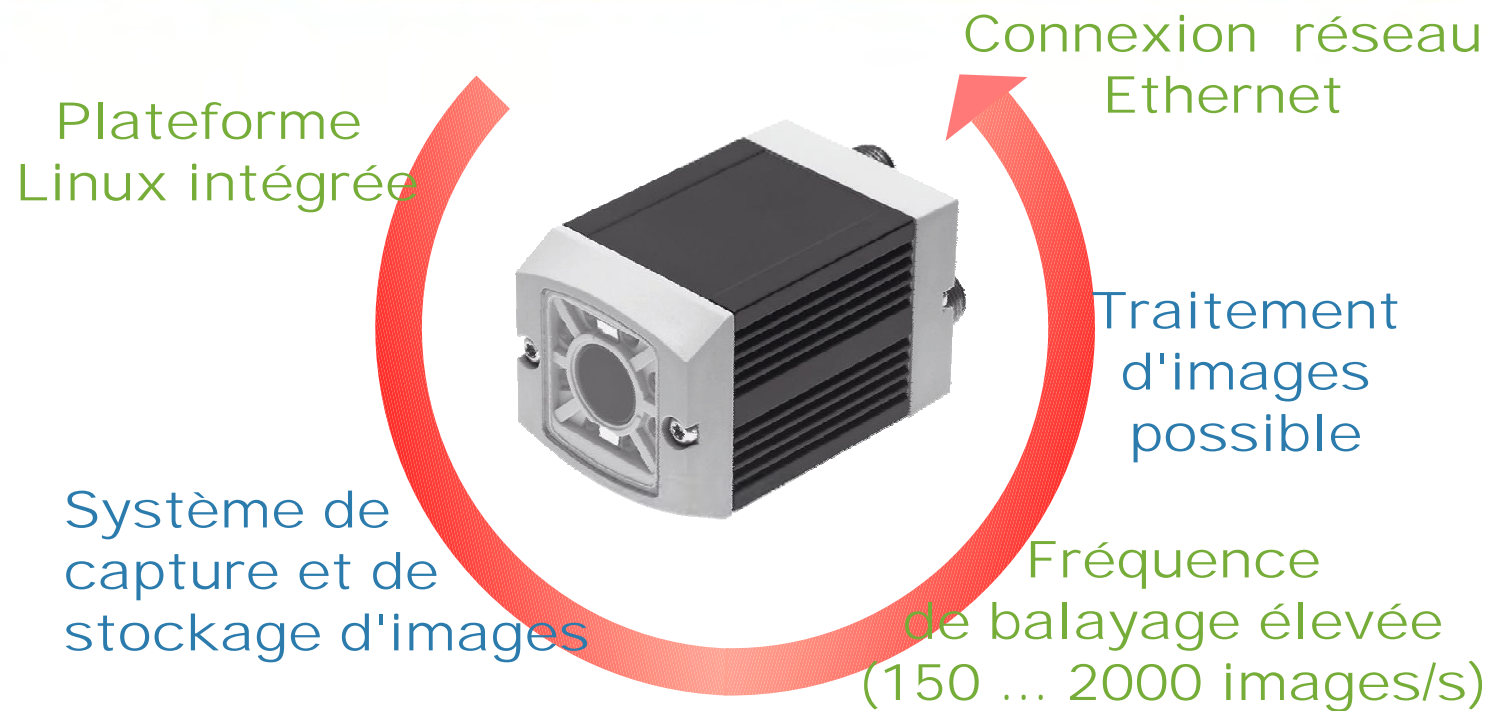
Connexion réseau
Ethernet

Traitement
d'images
possible

Fréquence
de balayage élevée
(150 ... 2000 images/s)

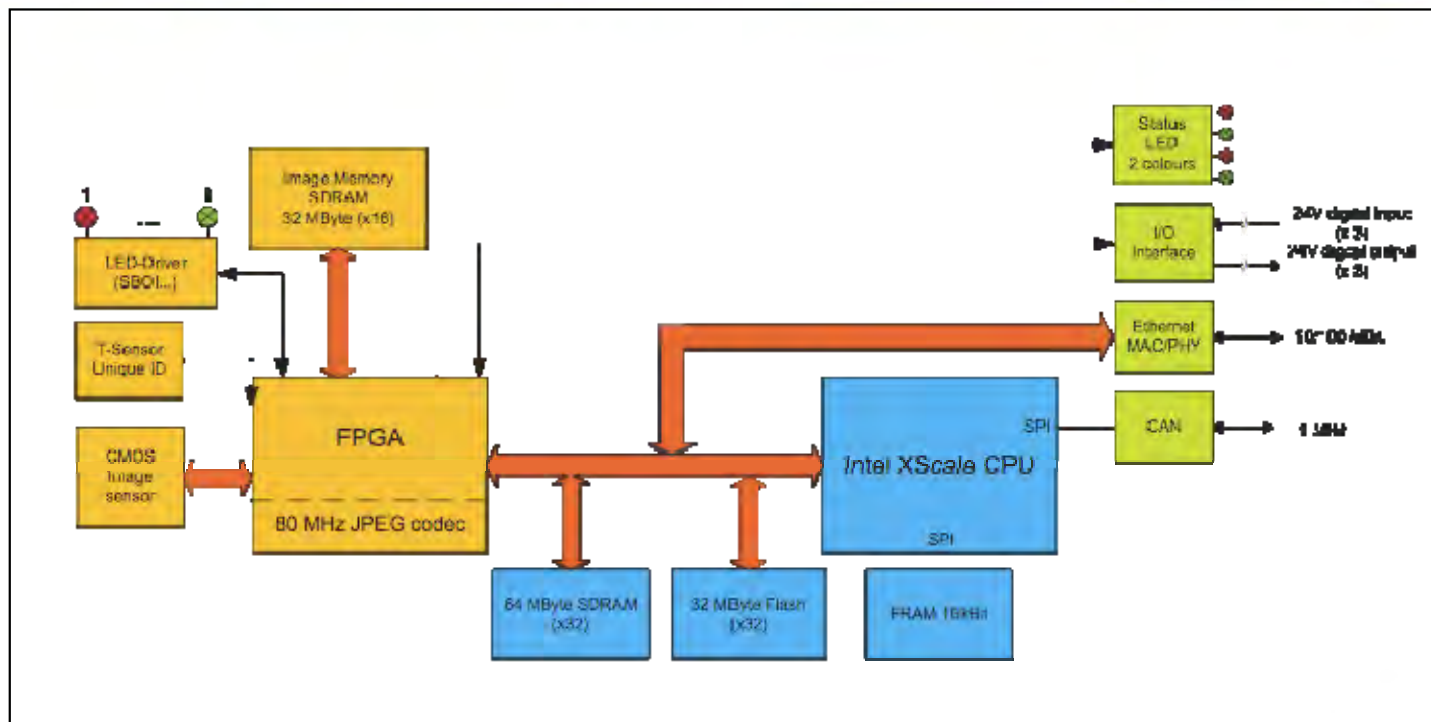
Présentation de la caméra

Principales caractéristiques



Présentation de la caméra

Diagramme



Déroulement du projet



Etape 1: Prise en main du système

Recherche documentaire sur l'architecture
du système



Etape 2: Connexion au système

En local, puis à distance



Etape 3: Implémentation d'un programme

Mettre en oeuvre les outils et les procédures de
programmation

Déroulement du projet

 Etape 4: Traitement d'images
Développement simple

 Etape 5: Temps réel
Utilisation des FPGA

Etape 1: prise en main du système



Etape 1: prise en main du système

Etape 1: prise en main du système

Etape 1: prise en main du système



Système inconnu

- comment alimenter le système?
- branchements?
- comment communiquer avec la caméra?

Documentation technique, oui mais pas suffisante

- rien sur la connexion depuis une plate-forme Linux



Etape 1: prise en main du système

Système inconnu

- comment alimenter le système?
- branchements?
- comment communiquer avec la caméra?

Documentation technique, oui mais pas suffisante

- rien sur la connexion depuis une plate-forme Linux

Etape 1: prise en main du système

✦ Système inconnu

- comment alimenter le système?
- branchements?
- comment communiquer avec la caméra?

✦ Documentation technique, oui mais pas suffisante

- rien sur la connexion depuis une plate-forme Linux

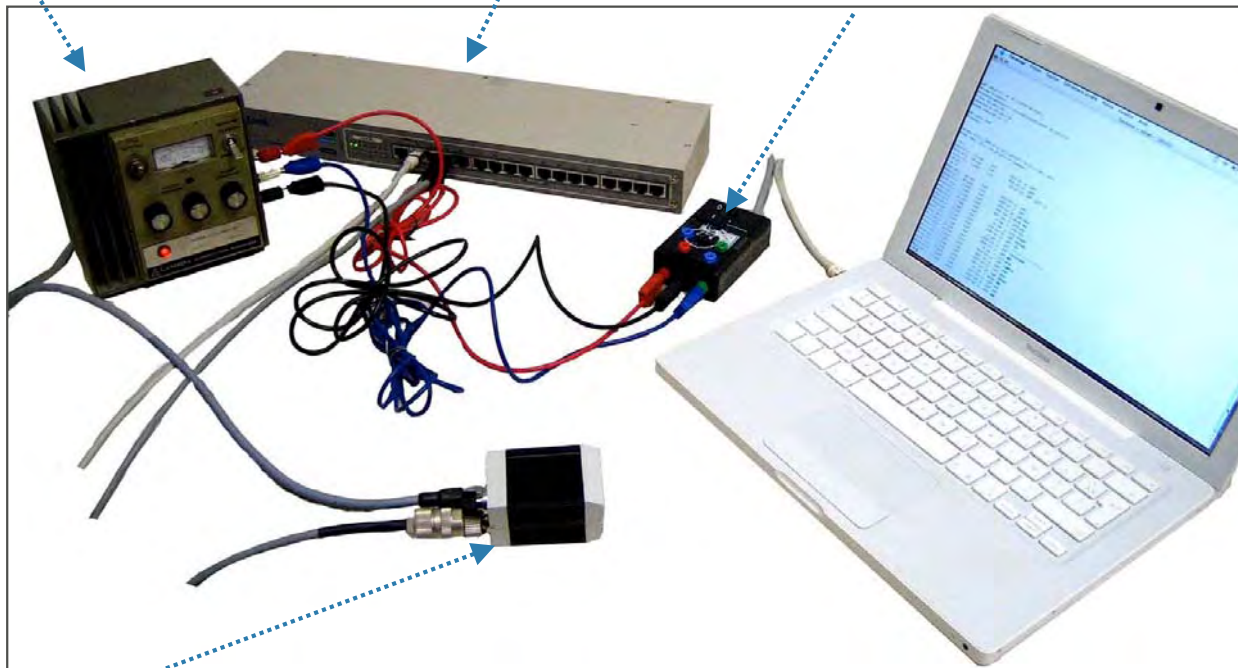
Etape 1: prise en main du système

Montage

Alimentation (24V – 300mA)

Hub

Boîtier de connecteurs bananes
(alimentation, triggers et sorties)



Caméra intelligente



Projet 23 Etude d'une caméra intelligente

30/04/08

18/50

Etape 1: prise en main du système

Enregistrement permanent

- ✦ Les modifications ne sont pas enregistrées...
 - faut-il tout charger à chaque réinitialisation?
 - existe-t-il une commande pour enregistrer?

- ✦ Sortir de la logique de l'ordinateur

Etape 1: prise en main du système

Mémoires du système

Deux types de mémoire

RAM

- racine du système (/)
- données effacées après réinitialisation

Flash

- /ffx et /ffxusr
- données enregistrées de manière permanente

Etape 2: connexion au système

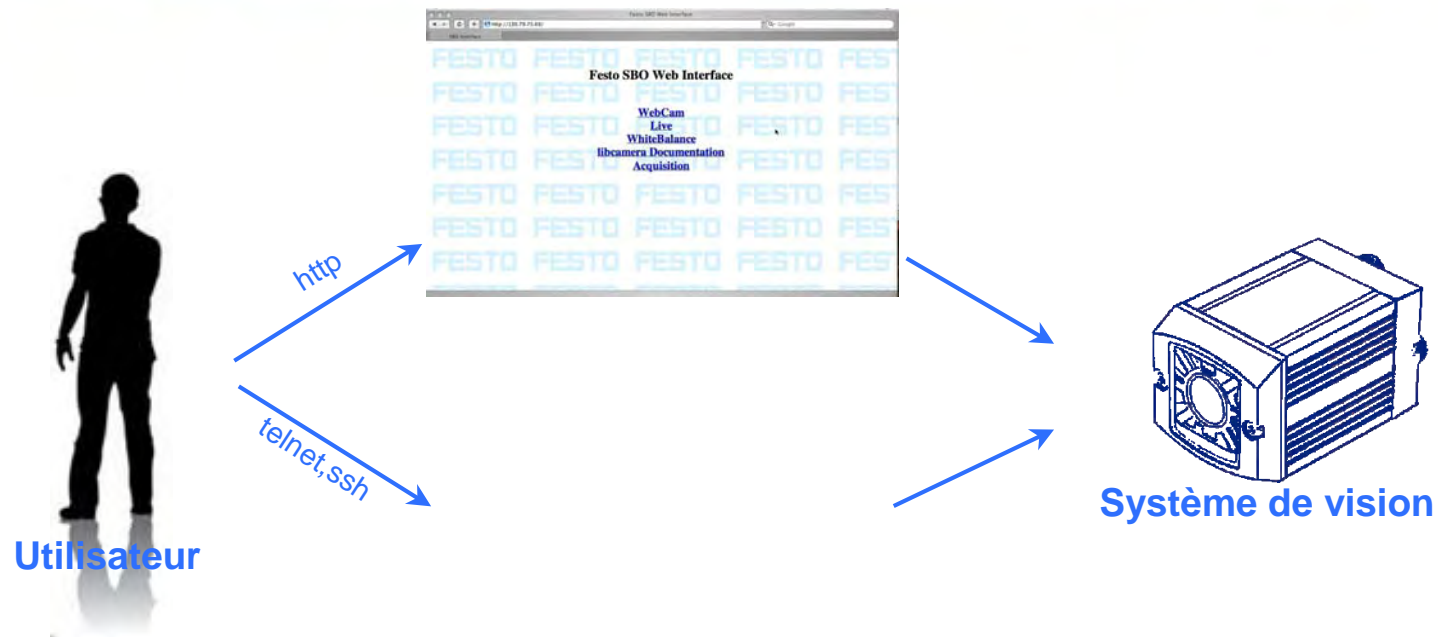


Etape 2: connexion au système

Etape 2: connexion au système

Etape 2: connexion au système

Deux modes de connexion



Etape 2: connexion au système protocole SSH

Telnet: protocole installé par défaut

Ftp: protocole de transfert de fichiers

- protocole sécurisé
- n'est pas présent sur le système par défaut

Etape 2: connexion au système protocole SSH

 **Telnet:** protocole installé par défaut

Ftp: protocole de transfert de fichiers

- protocole sécurisé
- n'est pas présent sur le système par défaut

Etape 2: connexion au système protocole SSH

✦ **Telnet**: protocole installé par défaut

✦ **Ftp**: protocole de transfert de fichiers

- protocole sécurisé
- n'est pas présent sur le système par défaut

Etape 2: connexion au système protocole SSH

✦ **Telnet**: protocole installé par défaut

✦ **Ftp**: protocole de transfert de fichiers

Protocoles peu sécurisés => conflits avec les firewalls

- protocole sécurisé
- n'est pas présent sur le système par défaut

Etape 2: connexion au système protocole SSH

✦ **Telnet**: protocole installé par défaut

✦ **Ftp**: protocole de transfert de fichiers

Protocoles peu sécurisés => conflits avec les firewalls

Solution: utiliser le protocole **SSH**

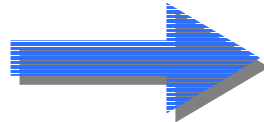
- protocole sécurisé
- n'est pas présent sur le système par défaut

Etape 2: connexion au système En réseau local

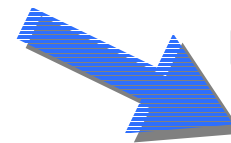


Système de vision
intelligente

IP 192 168 2 10



Hub



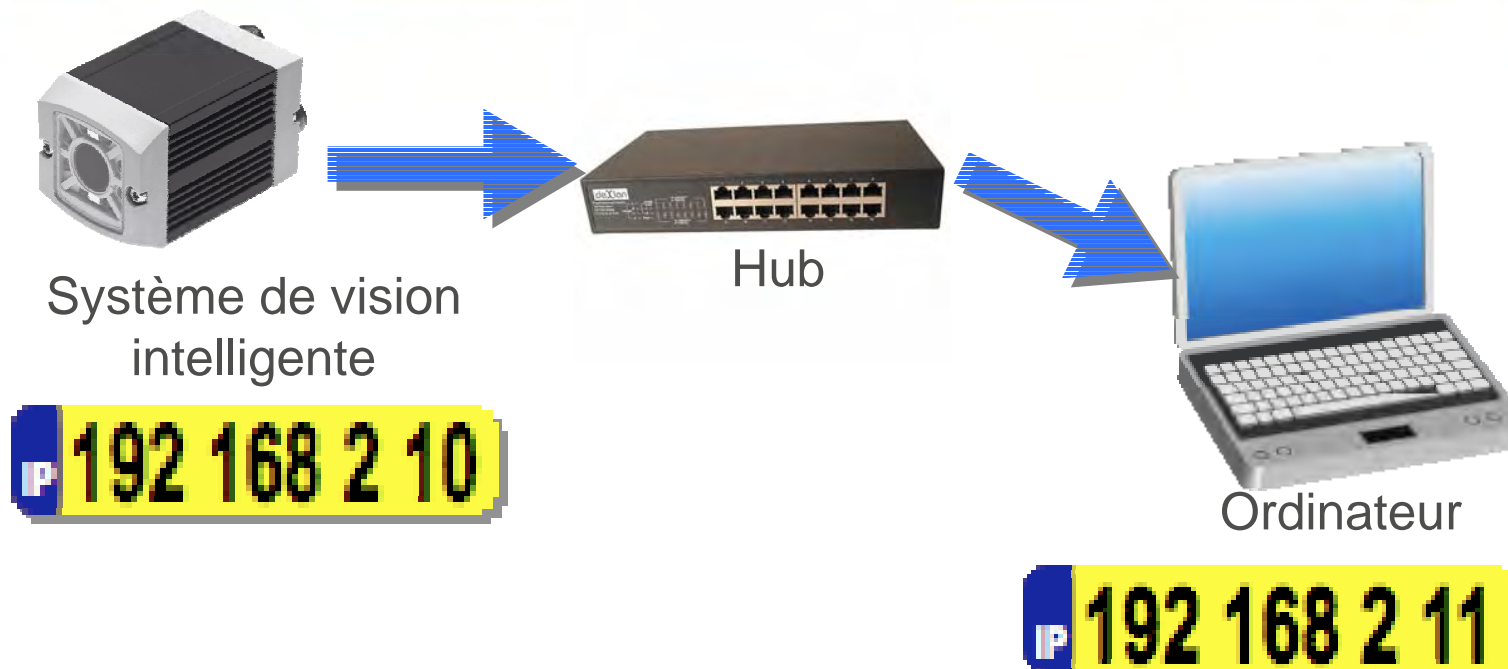
Ordinateur

IP 192 168 2 11



Adresses non routables

Etape 2: connexion au système En réseau local



✦ Adresses non routables

Etape 2: connexion au système A distance



Système de vision
intelligente

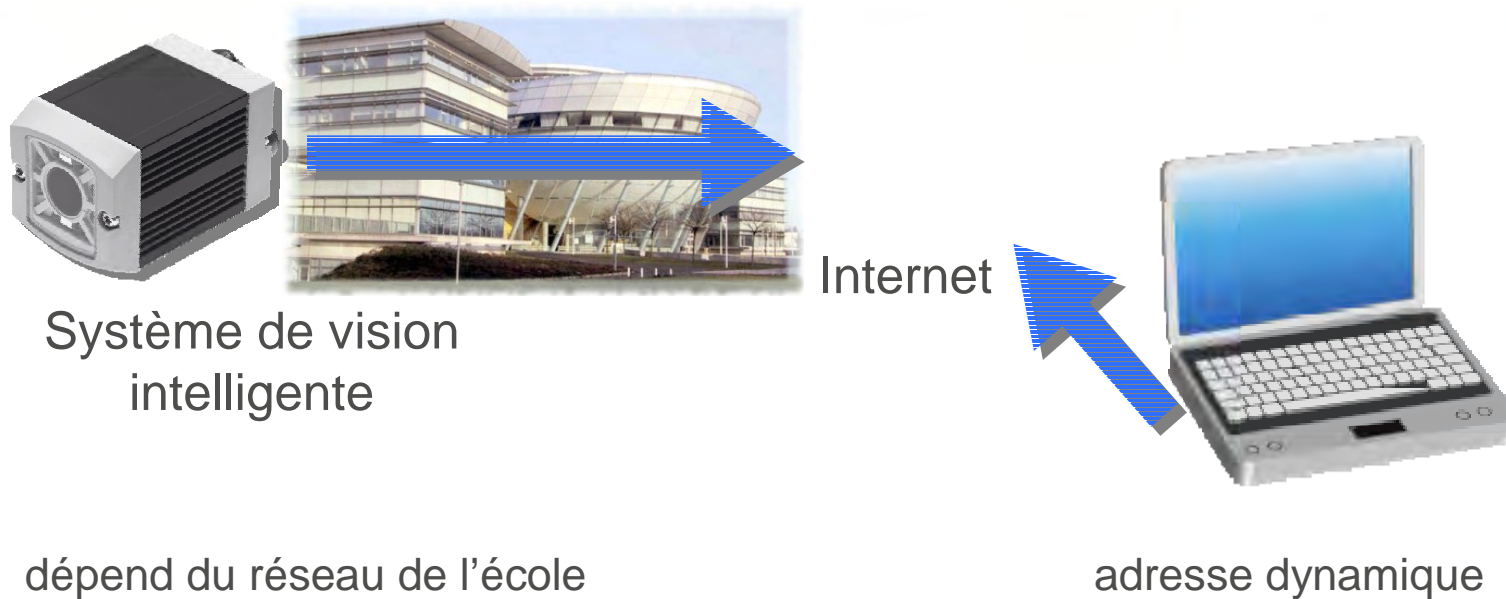
dépend du réseau de l'école

Internet

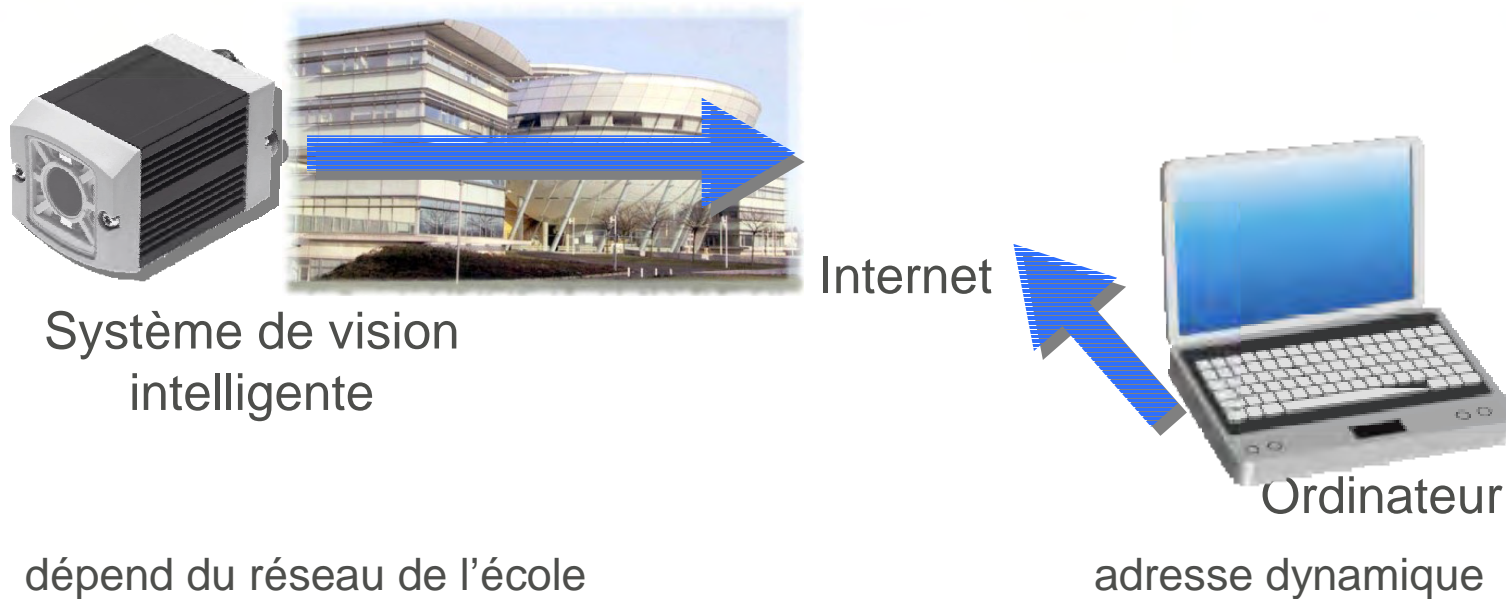


adresse dynamique

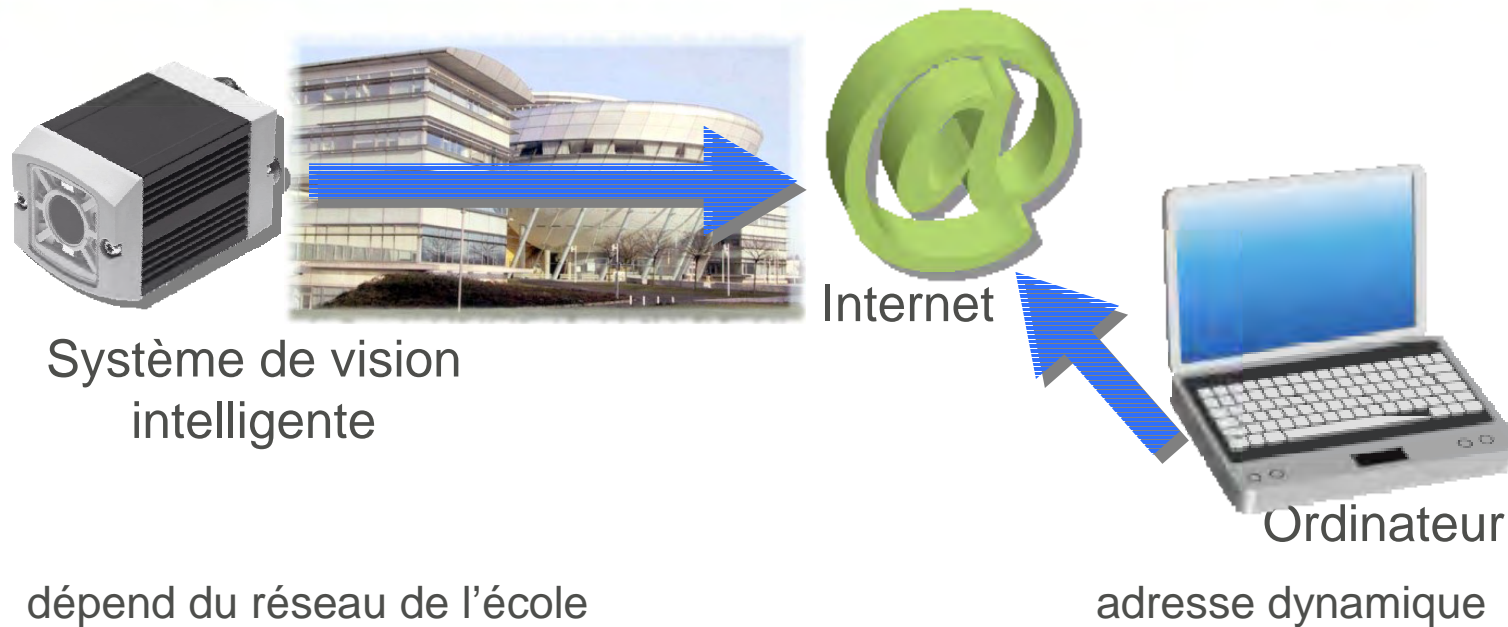
Etape 2: connexion au système A distance



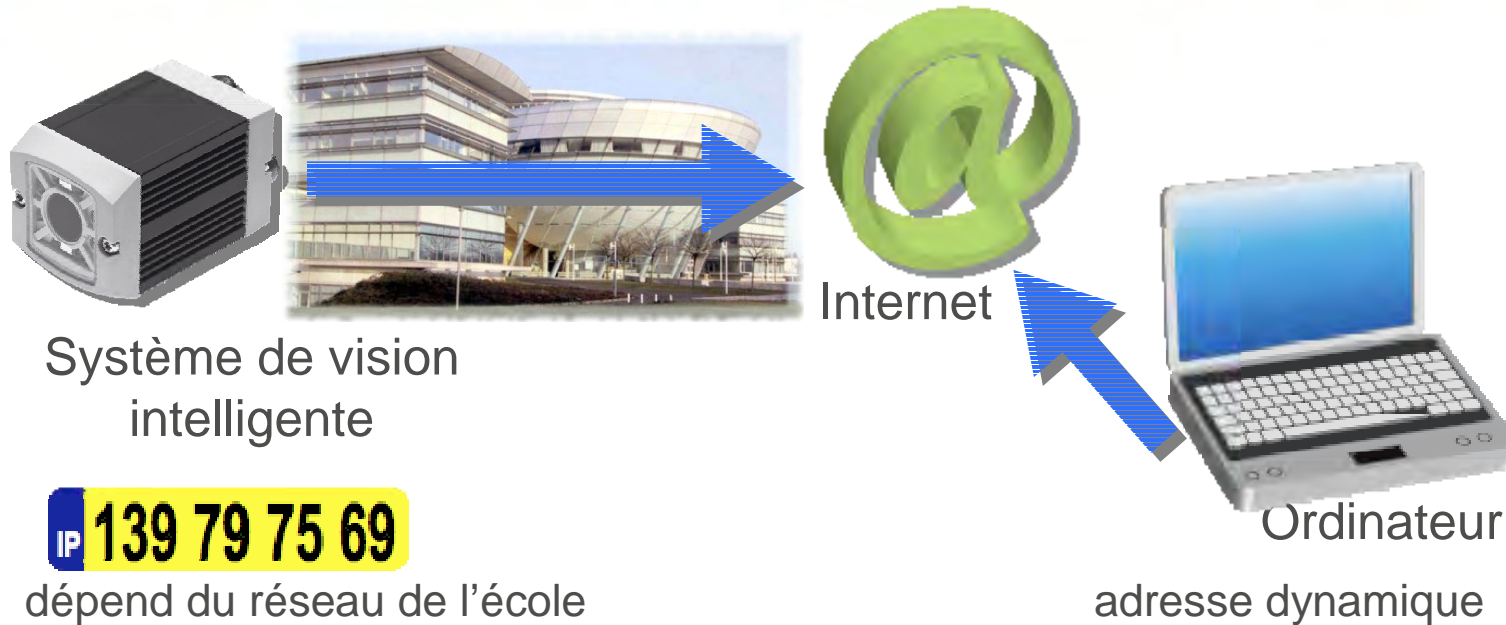
Etape 2: connexion au système A distance



Etape 2: connexion au système A distance



Etape 2: connexion au système A distance



Etape 3: implémentation d'un programme

Etape 3: implémentation d'un programme

Etape 3: implémentation d'un programme

Etape 3: Implémentation d'un programme



Un compilateur intègre les caractéristiques du système d'exploitation.

L'ordinateur et la caméra n'ont pas le même OS...

Comment compiler un programme sur l'ordinateur?

Etape 3: Implémentation d'un programme



Un compilateur intègre les caractéristiques du système d'exploitation.

L'ordinateur et la caméra n'ont pas le même OS...



Comment compiler un programme sur l'ordinateur?

Etape 3: Implémentation d'un programme



Un compilateur intègre les caractéristiques du système d'exploitation.

L'ordinateur et la caméra n'ont pas le même OS...

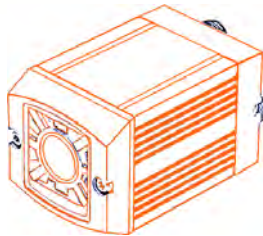


Comment compiler un programme sur l'ordinateur?

Solution: utiliser un compilateur croisé

Etape 3: Implémentation d'un programme

Compilation croisée



Cible

transfert
de l'exécutable

récupération
des données

```
#include <iostream>  
#include <stdio.h>
```

```
using namespace std;
```

```
int main()
```

```
//code
```

```
cout<<"hello world!";
```

```
return 0;
```

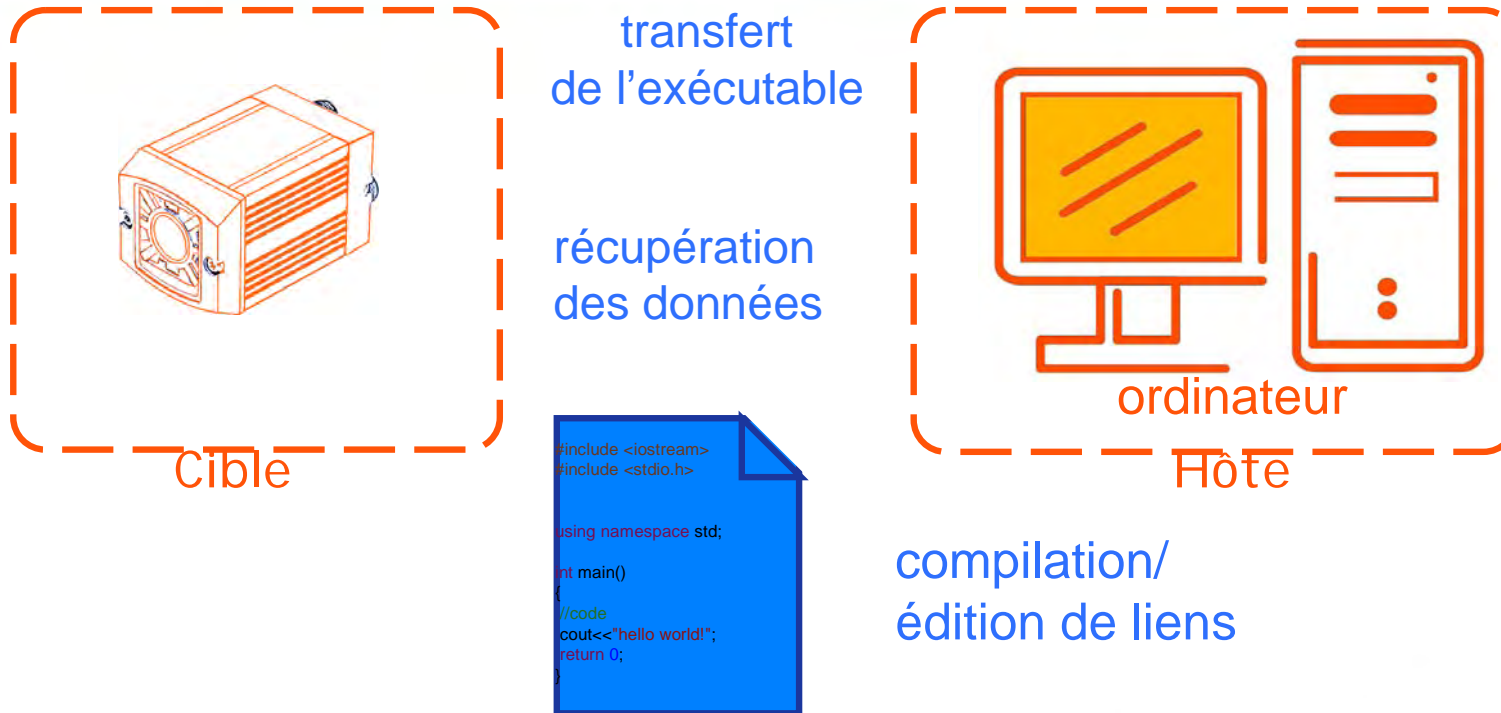


Hôte

compilation/
édition de liens

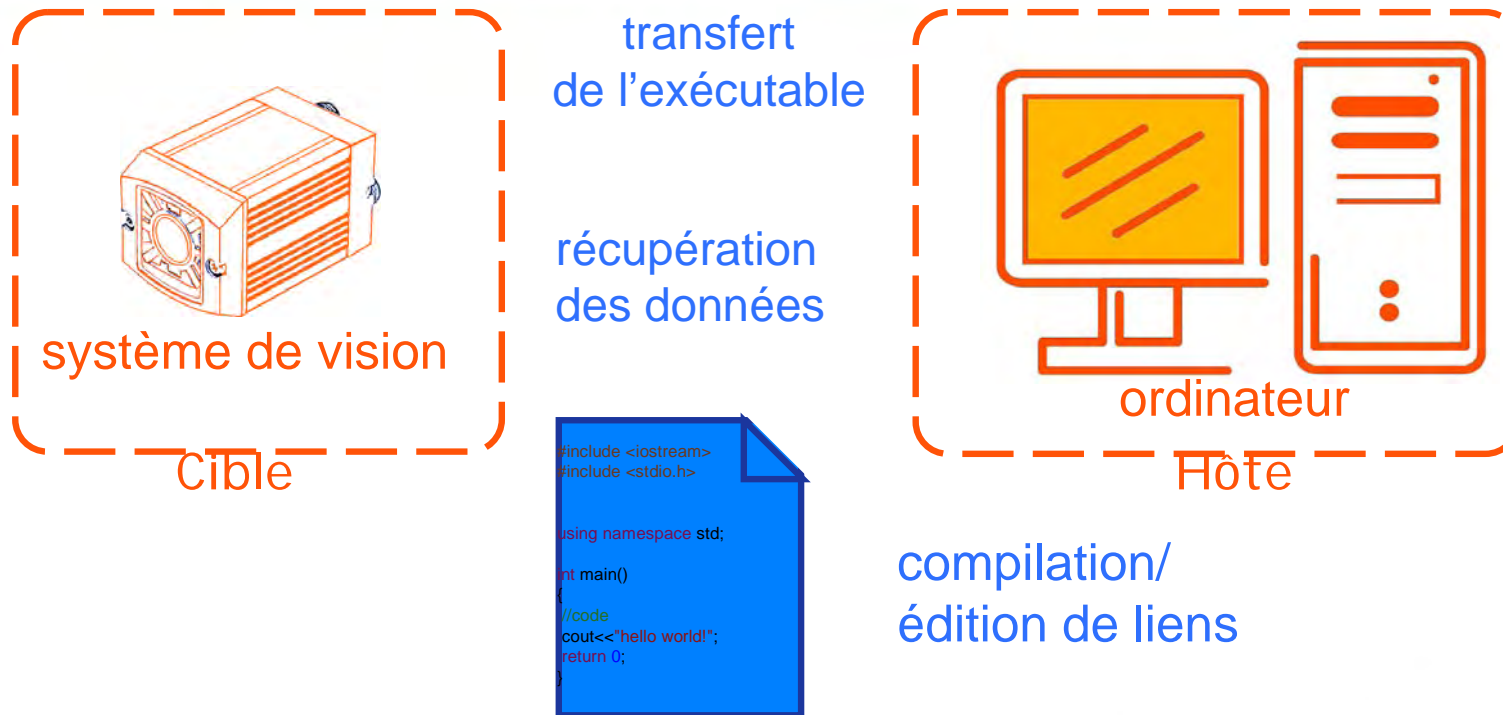
Etape 3: Implémentation d'un programme

Compilation croisée



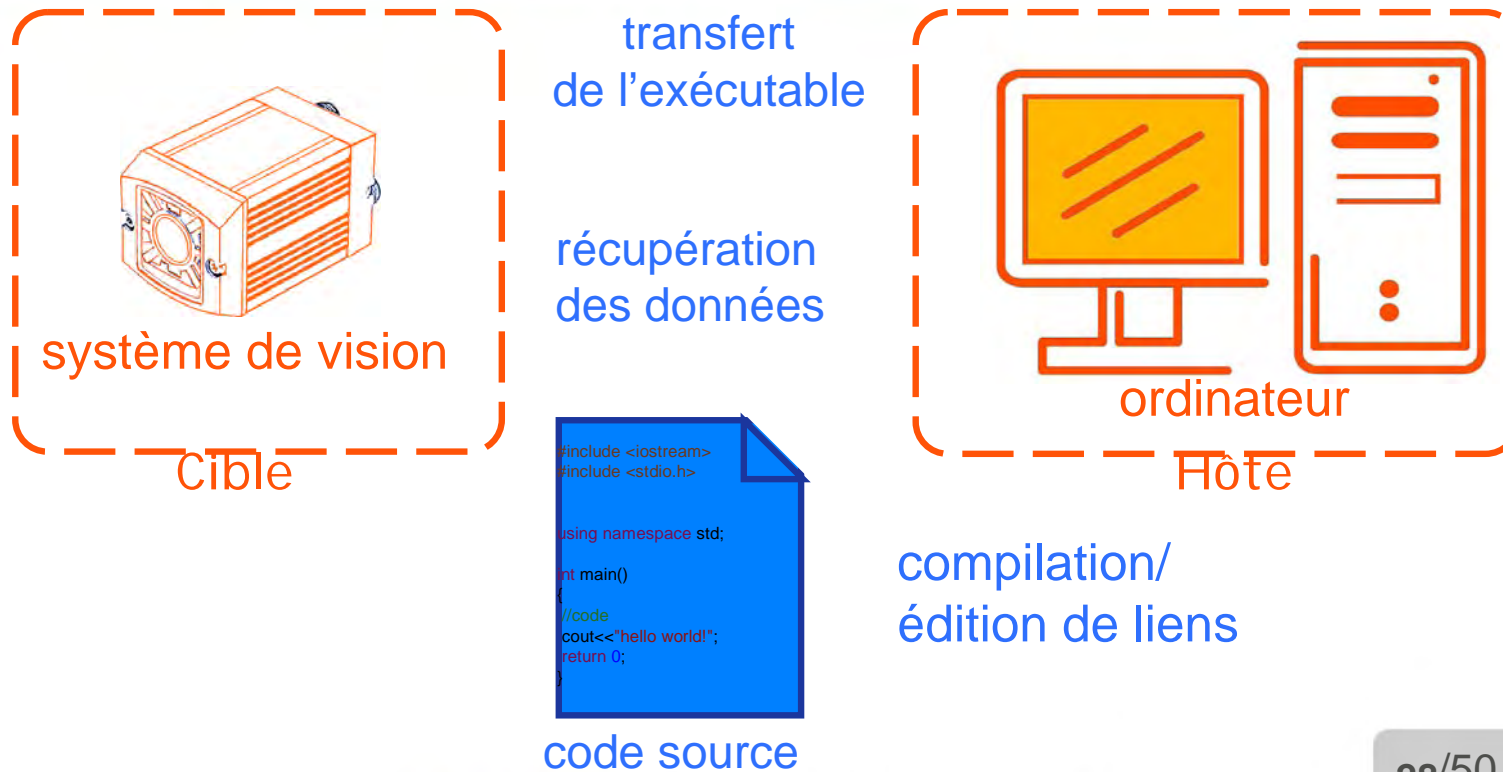
Etape 3: Implémentation d'un programme

Compilation croisée



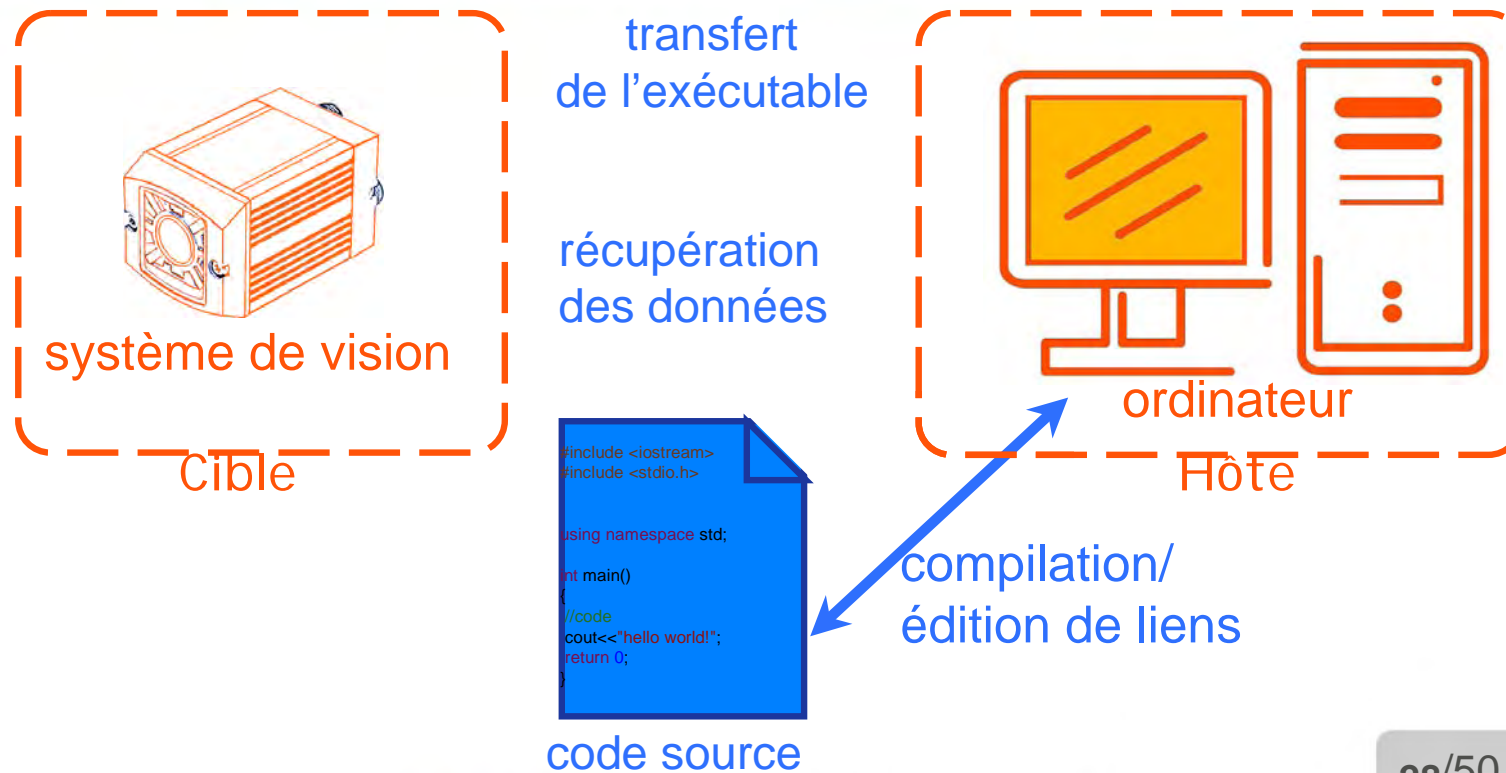
Etape 3: Implémentation d'un programme

Compilation croisée



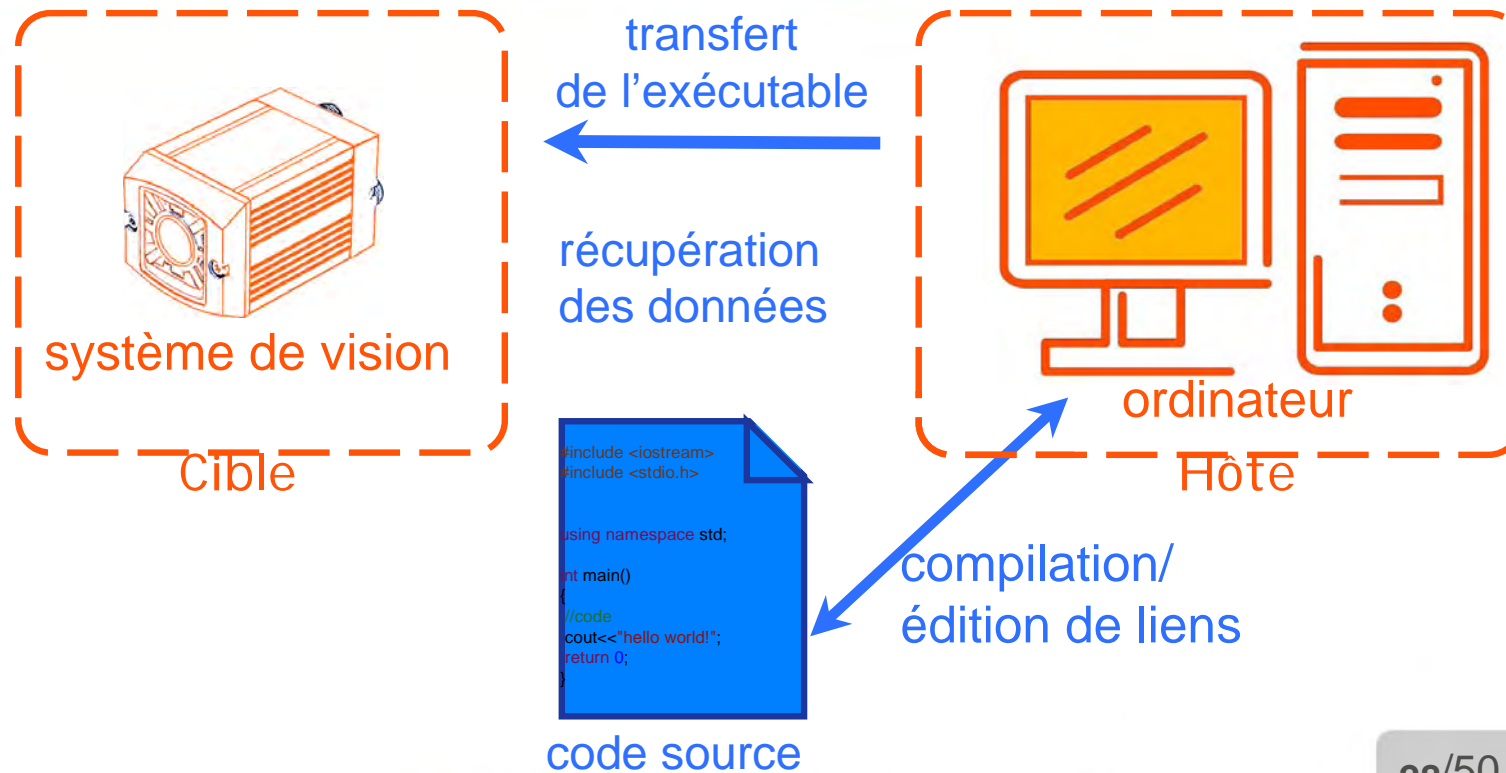
Etape 3: Implémentation d'un programme

Compilation croisée



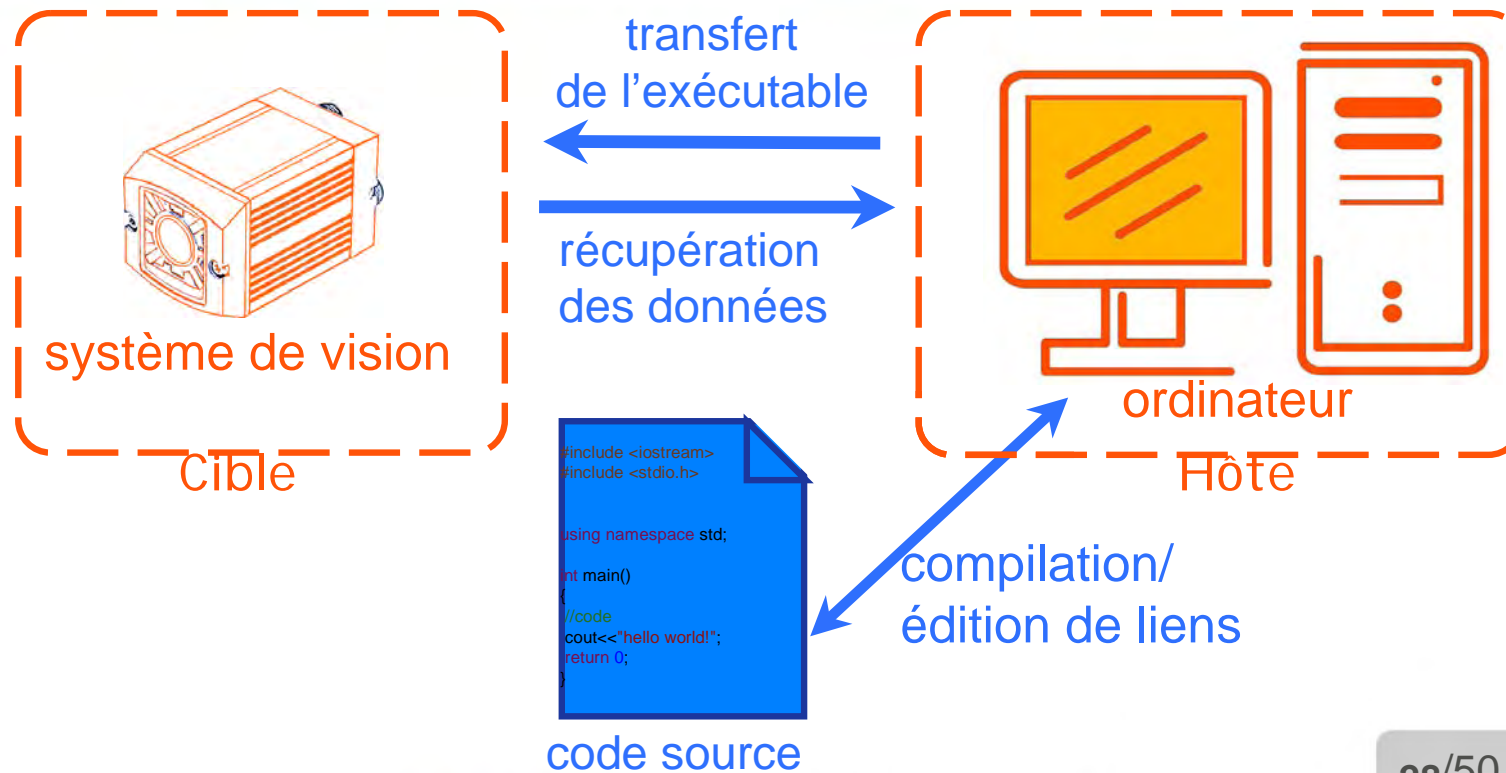
Etape 3: Implémentation d'un programme

Compilation croisée



Etape 3: Implémentation d'un programme

Compilation croisée



Etape 3: Implémentation d'un programme

Les premiers résultats

✦ Résultat des programmes d'essai:

```
Hello!
Actual LED-State 0x2 (D C B A)
  LED A green red = ON OFF
  LED B green red = OFF OFF
  LED C yellow red = OFF OFF
  LED D yellow red = OFF OFF
Actual LED-State 0xf0
  LED C yellow red = ON ON
  LED D yellow red = ON ON
Actual LED-State 0x90
  LED C yellow red = OFF ON
  LED D yellow red = ON OFF
Actual LED-State 0x10
  LED C yellow red = OFF ON
  LED D yellow red = OFF OFF
Goodbye!
```

allume successivement
les LED de l'appareil

```
/ffx $ ./SetOutput
```

Hello!

Actual Out
Out
Actual Out
Out
Actual Out
Actual Out

Goodbye!



teste les sorties
de la caméra

Etape 3: Implémentation d'un programme

Les premiers résultats

✦ Résultat des programmes d'essai:

```
Hello!
Actual LED-State 0x2 (D C B A)
  LED A green red = ON OFF
  LED B green red = OFF OFF
  LED C yellow red = OFF OFF
  LED D yellow red = OFF OFF
Actual LED-State 0xf0
  LED C yellow red = ON ON
  LED D yellow red = ON ON
Actual LED-State 0x90
  LED C yellow red = OFF ON
  LED D yellow red = ON OFF
Actual LED-State 0x10
  LED C yellow red = OFF ON
  LED D yellow red = OFF OFF
Goodbye!
```

allume successivement
les LED de l'appareil

```
/ffx $ ./SetOutput
```

Hello!

Actual Out
Out
Actual Out
Out
Actual Out
Actual Out

Goodbye!



teste les sorties
de la caméra

Etape 3: Implémentation d'un programme

Nous savons faire tourner un programme sur le système.



Faire tourner un programme utilisant les fonctions du système
(capture d'image, enregistrement,...)



Etape 3: Implémentation d'un programme

Nous savons faire tourner un programme sur le système.

Etape suivante:

Faire tourner un programme utilisant les fonctions du système
(capture d'image, enregistrement,...)



Etape 3: Implémentation d'un programme

Nous savons faire tourner un programme sur le système.

Etape suivante:

Faire tourner un programme utilisant les fonctions du système
(capture d'image, enregistrement,...)

Programme de traitement d'image

Etape 3: Implémentation d'un programme

Programme appelant la caméra



Interface avec la caméra

- Création de l'interface
- Ajustement des propriétés de la caméra (taille de la fenêtre, mode d'acquisition, temps d'exposition, gain ...)



Traitement de l'image

- Création d'un objet permettant de stocker l'image
- Détection des contours
- Conversion en JPEG, Bitmap...



Libération de la mémoire

- Libération de l'objet image
- Destruction de l'interface



Génération de la page HTML



Etape 4: traitement d'images

Etape 4: traitement d'images

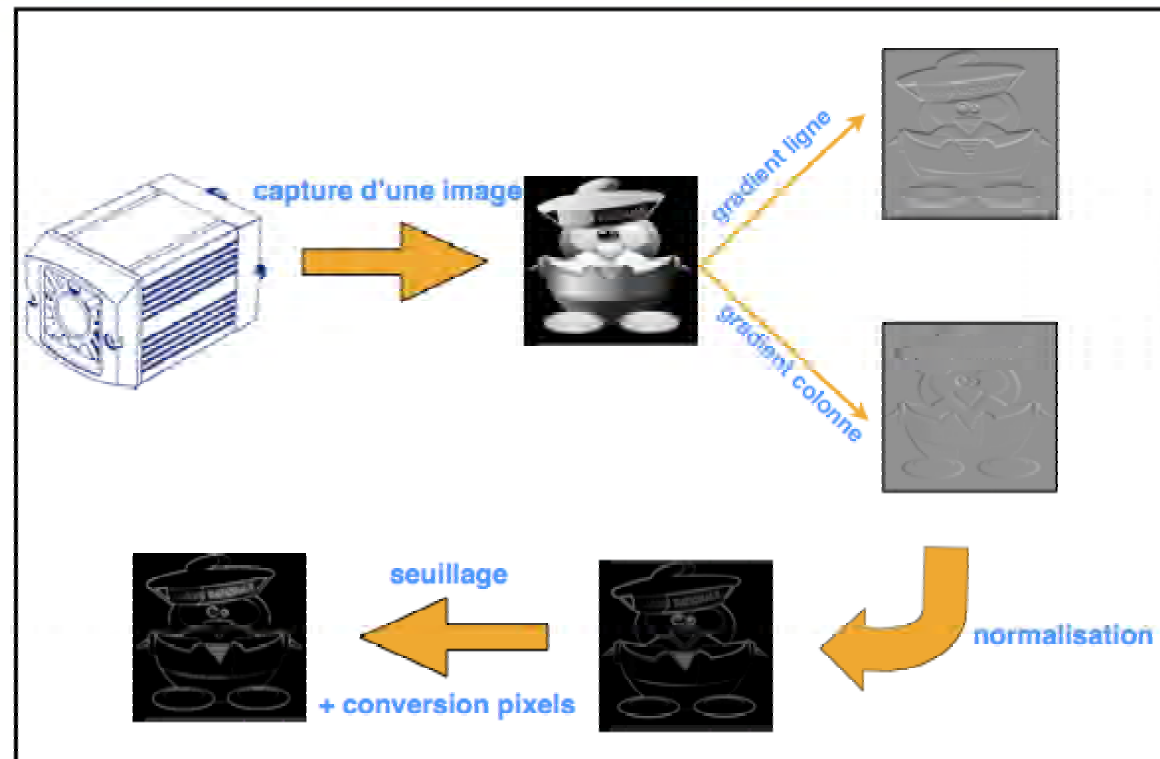
Etape 4: traitement d'images

Etape 4: Traitement d'images

- ✦ Réalisation d'un programme de détection de cercles pour en déterminer son diamètre
 Au début à la main....
- ✦ Détection de contours
- ✦ puis détection de cercles par transformée de Hough...

Etape 4: Traitement d'images

Recherche des contours d'une image



Etape 4: Traitement d'images

Temps d'exécution

	Temps d'exécution		
	Système de vision	Ordinateur	% de différence
Fréquence processeur	400MHz	2GHz	
programme 1	$2,400.10^{-3}$	$1,80006.10^{-4}$	7,50%
programme 2	$4,4632.10^{-2}$	$3,31402.10^{-3}$	7,43%
programme 3	$3,23719.10^{-1}$	$3,23949.10^{-2}$	10%

Programme 1 : « Hello World ! » Programme 2 : Seuillage
Programme 3 : Détection de cercles



Etape 4: Traitement d'images

Temps d'exécution

	Temps d'exécution		
	Système de vision	Ordinateur	% de différence
Fréquence processeur	400MHz	2GHz	
programme 1	$2,400.10^{-3}$	$1,80006.10^{-4}$	7,50%
programme 2	$4,4632.10^{-2}$	$3,31402.10^{-3}$	7,43%
programme 3	$3,23719.10^{-1}$	$3,23949.10^{-2}$	10%

Programme 1 : « Hello World ! » Programme 2 : Seuillage
Programme 3 : Détection de cercles



Programme trop complexe, non optimisé en temps
Utilisation de bibliothèques de traitement d'images

Etape 4: Traitement d'images

La librairie OpenCV



Open Source Computer Vision Library

Bibliothèque de traitement d'images complète
mais complexe (nombreux fichiers)

Possibilité de réaliser de la détection de formes

Etape 4: Traitement d'images

La librairie OpenCV



Open Source Computer Vision Library



Bibliothèque de traitement d'images complète
mais complexe (nombreux fichiers)

Possibilité de réaliser de la détection de formes

Etape 4: Traitement d'images

La librairie OpenCV

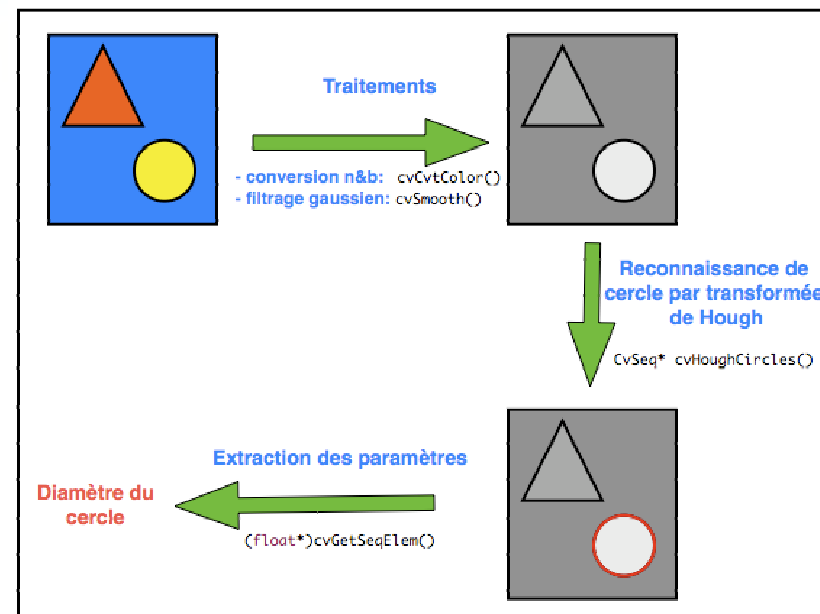


Open Source Computer Vision Library

- ✦ Bibliothèque de traitement d'images complète mais complexe (nombreux fichiers)
- ✦ Possibilité de réaliser de la détection de formes

Etape 4: Traitement d'images

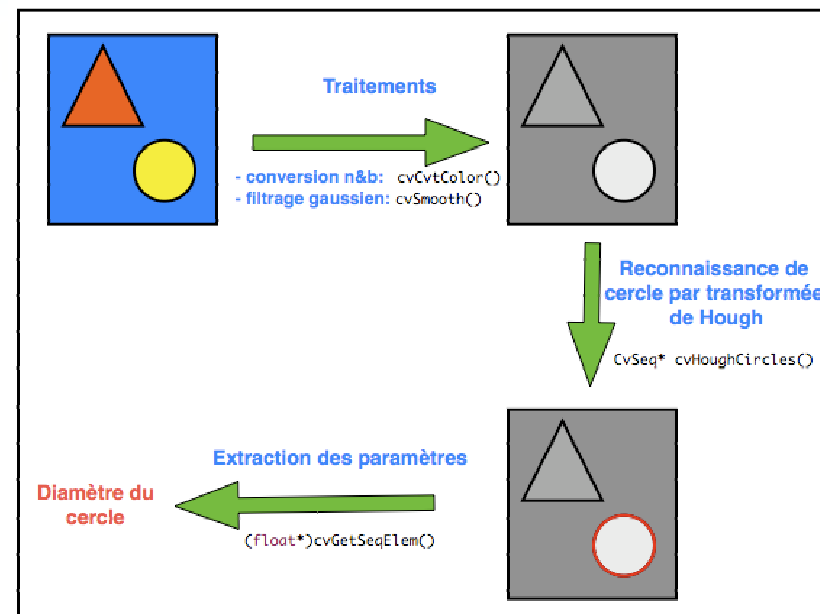
La librairie OpenCV



Temps d'exécution non optimisé

Etape 4: Traitement d'images

La librairie OpenCV



⊞ Temps d'exécution non optimisé

Etape 4: Traitement d'images

La librairie Camellia



Camellia
Image Processing &
Computer Vision library

Temps d'exécution optimisé

... et occupe moins de place mémoire

De nombreuses fonctions de traitement d'images

possibilité d'utiliser Camellia et openCV conjointement

On inclut la librairie à chaque compilation

```
~$ >g++ -Wl,libCamellia.dylib test.cpp -o test
```



Etape 4: Traitement d'images

La librairie Camellia



Camellia
Image Processing &
Computer Vision library



Temps d'exécution optimisé
... et occupe moins de place mémoire

De nombreuses fonctions de traitement d'images

possibilité d'utiliser Camellia et openCV conjointement

On inclut la librairie à chaque compilation

```
~$ >g++ -Wl,libCamellia.dylib test.cpp -o test
```



Etape 4: Traitement d'images

La librairie Camellia



Camellia

Image Processing &
Computer Vision library

- ✦ Temps d'exécution optimisé
... et occupe moins de place mémoire
 - ✦ De nombreuses fonctions de traitement d'images
- possibilité d'utiliser Camellia et openCV conjointement

On inclut la librairie à chaque compilation

```
~$ >g++ -Wl,libCamellia.dylib test.cpp -o test
```



Etape 4: Traitement d'images

La librairie Camellia



Camellia

*Image Processing &
Computer Vision library*

- ✦ Temps d'exécution optimisé
... et occupe moins de place mémoire
- ✦ De nombreuses fonctions de traitement d'images
- ✦ possibilité d'utiliser Camellia et openCV conjointement

On inclut la librairie à chaque compilation

```
~$ >g++ -Wl,libCamellia.dylib test.cpp -o test
```



Etape 4: Traitement d'images

La librairie Camellia



Camellia

*Image Processing &
Computer Vision library*

- ✦ Temps d'exécution optimisé
... et occupe moins de place mémoire
- ✦ De nombreuses fonctions de traitement d'images
- ✦ possibilité d'utiliser Camellia et openCV conjointement
- ✦ On inclut la librairie à chaque compilation

```
~$ >g++ -Wl,libCamellia.dylib test.cpp -o test
```

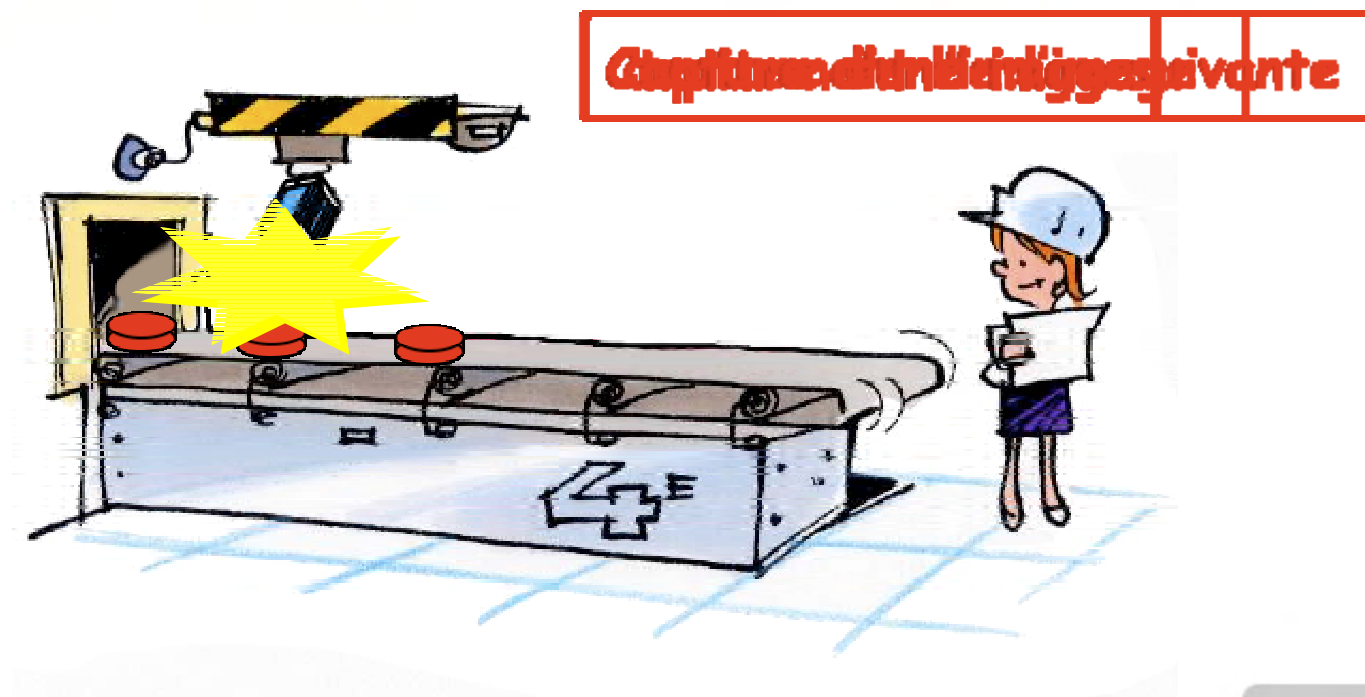


Résultats

Page HTML

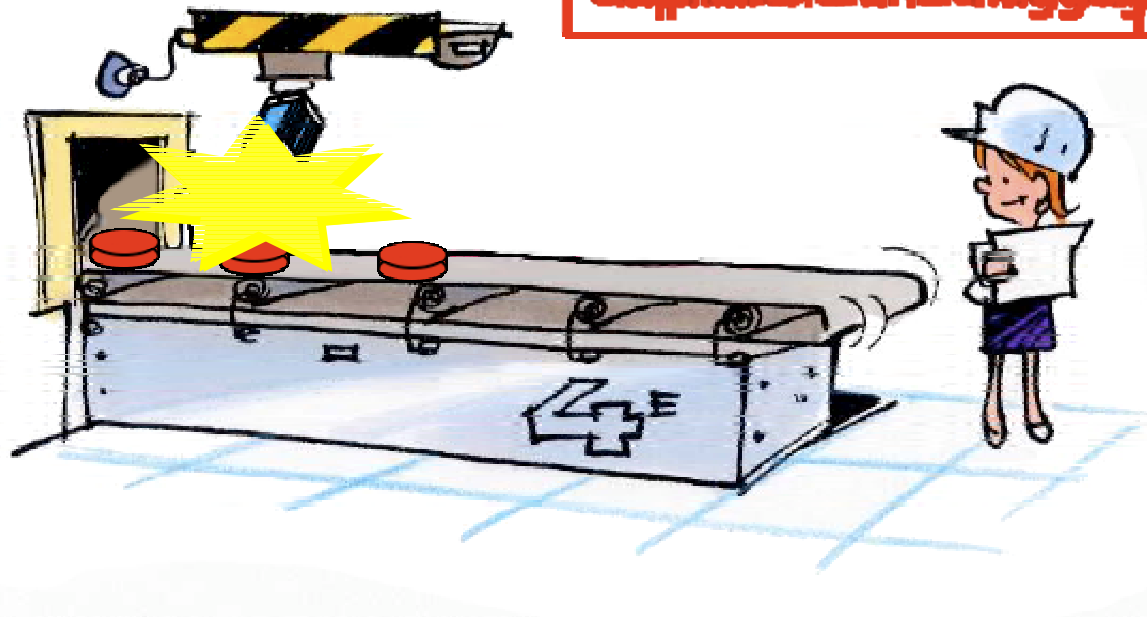


Notion de temps réel



Notion de temps réel

Chapitre 23 : Intelligence artificielle



Résultats

Démonstration



Résultats

Démonstration



QuickTime™ et un
décompresseur
sont requis pour visionner cette image.



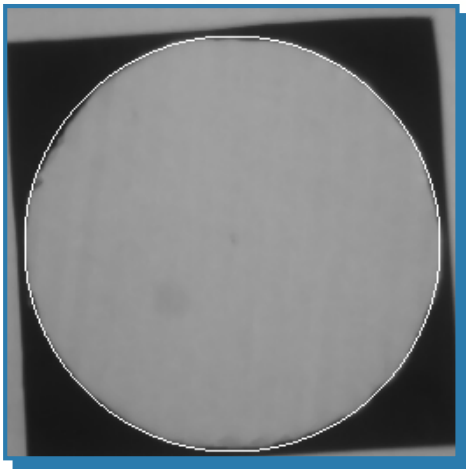
Résultats

Démonstration



Résultats

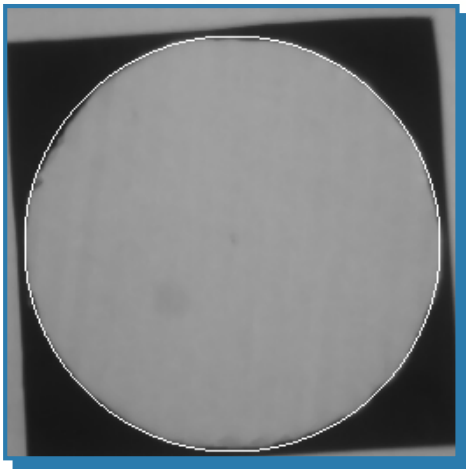
Démonstration



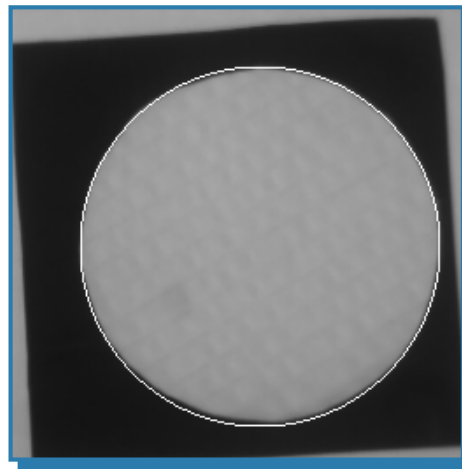
Cercle 1

Résultats

Démonstration



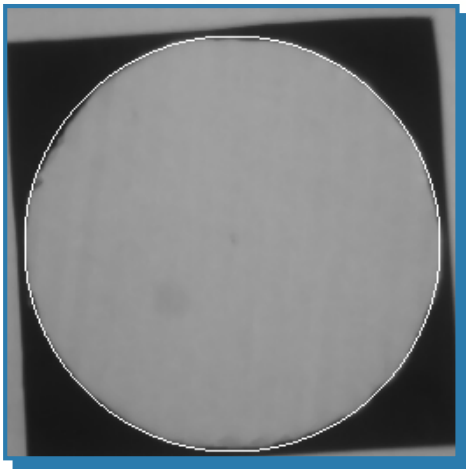
Cercle 1



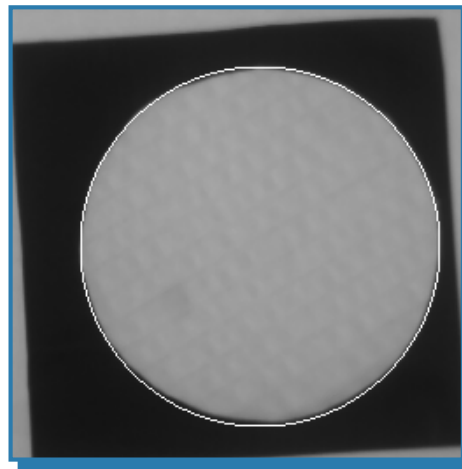
Cercle 2

Résultats

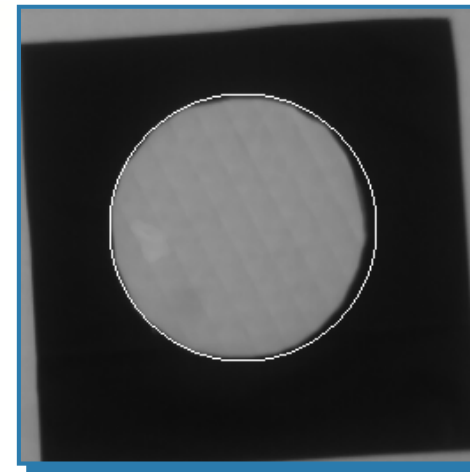
Démonstration



Cercle 1



Cercle 2

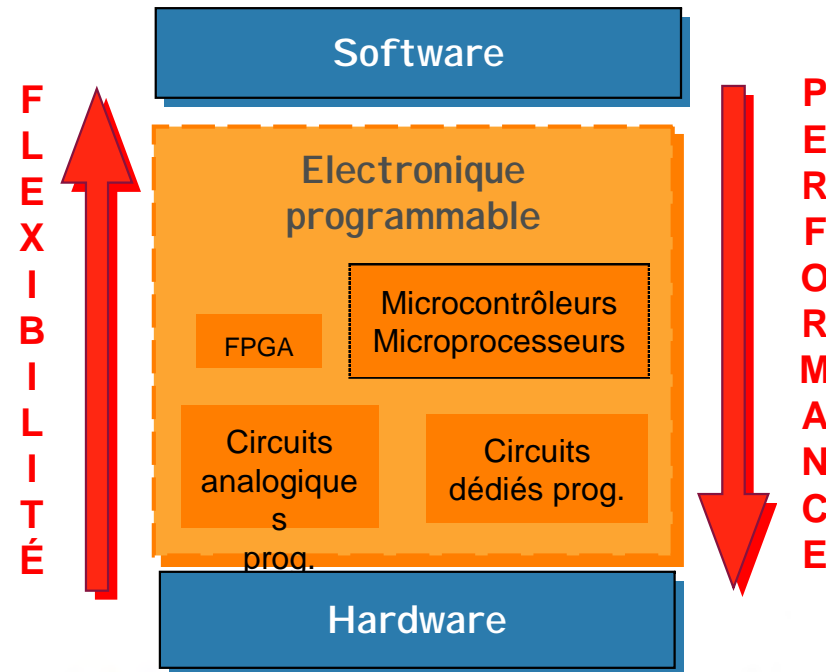


Cercle 3

Notion de temps réel

Utilisation des FPGA

 **FPGA** = *Field Programmable Gate Array*



Notion de temps réel

Utilisation des FPGA

Comparatif des temps d'acquisition d'une image

Séquence de 10 images:

Sans FPGA : *MODE_FREE_RUN*

Avec FPGA : *MODE_FREE_RUN_SLAVE*

Notion de temps réel

Utilisation des FPGA

❖ Comparatif des temps d'acquisition d'une image

Séquence de 10 images:

❖ Sans FPGA : *MODE_FREE_RUN*

Avec FPGA : *MODE_FREE_RUN_SLAVE*

Notion de temps réel

Utilisation des FPGA

❖ Comparatif des temps d'acquisition d'une image

Séquence de 10 images:

❖ Sans FPGA : *MODE_FREE_RUN*

❖ Avec FPGA : *MODE_FREE_RUN_SLAVE*

Notion de temps réel

Utilisation des FPGA

Comparatif des temps d'acquisition d'une image

Séquence de 10 images:

Sans FPGA : *MODE_FREE_RUN*

Avec FPGA : *MODE_FREE_RUN_SLAVE*

image N°	1	2	3	4	5
Sans FPGA (ms)	9.36604	1.58596	3.94106	4.02999	4.07505
Avec FPGA (ms)	5,0787	0.036955	0.036955	0.036955	0.036955
image N°	6	7	8	9	10
Sans FPGA (ms)	4.05097	4.03404	3.65995	4.80198	4.74787
Avec FPGA (ms)	0.038147	0.038147	0.037909	0.036955	0.036001

Conclusion



- Compilation de programmes tests
- Récupération et affichage d'une image
- Traitement d'une image capturée
- Utilisation des bibliothèques OpenCV et Camellia
- Réalisation d'une séquence de traitement d'images
- Optimisation du temps d'acquisition

Conclusion



Nous sommes parti d'un système inconnu
plus value: manuel d'utilisation clair et exhaustif

- Compilation de programmes tests
- Récupération et affichage d'une image
- Traitement d'une image capturée
- Utilisation des bibliothèques OpenCV et Camellia
- Réalisation d'une séquence de traitement d'images
- Optimisation du temps d'acquisition

Conclusion



Nous sommes parti d'un système inconnu
plus value: manuel d'utilisation clair et exhaustif

- Connexion à distance via un terminal / page HTML
- Compilation de programmes tests
- Récupération et affichage d'une image
- Traitement d'une image capturée
- Utilisation des bibliothèques OpenCV et Camellia
- Réalisation d'une séquence de traitement d'images
- Optimisation du temps d'acquisition

Conclusion

Supervision d'automates avec CodeSys



Conclusion

 Supervision d'automates avec CodeSys



Conclusion

- ✦ Supervision d'automates avec CodeSys
- ✦ Possibilité de poursuivre le projet en utilisant les FPGA

Conclusion

- ✦ Supervision d'automates avec CodeSys
- ✦ Possibilité de poursuivre le projet en utilisant les FPGA

