

# Enseigner les Sciences du Numérique avec MATLAB et Simulink



## Découvrez comment aborder les Sciences du Numérique avec MATLAB et Simulink :

- Langage et programmation,
- Représentation binaire et opérations booléennes,
- Acquisition et analyse d'une image numérique,
- Acquisition d'un son,
- Parcours de graphes,
- Architecture des ordinateurs et communication réseau,
- Programmation d'un robot...

# Préambule

## Introduction :

Ce livret est destiné aux professeurs des collèges et lycées enseignant la programmation et les Sciences du Numérique.

Il présente comment aborder les différents thèmes du programme d'Informatique et Sciences du Numérique à l'aide des logiciels MATLAB et Simulink.

Pour chacun de ces thèmes :

- Les objectifs pédagogiques du programme sont rappelés,
- Une App, un modèle ou un programme est proposé pour l'aborder,
- Un jeu de questions est suggéré pour exploiter ces programmes en classe,
- Des éléments de réponse sont donnés en fin de manuel.

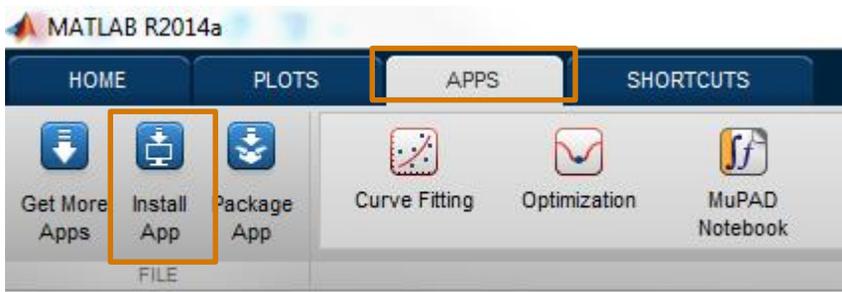
## Ressources :

- Site web académique: <http://fr.mathworks.com/academia/lycees/>
- Site web pour l'ISN : <http://fr.mathworks.com/academia/lycees/sisn.html>
- Ressources pour l'enseignement: <http://fr.mathworks.com/academia/classroom-resources/index.html>
- Cibles matérielles supportées: <http://fr.mathworks.com/academia/hardware/index.html>
- Didacticiels: [http://fr.mathworks.com/academia/student\\_center/tutorials/index.html](http://fr.mathworks.com/academia/student_center/tutorials/index.html)
- Webinars académiques:  
<http://fr.mathworks.com/company/events/webinars/index.html?q=%2520sector:%2522Education%2522>

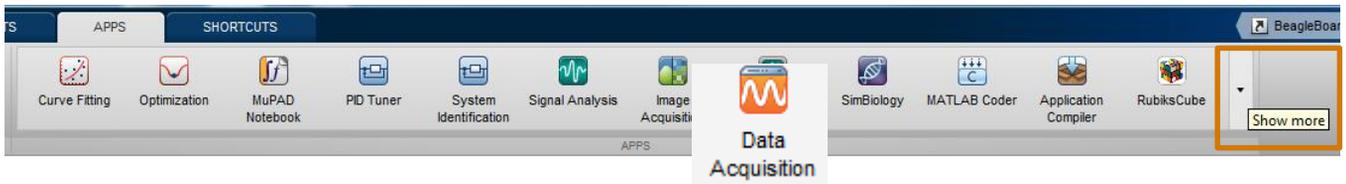
# Démarrage et installation

## MATLAB App :

1. Les programmes, Apps et modèles décrits dans ce livret peuvent être téléchargés à cette adresse : <http://www.mathworks.com/matlabcentral/fileexchange/47854>
2. Pour installer les Apps : cliquez sur Install App dans le bandeau MATLAB. Sélectionnez le fichier monApp.mlappinstall. Puis Install.



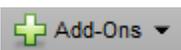
3. Faire de même pour le fichier DataAcquisition.mlappinstall
4. Dans l'onglet Apps, cliquez sur la flèche pour voir toutes les Apps.



5. Double-cliquez sur l'app monApp :

## Support Package pour Raspberry Pi et LEGO MINDSTORMS :

Pour se connecter et programmer les cartes Raspberry Pi et les robots LEGO Mindstorms EV3 et NXT depuis Simulink :

1. Dans le bandeau MATLAB, sélectionnez  puis **Get Hardware Support Packages**.
2. Sélectionnez les supports packages requis et suivre la procédure d'installation.

# Langages et programmation avec MATLAB (1/2)

## Objectifs du programme :

- Choisir un type de données en fonction d'un problème à résoudre
- Concevoir l'en-tête d'une fonction, puis la fonction elle-même

```

Command Window
>> a = 26
b = uint8(10)
c = 'MATLAB is cool'
d = true

a =

    26

b =

    10

c =

MATLAB is cool

d =

     1

>> whos

Name      Size      Bytes  Class  Attributes

a         1x1         8  double
b         1x1         1  uint8
c         1x14        28  char
d         1x1         1  logical
    
```

Name	Value	Size	Bytes	Min	Max	Class
a	26	1x1	8	26	26	double
b	10	1x1	1	10	10	uint8
c	'MATLAB is cool'	1x14	28			char
d	1	1x1	1			logical

### Exemple 1: Les types de données

1. MATLAB détermine automatiquement le type de données de vos variables (double, char, booléen).
2. Tapez **whos** pour visualiser les variables du workspace, leur type et la taille qu'elles occupent en mémoire. Sur combien d'octets est stocké un entier? Un double? Un caractère?
3. La fenêtre **workspace** vous permet de consulter ces mêmes variables.

[Pour en savoir plus sur les types de données dans MATLAB.](#)

### Exemple 2: Concevoir l'en-tête d'une fonction

1. L'en-tête d'une fonction `myfun` prenant `x1, ..., xM` en entrée et renvoyant `y1, ..., yN` en sortie s'écrit :

```
function [y1, ..., yN] = myfun(x1, ..., xM)
```

1. Le code Analyzer aide à écrire correctement l'en-tête de sa fonction (variable d'entrée non utilisée, variable de sortie non affectée).

```

function [T_trie,A] = tri_insertion(T,B)
% Algorithme de tri par insertion
% T Tableau à trier
% T_trie tableau trié

T_trie = T;
for i = 2:length(T)
    x = T_trie(i);%valeur à insérer au tour i
    j = i;
    while (j>1 && T_trie(j-1)>x)
        T_trie(j)=T_trie(j-1);%décalage des + grandes valeurs
        j=j-1;
    end
    T_trie(j)=x;%insérer dans la bonne case
    
```

Line1: function [T\_trie,A] = tri\_insertion(T,B)

⚠ Line 1: The function return value 'A' might be unset. Details Fix

⚠ Line 1: Input argument 'B' might be unused. If this is OK, consider replacing it by ~. Details Fix

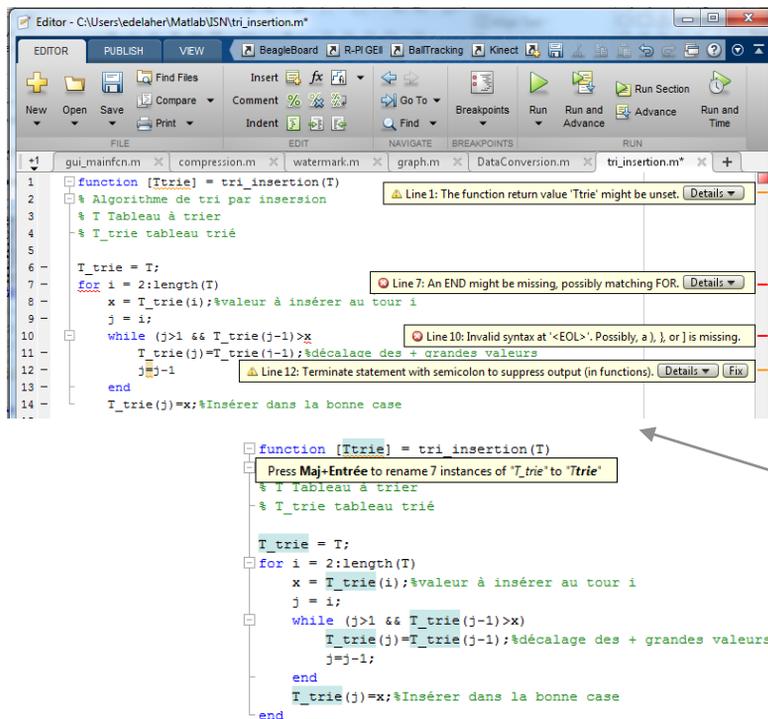
## Pour en savoir plus

- [Vidéo : Introducing MATLAB fundamental data types](#)
- [Vidéo : Programming and developing algorithms with MATLAB](#)

# Langages et programmation avec MATLAB (2/2)

## Objectif du programme :

- Mettre au point un programme en le testant, en l'instrumentant



**Exemple 3:** Correction des erreurs de syntaxe facilitée avec le Code Analyzer.

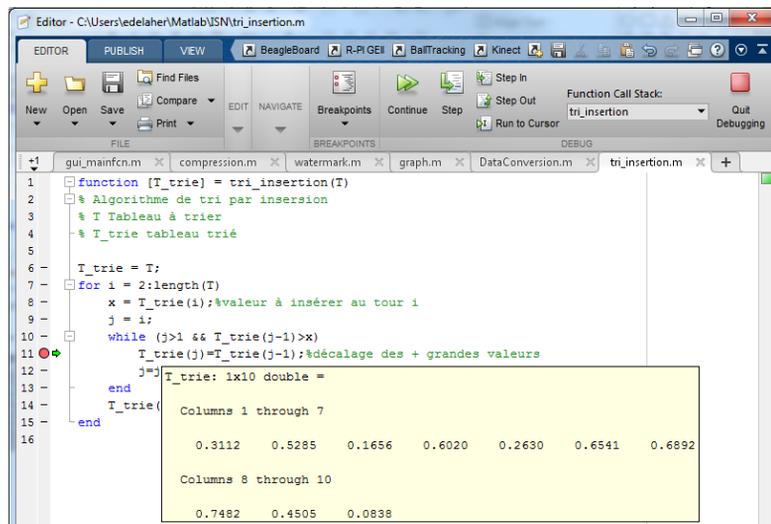
Le Code Analyzer affiche les erreurs et problèmes potentiels de votre code, ainsi que des améliorations possibles. Il facilite la correction des erreurs et donc l'apprentissage de la syntaxe.

- La fonction ci-contre contient quatre erreurs de syntaxe. Corrigez-les en vous aidant des warnings et erreurs du Code Analyzer.
- Renommer la variable T\_trie en Trie dans tout le programme. Modifiez la première occurrence puis appuyez sur Maj+Entrée pour renommer toutes les instances.

## Exemple 4: Débogage des programmes

Le débogueur permet de :

- positionner des points d'arrêt et des points d'arrêt conditionnels (arrêt si une variable atteint une certaine valeur) dans vos programmes.
- Exécuter le programme pas à pas.
- Visualiser les valeurs des variables de manière interactive.

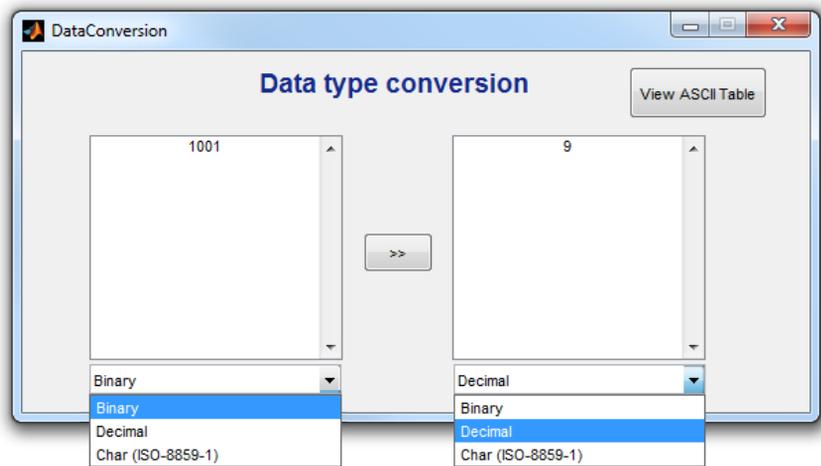


# Représentation binaire et opérations booléennes

## Objectifs du programme :

- Manipuler à l'aide d'opérations élémentaires les trois unités de base : bit, octet, mot
- Exprimer des opérations logiques simples par combinaison d'opérateurs de base
- Coder un nombre, un caractère au travers d'un code standard

**Exemple 1 :** App de conversion entre types binaire, décimal et caractère  
(à installer depuis le dossier \DataTypeConversion)



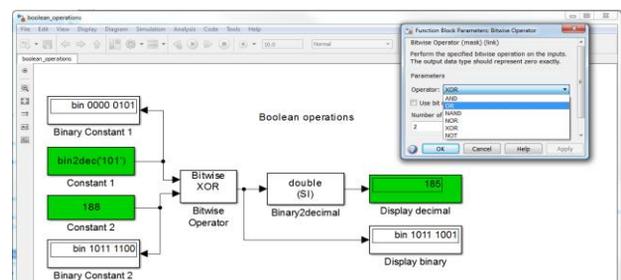
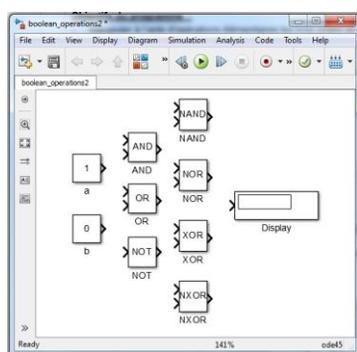
- Convertir '10' de :
    - Binaire à binaire
    - Décimal à binaire
    - Caractère à binaire
    - Hexadécimal à binaire
- Le résultat est-il identique?

- Que se passe-t-il lors de la conversion de '10' du type caractère au type binaire?
- Retrouver le résultat en 2 étapes en passant de :
  - Caractère à décimal en vous aidant de la table ASCII
  - Puis décimal à binaire

**Exemple 2 :** Opération booléennes avec Simulink  
(modèle dans \Boolean)

- Choisir et relier les portes logiques fournies dans le modèle et remplir la table de vérité.

a	b	NON a	NON b	((NON a) ET (NON b))	((NON a) OU (NON b))	(a OU b)
0	0					
0	1					
1	0					
1	1					

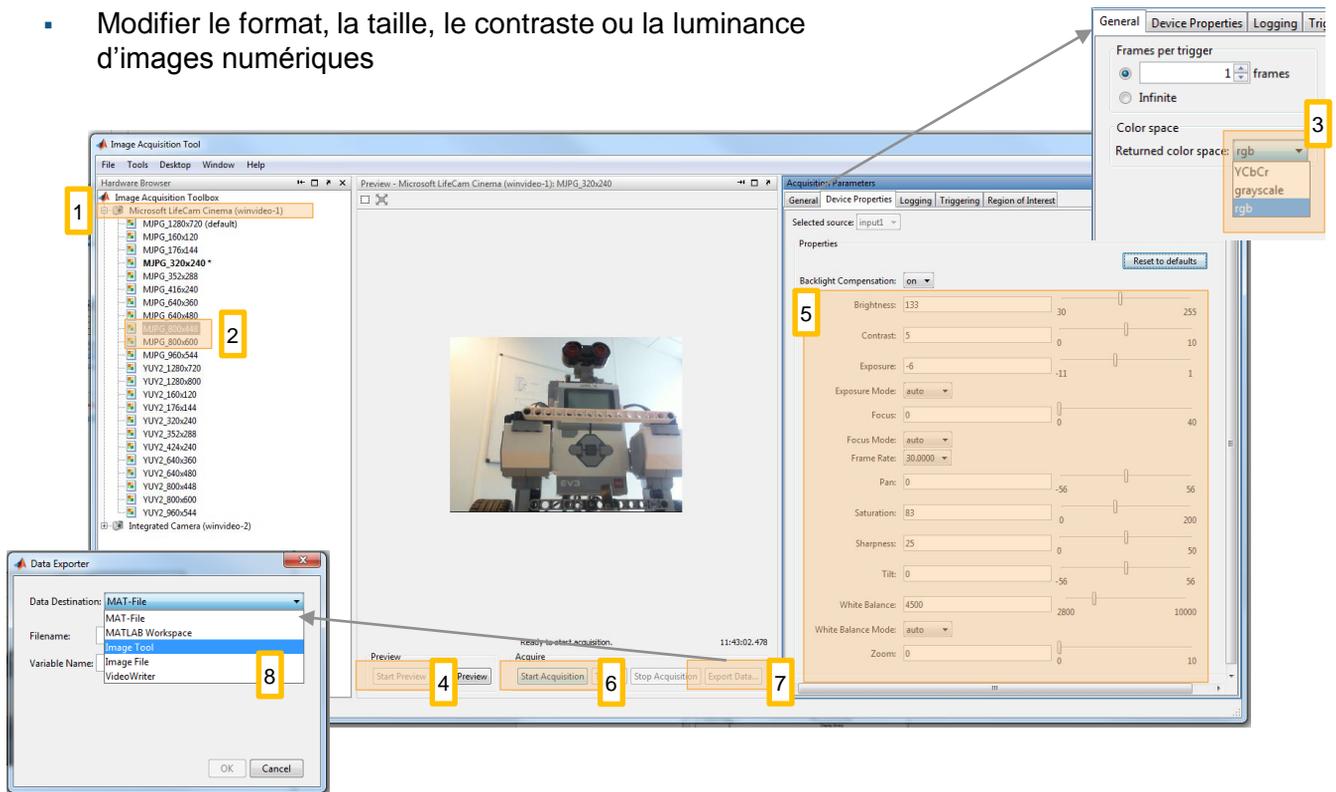


- Réalisez les opérations suivantes:
  - $10001_2$  ET  $100010_2$
  - $90_{10}$  OU  $100010_2$

# Numérisation d'une image

## Objectifs du programme:

- Numériser une image sous forme d'un tableau de valeurs numériques
- Modifier le format, la taille, le contraste ou la luminance d'images numériques



## Exemple : Acquisition d'une image numérique

Tapez imaqttool à l'invite de commande MATLAB ou lancer l'App Image Acquisition

1. Sélectionner la caméra
2. Choisir la résolution
3. Sélectionner l'espace couleur (RGB ou niveau de gris)
4. Lancer la Preview.
5. Ajuster les paramètres de contraste, luminosité, exposition...
6. Pour chaque jeu de paramètres, lancez une acquisition
7. Sauvegarder l'image capturée dans le Workspace de MATLAB ou dans un fichier (.MAT ou format image).
8. Pour analyser l'image, choisir Image Tool.
9. Vous pouvez également définir une région d'intérêt dans l'image (Onglet Region of Interest)
10. Faites l'acquisition de plusieurs frames (une vidéo!). Pour cela, allez dans l'onglet Triggering et modifiez « Number of Triggers », mettre le nombre de frames à enregistrer ou Infini si vous souhaitez arrêter l'enregistrement manuellement.
11. Lancez/Arrêtez l'acquisition.

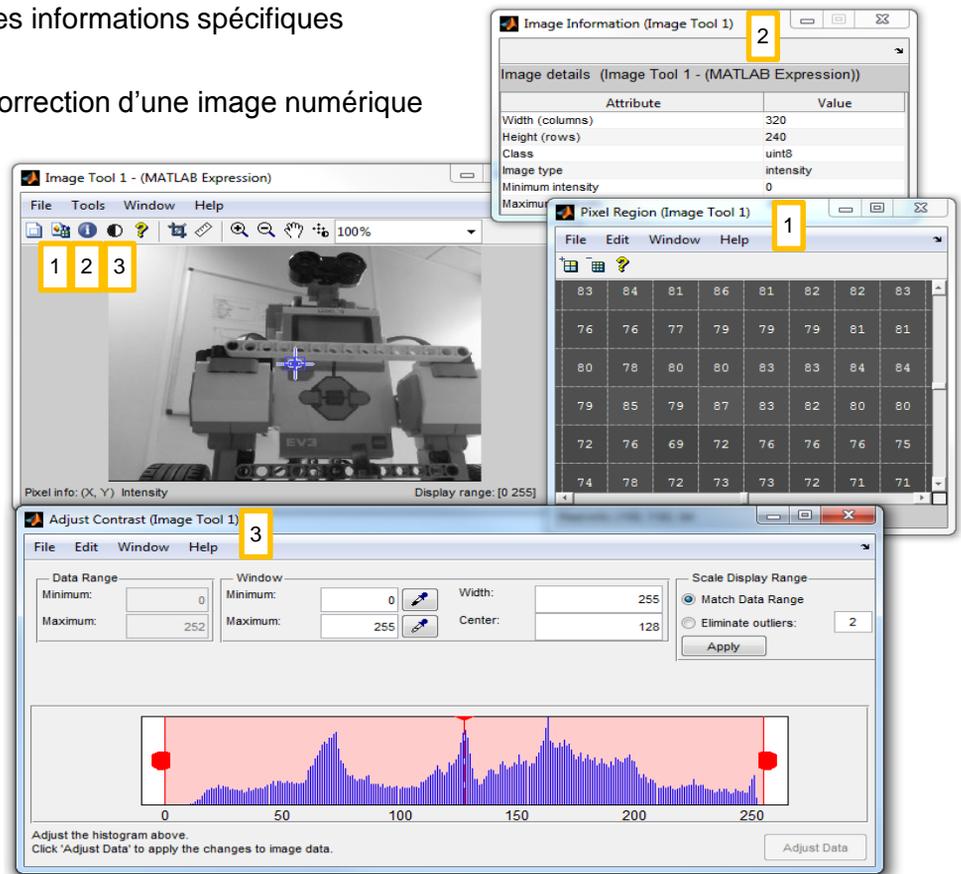


# Analyse d'une image numérique

## Objectifs du programme :

- Numériser une image sous forme d'un tableau de valeurs numériques
- Modifier le format, la taille, le contraste ou la luminance d'images numériques
- Filtrer et détecter des informations spécifiques

## Exemple : Analyse et correction d'une image numérique



The screenshot shows the MATLAB Image Tool interface with three numbered callouts:

- 1**: Points to the main image view where a region of interest is selected.
- 2**: Points to the 'Image Information' window showing details for 'Image Tool 1 - (MATLAB Expression)'. The table below shows the image attributes:

Attribute	Value
Width (columns)	320
Height (rows)	240
Class	uint8
Image type	intensity
Minimum intensity	0
Maximum intensity	255

- 3**: Points to the 'Adjust Contrast' window, which includes a histogram and various adjustment parameters.



Taper `imtool` à l'invite de commande MATLAB ou lancez l'App Image viewer.

- 1. Pixel region** : Regarder les valeurs des pixels dans différentes régions (claire, sombre). Comment évolue la couleur en fonction de la valeur du pixel?
- 2. Image information** : Regarder les caractéristiques de l'image. Quelle est sa taille? La classe de l'image est `uint8`. Qu'est-ce que cela signifie? Sur combien de bits est codé chaque pixel? Qu'elles sont les valeurs minimales et maximales que peuvent prendre les pixels? Qu'elles sont les valeurs minimales et maximales que prennent les pixels sur cette image ?
- 3. Adjust contrast** (pour les images en niveau de gris uniquement) Que représente l'histogramme? Quelle valeur de pixel est la plus souvent représentée (à peu près)? Déplacer le curseur de droite de 255 à 200. Puis cliquez sur `Adjust data`. Que s'est-il passé pour tous les pixels dont la valeur était  $>200$ ?
- 4.** Réglez les paramètres comme ci-contre. Comment est devenue l'image? Comment s'appelle ce type d'image? Combien de couleurs différentes y a-t-il dans l'image?



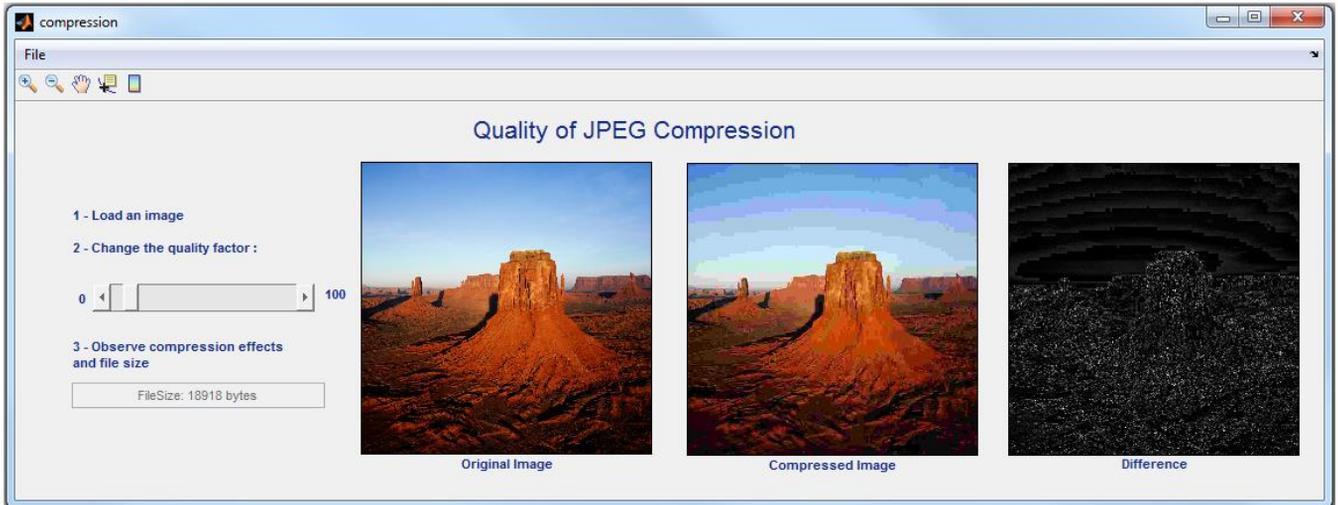
Close-up of the 'Adjust Contrast' window parameters:

Window	
Minimum:	128
Maximum:	129
Width:	1
Center:	129

# Compression

## Objectif du programme :

- Utiliser un logiciel de compression



**Exemple :** Compression JPEG et facteur de qualité (à installer depuis le dossier \JPEGCompression)



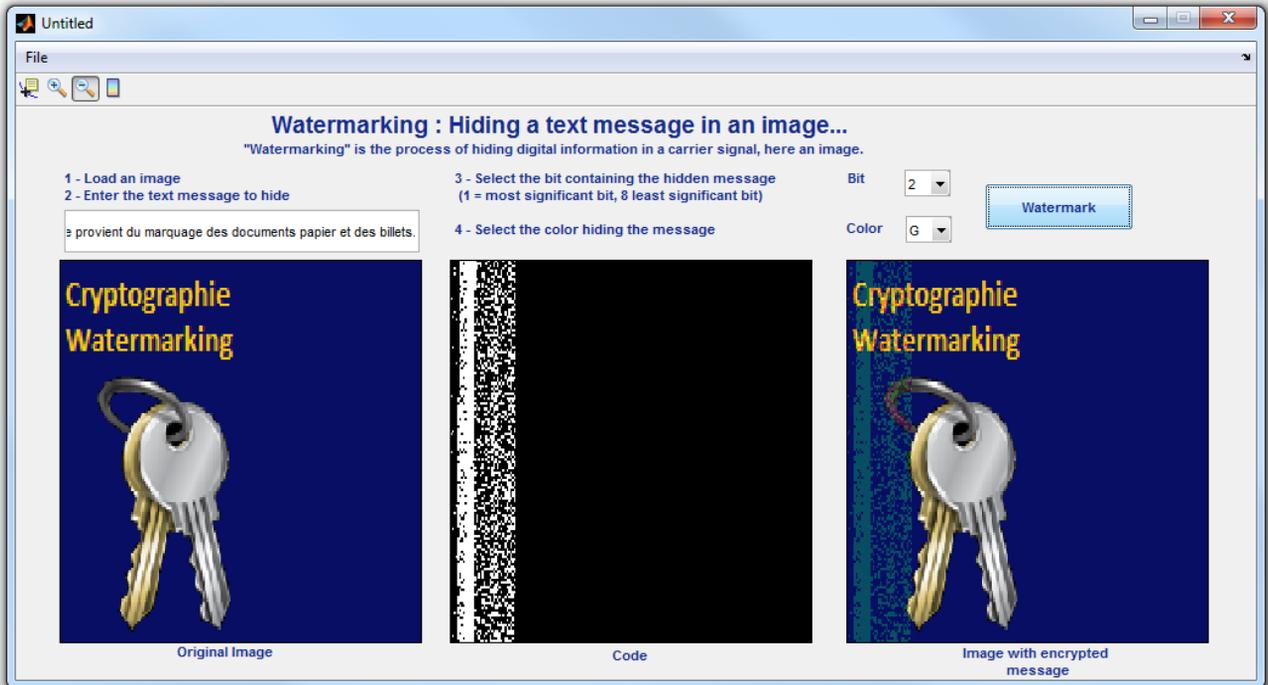
Lancez l'App Compression JPEG.

- Charger une image (File>Open...)
- Mettre le facteur de qualité à 100 et baisser le facteur de qualité progressivement. A partir de quelle valeur vous remarquez une différence entre l'image originale et l'image compressée? Noter cette valeur.
- Continuer à baisser le facteur de qualité. Dans quelles zones de l'image la qualité se dégrade-t-elle plus rapidement?
- Est-ce que la taille du fichier et le facteur de qualité sont proportionnels?
- Charger une nouvelle image.
- Remettre le facteur de qualité à 100 et baisser le facteur de qualité jusqu'à ce que vous remarquiez une différence entre l'image originale et l'image compressée. Est-ce que la valeur est la même que celle notée pour l'image n°1? Pourquoi?

# Application du traitement des images : Le tatouage numérique

## Objectifs du programme :

- Mettre un programme au point
- Comprendre et expliquer ce que fait un algorithme
- Filtrer et détecter des informations spécifiques
- Numériser une image



**Exemple :** Initiation au tatouage numérique (à installer depuis le dossier \Watermarking)

Le **tatouage numérique** (en anglais *digital watermark*) permet d'ajouter des informations de copyright ou d'autres messages de vérification à un fichier audio, vidéo, une image ou un autre document numérique. Le message inclus dans le signal hôte est un ensemble de bits, dont le contenu dépend de l'application. Il peut s'agir du nom du propriétaire ou encore d'une forme de signature. Le nom de cette technique provient du marquage des documents papier et des billets.

Dans le cas présent, le message est converti en binaire puis stocké dans la valeur des pixels de l'image. Le numéro du bit et la composante couleur dans lesquels sont stockés le message sont laissés au choix de l'utilisateur.

Lancer l'App Watermarking

1. Charger une image
2. Entrer un message
3. Sélectionner le numéro de bit et la composante couleur qui stocke l'information.
4. Quel numéro de bit faut-il choisir pour que le message soit peu visible? Pourquoi?

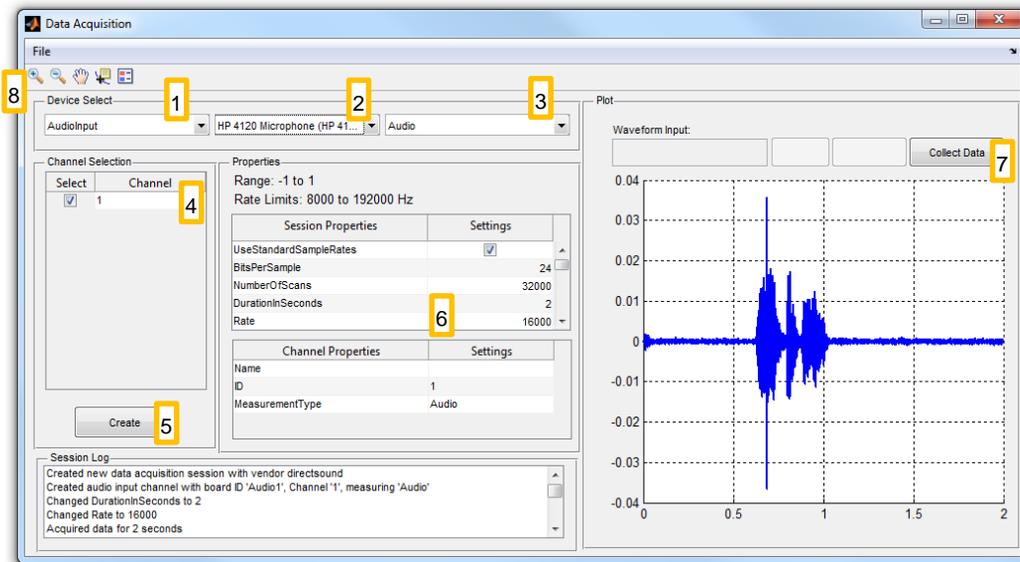


# Acquisition d'un son

## Objectif du programme :

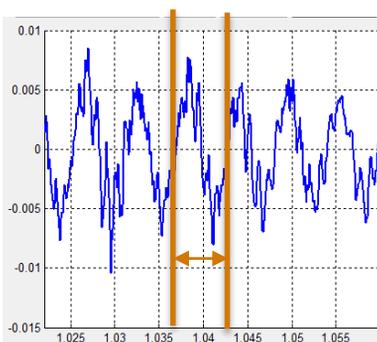
- Numériser un son sous forme d'un tableau de valeurs numériques

**Exemple :** Acquisition, numérisation et écoute d'un son (à installer depuis <http://www.mathworks.com/matlabcentral/fileexchange/44234-data-acquisition>)



## Lancer l'App Data Acquisition

- Sélectionner le type de device : AudioInput
- Sélectionner le microphone
- Sélectionner le type de mesure : audio
- Sélectionner le nombre de canaux
- Créer une session d'acquisition
- Régler les paramètres de l'acquisition, en particulier la durée et la fréquence d'échantillonnage  $F_s$  (ici 1s et 16kHz)
- Lancer l'acquisition en cliquant sur Collect Data
- Une fois l'acquisition terminée, vous pouvez :
  - Zoomer sur le signal. Observer que le signal de parole est un signal pseudo-périodique.

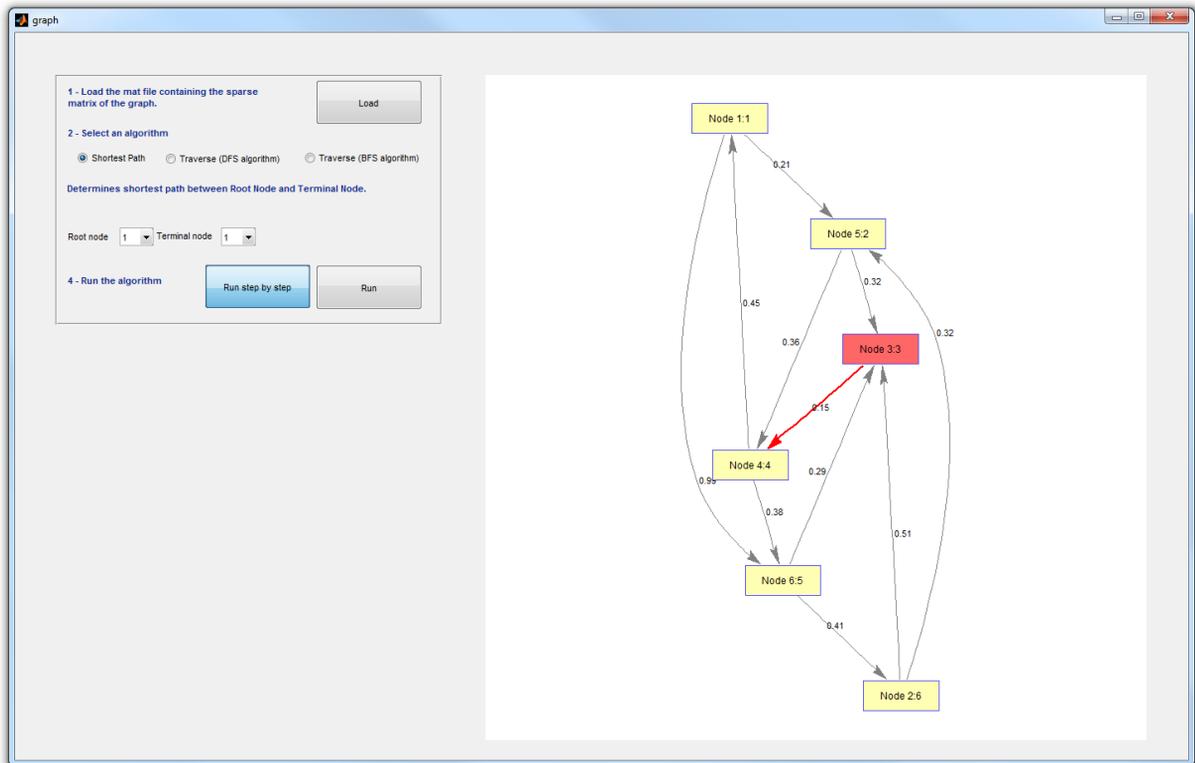


- Utiliser le curseur pour relever les valeurs numériques du signal.
- Quelle est la pseudo-période du signal? En déduire la fréquence fondamentale du signal de parole?
- Sauvegarder votre acquisition : File> Save>monfichier.mat
- Dans MATLAB, tapez les commandes suivantes :
  - `>> load monfichier`
  - Les variables `data` et `time` sont chargées dans le workspace
  - Double-cliquer sur `data` pour voir les valeurs.
- Pour écouter, taper `soundsc(data, Fs)`
- Que se passe-t-il si vous choisissez une fréquence d'échantillonnage plus faible ou plus élevée que pour l'acquisition?

# Algorithmes avancés : graphes

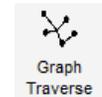
## Objectifs du programme :

- Recherche d'un chemin dans un graphe par un parcours en profondeur (DFS) ou en largeur (BFS)
- Recherche d'un plus court chemin par un parcours en largeur (BFS)



**Exemple :** App de parcours de graphes (à installer depuis le dossier \Graph)

Lancez l'App Graph Traverse



- Créer la matrice sparse représentant votre graphe (doc sparse). Si la valeur de la matrice à la ligne  $i$  et à la colonne  $j$  vaut  $a$  ( $a > 0$ ), cela signifie qu'il y a un arc orienté entre les nœuds  $i$  et  $j$ , de poids  $a$ .

La matrice correspondant au graphe ci-dessus se construit avec la syntaxe suivante :

```
>>W = [.41 .99 .51 .32 .15 .45 .38 .32 .36 .29 .21];
>>G = sparse([6 1 2 2 3 4 4 5 5 6 1], [2 6 3 5 4 1 6 3 4 3 5], W)
```

Arc orienté entre le nœud 6 et le nœud 2 de poids 0.41.

- Sauvegarder la matrice dans le fichier graph.mat :

```
>> save graph.mat G
```

- Dans l'app, charger le fichier du graph.mat.

- Choisissez un algorithme :

- « Plus court chemin » : sélectionner le nœud de départ et le nœud d'arrivée
- « Parcours DFS » ou « Parcours BFS » : sélectionner le nœud de départ

- Lancez l'algorithme en mode « pas à pas » ou « normal ».

Nœud de départ
Nœud d'arrivée
Poids

# Architecture des ordinateurs et réseaux (1/2)

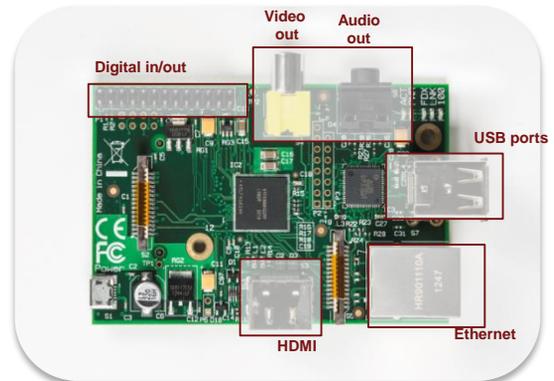
## Objectifs du programme :

- Expliquer le rôle des constituants d'un ordinateur
- Etablir une communication sérielle entre deux machines
- Décrire une situation d'adressage sur un type de réseau particulier

**Exemple :** Communication wifi entre deux cartes Raspberry Pi (modèles dans \RaspberryPiCommunicationWifi)

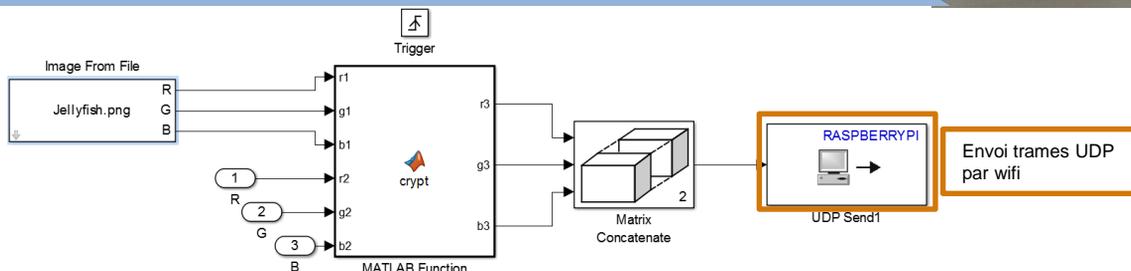
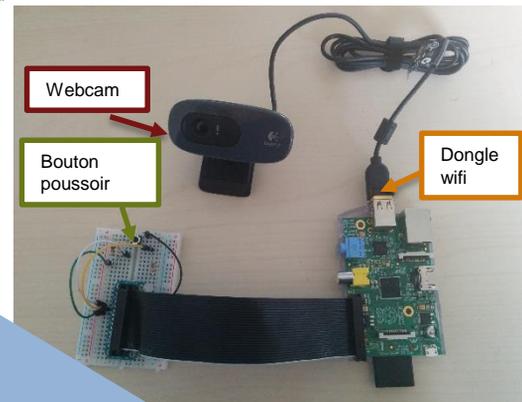
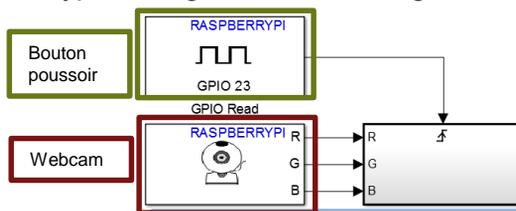
1. La Raspberry Pi est un micro-processeur. Identifier les différents constituants de la carte :

- Mémoire
- Processeur
- Entrées et sorties audio
- Port Ethernet
- GPIO
- Ports USB pour connecter divers périphériques (clavier, souris, webcam...).
- HDMI pour connecter un écran



2. Les modèles ci-dessous permettent d'établir une communication d'image cryptée entre deux cartes Raspberry Pi par wifi :

- L'émetteur (sender\_crypt.slx): Fait l'acquisition d'une image (bouton poussoir), encode l'image et envoie l'image vers le récepteur



3. Emetteur : Double-cliquer sur le bloc « crypt » et analyser l'algorithme de cryptage. Comment fonctionne-t-il? A quoi sert l'image Jellyfish.png? Quel est le risque d'une telle méthode de cryptage?
4. Emetteur : Double-cliquer sur le bloc UDP Send. Quelle adresse IP doit-on donner? Quel port?

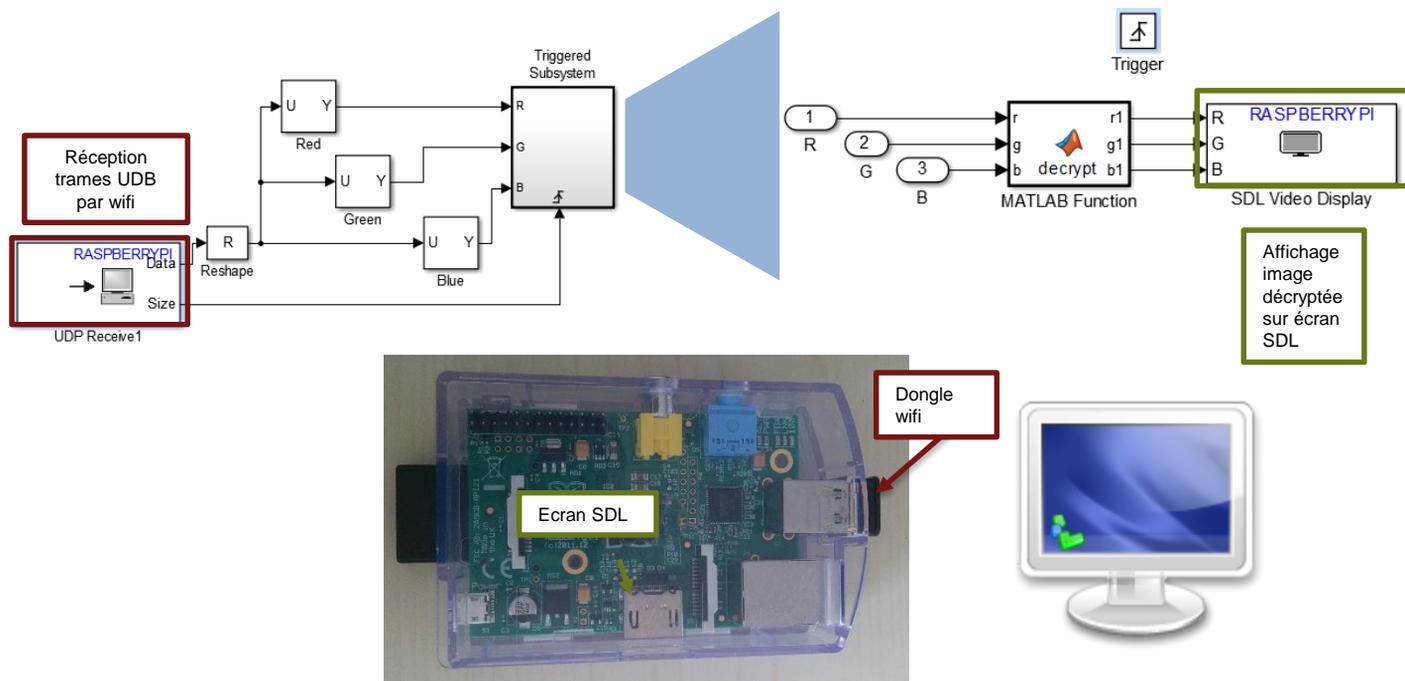
# Architecture des ordinateurs et réseaux (2/2)

## Objectifs du programme :

- Expliquer le rôle des constituants d'un ordinateur
- Etablir une communication sérielle entre deux machines
- Décrire une situation d'adressage sur un type de réseau particulier

## Exemple : Communication wifi entre deux cartes Raspberry Pi

Le récepteur (receiver\_decrypt.slx): Reçoit l'image sous forme de trames, la remet en forme, extrait l'image cryptée et l'affiche.



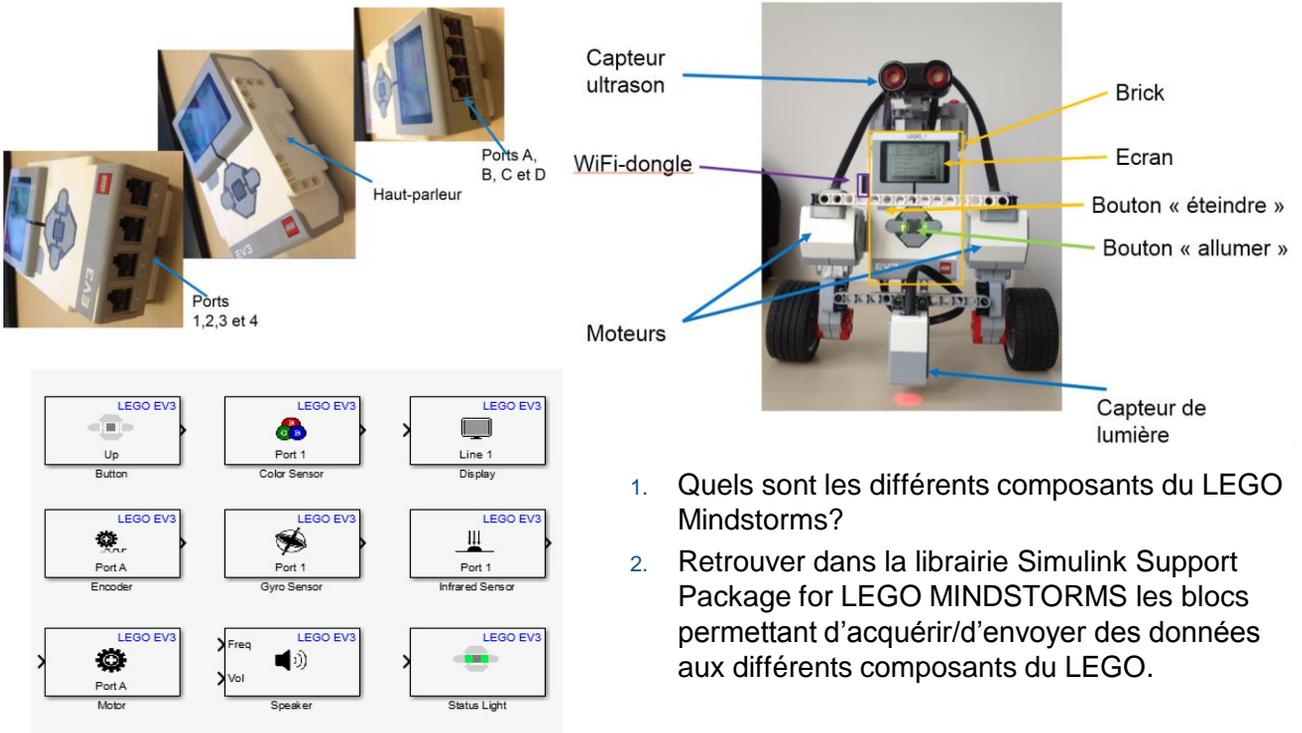
5. Récepteur : Double-cliquer sur le bloc UDP Receive. Quel port doit-on donner? Comment retrouver la taille de la trame reçue (data size) en fonction de la taille de l'image initiale?
6. Récepteur : Double-cliquer sur le bloc « decrypt » et analyser l'algorithme de décryptage. Comment fonctionne-t-il? L'image décryptée correspond-elle exactement à l'image de départ?

# Découverte d'un système numérique et de sa programmation

## Objectifs du programme :

- Identifier les différents composants d'un minirobot et comprendre leurs rôles
- Décrire un système à événements simple à l'aide d'une machine à états finis
- Programmer dans un langage haut niveau un minirobot pour lui faire exécuter une tâche complexe

### Exemple 1 : Identifier les composants du LEGO Mindstorms



1. Quels sont les différents composants du LEGO Mindstorms?
2. Retrouver dans la librairie Simulink Support Package for LEGO MINDSTORMS les blocs permettant d'acquérir/d'envoyer des données aux différents composants du LEGO.

### Exemple 2 : Programmer une suite d'événements sur le LEGO (Modèle dans \LEGO)

Stateflow est une librairie intégrée à Simulink permettant de représenter une logique événementielle à partir de machines d'états.

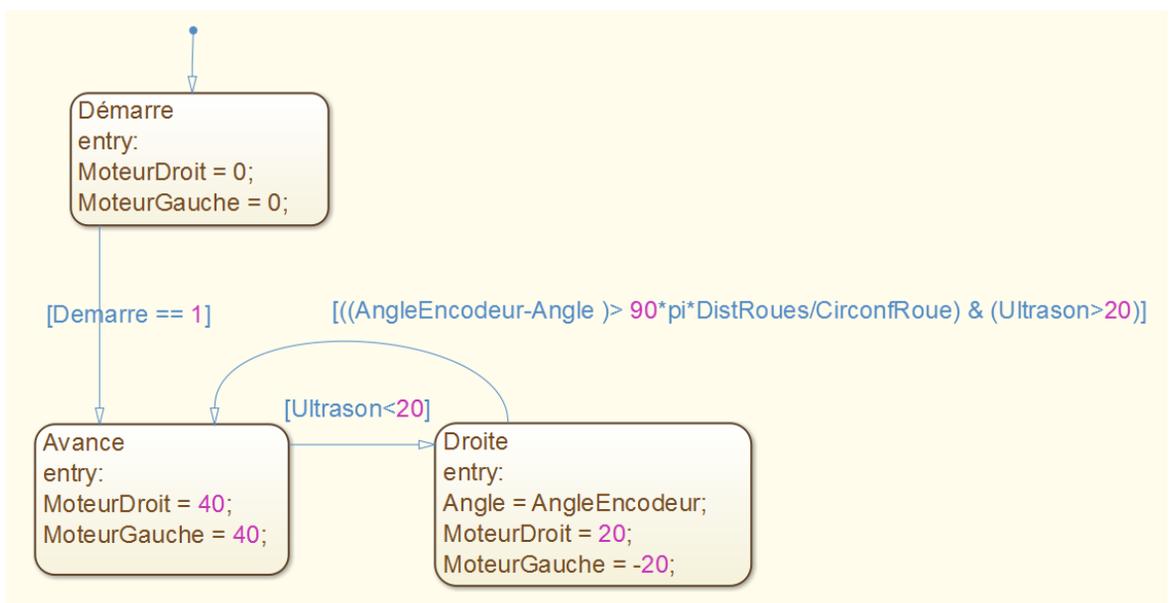
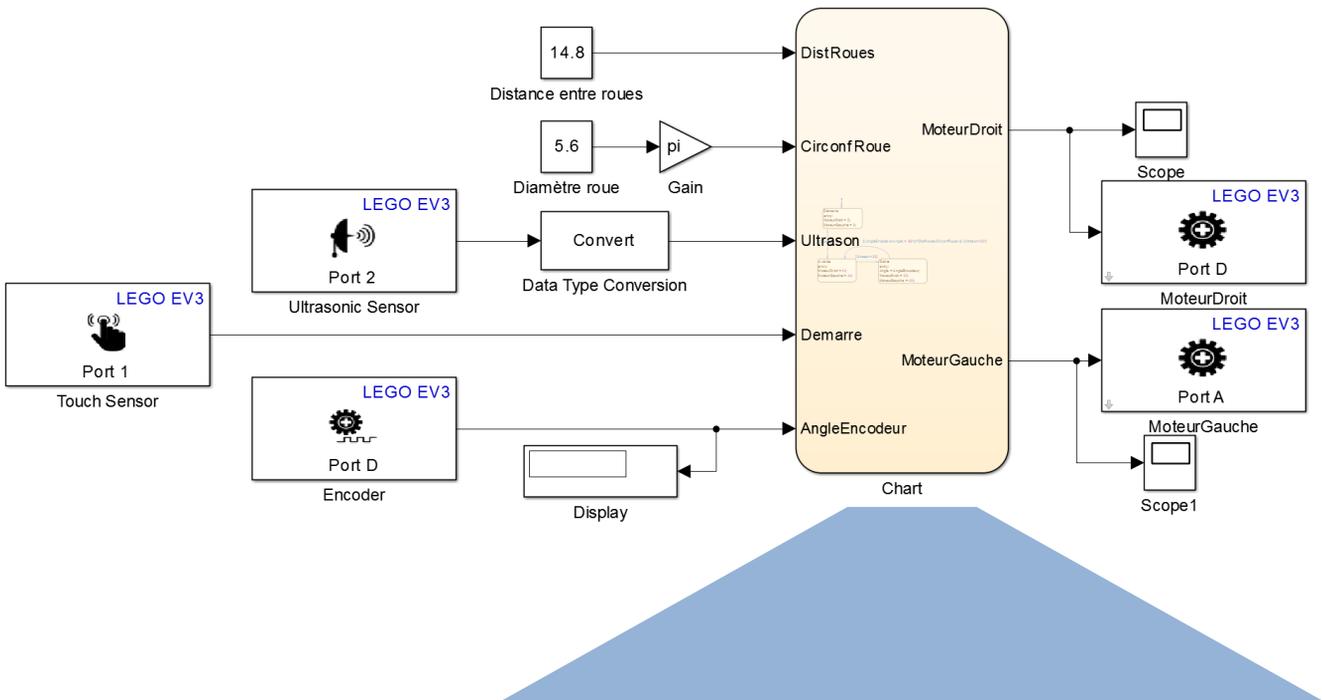
On fait réaliser la suite d'événements suivant au LEGO :

- Démarrage lorsqu'on appuie sur le capteur tactile
  - Avancée en ligne droite
  - Tourne à droite en cas de présence d'un obstacle puis reprend sa trajectoire en ligne droite
1. Ouvrir le modèle lego.slx. Retrouvez les différents états décrits précédemment. A quelle distance de l'obstacle commence-t-on à tourner?
  2. Modifiez les paramètres géométriques (diamètre des roues, distance entre roues) pour qu'ils correspondent à votre robot.
  3. Modifiez le diagramme pour que le robot :
    - tourne à gauche lorsqu'il rencontre un obstacle
    - s'arrête lorsqu'on appuie de nouveau sur son capteur tactile.
    - s'arrête lorsqu'il passe sur une zone rouge

# Découverte d'un système numérique et de sa programmation

## Objectifs du programme :

- Identifier les différents composants d'un minirobot et comprendre leurs rôles
- Décrire un système à événements simple à l'aide d'une machine à états finis
- Programmer dans un langage haut niveau un minirobot pour lui faire exécuter une tâche complexe



# Corrigé

## Langages et programmation avec MATLAB:

### Exemple 1

2. Un entier est stocké sur 1 octet, un double sur 8 octets, un caractère sur 2 octets.

### Exemple 3

1. Programme corrigé :

```
function [T_trie] = tri_insertion(T)
% Algorithme de tri par insertion
% T Tableau à trier
% T_trie tableau trié

T_trie = T;
for i = 2:length(T)
    x = T_trie(i);%valeur à insérer au tour i
    j = i;
    while (j>1 && T_trie(j-1)>x)
        T_trie(j)=T_trie(j-1);%décalage des + grandes valeurs
        j=j-1;
    end
    T_trie(j)=x;%Insérer dans la bonne case
end
```

## Représentation binaire et opérations booléennes:

### Exemple 1

- 10 en décimal est codé 1010 en binaire et 10 en caractère est codé 01100010110000 en binaire.
- Chaque caractère de '10' (1 et 0) est converti. Le résultat est la concaténation des conversions de 1 et 0 de caractère à binaire.
- '1' (char) -> 49 (ASCII) -> 0110001 (binaire)  
'0' (char) -> 48 (ASCII) -> 0110000 (binaire)

Par concaténation '10' (char) -> 01100010110000 (binaire)

### Exemple 2

a	b	NON a	NON b	(NON a) ET (NON b)	((NON a) OU (NON b))	(a OU b)
0	0	1	1	1	1	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	0	0	1

- $(10001_2 \text{ ET } 100010_2) = (0_2)$
- $90_{10} \text{ OU } 100010_2 = (0111 \ 1010_2)$

# Corrigé

## Analyse d'une image couleur

1. Plus la région est sombre, plus la valeur du pixel est faible.
2. Sur cet exemple, l'image est de taille 240\*320. Chaque pixel de l'image est représenté par un entier non signé sur 8 bits. Les valeurs des pixels vont de 0 à 255 ( $2^8 - 1$ ). Pour les valeurs min et max sur l'image, regarder les données Minimum Intensity et Maximum Intensity.
3. L'histogramme représente le nombre de pixels prenant chacune des valeurs possibles de 0 à 255. Dans l'exemple représenté, la valeur la plus représentée se situe autour de 160. Tous les pixels dont la valeur était supérieure à 200 sont mis à 255.
4. On a une image binaire (2 couleurs). Tous les pixels dont la valeur est inférieure à 128 sont devenus noirs et ceux dont la valeur était supérieure sont devenus blancs.

## Compression

3. La qualité de l'image se dégrade rapidement dans les zones de fortes variations spatiales des valeurs des pixels. En effet, la compression JPEG utilise le fait que l'œil humain est peu sensible aux changements rapides d'intensité des pixels. Ainsi on représente en priorité les zones de variation lente des pixels (dégradé par exemple) plutôt que les zones de variation rapide.
4. Non.
6. Pas nécessairement. Cela dépend du type d'image. On notera une différence avec un facteur de qualité beaucoup plus élevé sur l'image binaire d'un carré par exemple. Alors qu'avec une photo, on pourra opter pour une compression plus importante sans noter de différences visuelles.

## Application du traitement des images : le tatouage numérique

4. Le message sera peu visible si l'on utilise le bit de poids faible car il est porteur de peu d'information.

## Acquisition d'un son:

12. Si la fréquence d'écoute est plus élevée que la fréquence d'échantillonnage, le son sera joué plus vite qu'il n'a été enregistré. Dans le cas contraire, il sera joué au ralenti.

## Architecture des ordinateurs et réseaux

3. L'algorithme de cryptage remplace les 4 bits de poids faibles de l'image Jellyfish.png par les 4 bits de poids fort de l'image dont on a fait l'acquisition. Dans une telle méthode de cryptage, on perd de l'information.
4. On renseigne l'adresse IP et le port du récepteur.
5. On donne le même port que pour l'émetteur. La trame reçue correspond à la taille de l'image (ici  $120*160*3$ ).
6. L'algorithme de décryptage isole les 4 bits de poids faible de l'image reçue et les décale vers la gauche pour qu'ils deviennent les 4 bits de poids fort. Les 4 bits de poids faible sont mis à 0. Les 4 bits de poids faible de l'image de départ sont perdus. L'image reçue n'est donc pas exactement identique à celle de départ.

## Découverte d'un système numérique et de sa programmation

1. On tourne lorsque l'on est à moins de 20cm de l'obstacle.