



# Robotique pratique

Fabrice LE BARS



# Présentation générale de la matière

# Présentation générale de la matière

## ■ Objectifs

- Savoir comment construire rapidement des prototypes de robots mobiles simples : choix de capteurs, actionneurs, ordinateurs embarqués adéquats, CAO, etc.
- Savoir préparer des ordinateurs embarqués et stations de contrôle ainsi que les périphériques réseaux nécessaires
- Savoir utiliser des exemples de capteurs et actionneurs courants en robotique mobile
- Savoir programmer les robots pour qu'ils effectuent des missions intéressantes (suivi de trajectoire, d'objets, cartographie, etc.)



# Présentation générale de la matière

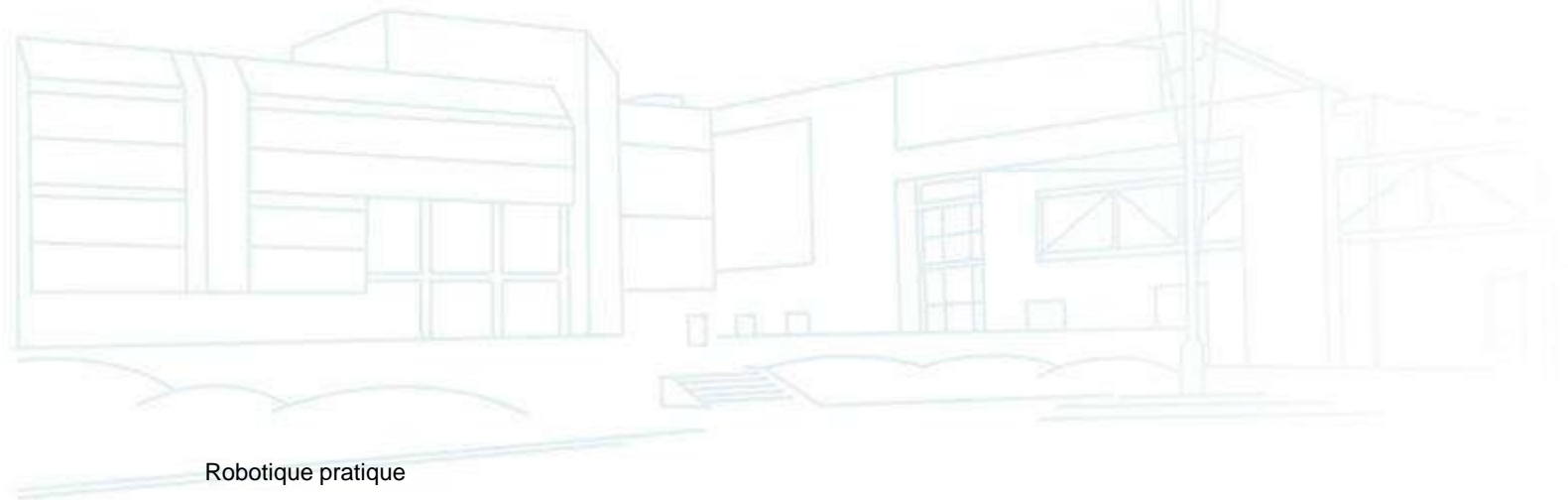
## ■ Planning

- 24H, en général le vendredi matin
- Pour les séances du début, environ 30 min de présentation suivi d'un TP, projet sur les séances de fin
- Amener ses PC portables et écouteurs
- Travaux sur ordinateurs de l'école, ordinateurs embarqués, manipulations de matériel, etc.
- Construction et tests de vrais robots
- Pour gérer les différences de chacun et limiter les achats de matériel, il y aura des sujets à se répartir en équipes
- Evaluation : rendre tous les travaux faits sur Moodle à la fin de chaque séance (codes, données, photos des tests, etc.)

# Présentation générale de la matière

## ■ Planning

- Concevoir un robot / Programmer les capteurs et actionneurs d'un robot (3H)
- Initiation CAO en mécanique avec CATIA (3H)
- Communiquer, contrôler à distance un système informatique autonome (3H)
- Algorithmes pour l'autonomie (3H)
- Projet (12H)



# Concevoir un robot

# Concevoir un robot

- A définir selon l'objectif du robot :
  - Structure mécanique
  - Actionneurs
  - Capteurs
  - Ordinateur embarqué
  - Communication
  - Energie
  
- Dans cette présentation, on considèrera principalement des **robots mobiles entre 50 cm et 4 m** ne pouvant pas embarquer de pilote humain (Unmanned Vehicles), souvent **inspirés du modélisme**

## ■ Vocabulaire :

- **Manned** : humain qui contrôle embarqué sur le robot, autrement on dit **unmanned**

UAV : Unmanned Aerial Vehicle, UGV : Unmanned Ground Vehicle,  
USV : Unmanned Surface Vehicle, UUV : Unmanned Underwater  
Vehicle

- **Autonome** : plusieurs niveaux d'autonomie possibles (e.g. avec surveillance continue par un opérateur vs sans surveillance), sens qui dépend du contexte, désigne en général un **robot se déplaçant sans contrôle continu par un opérateur**

AGV, ASV, AUV



- Plateforme du commerce vs conception en interne :
  - Beaucoup de plateformes toutes faites ou d'ensembles de pièces existent
  - Si le but du robot est de tester des algorithmes sur un prototype, il peut être bon de partir d'éléments du commerce pour minimiser le temps de construction
  - Si le robot risque de subir des modifications notables au cours du développement, il faut pouvoir avoir un contrôle total sur ses éléments
  - Attention à la fiabilité dans le temps des fournisseurs, certaines pièces peuvent vite disparaître des catalogues
  - Les fournisseurs de modélisme peuvent être intéressants pour certaines catégories de robots

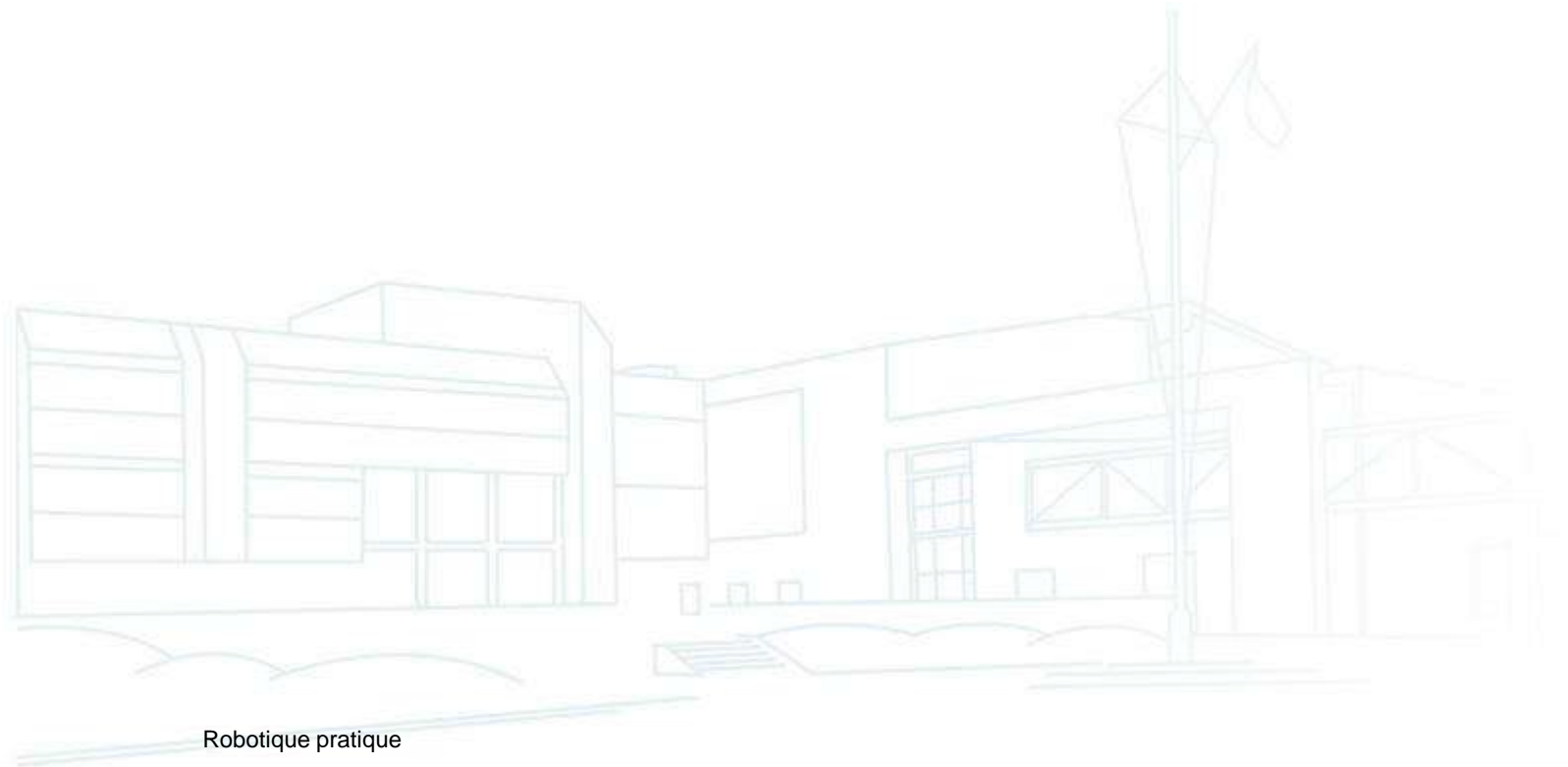
## ■ Structure mécanique :

- Peut être assez différent des systèmes existants conduits par humains
    - e.g. sur la forme ou la façon de se déplacer (décollage vertical, vol à l'envers, fortes accélérations, pas besoin d'ouvertures, etc.)
  - Souvent besoin de méthodes de prototypage rapide
    - e.g. imprimante 3D, découpe fil à chaud (pour polystyrène/mousse), voir aussi cours de CAO avec le logiciel CATIA :
- <https://moodle.ensta-bretagne.fr/mod/resource/view.php?id=32600>

Attention : on imprime pas n'importe quoi avec une imprimante 3D!



- Actionneurs :
  - Peuvent être les mêmes que certains systèmes conduits par les humains, mais très souvent électriques pour les plus petits robots



## ■ Actionneurs :

- Moteurs électriques : brushed vs brushless, pas-à-pas ou non?

### Moteurs brushed DC :

**2 fils (+ et -)**, on peut les reconnaître à cela)

DC = courant continu, pratique si on veut les tester directement une sur batterie

Un électroaimant tourne dans une cage fixe avec des aimants permanents, son alimentation s'inversant avec la rotation de l'axe

L'alimentation est sur l'axe, au **contact de 2 balais** (+ et -)

S'il prend l'eau, le contact risque de ne plus se faire...



## ■ Actionneurs :

- Moteurs électriques : brushed vs brushless, pas-à-pas ou non?

### **Moteurs brushless :**

**3 fils** (on peut les reconnaître à cela)

On ne peut pas facilement les tester directement sur une batterie, il faut une carte de puissance/variateur/ESC pour générer les bons signaux

**Pas de contact électrique entre la partie tournante et la partie fixe**

Souvent la cage est tournante et l'axe est fixe (**outrunner**, l'inverse est **inrunner**)



## ■ Actionneurs :

- Moteurs électriques : brushed vs brushless, pas-à-pas ou non?

Moteurs brushless :

On peut donc le mettre dans un fluide sans trop de problèmes (attention cependant à corrosion des pièces mécaniques, pertes de courant si fluide conducteur et électrolyse...)

Étanchéification dans l'eau de mer :

Dans caisson étanche rempli d'huile si haute pression

Directement baignant dans l'eau si pièces résistantes à la corrosion et bobines de l'électroaimant avec vernis ou résine



## ■ Actionneurs :

- Moteurs électriques : brushed vs brushless, pas-à-pas ou non?

Moteurs brushless :

Prix souvent plus élevé que brushed mais meilleur en conso, couple, vitesse, longévité à taille, poids équivalent, notamment grâce à l'absence des balais (moins de frottements, d'usure, etc.)



## ■ Actionneurs :

- Moteurs électriques : brushed vs brushless, pas-à-pas ou non?

Moteurs brushless :

Souvent utilisés en modélisme aérien

Paramètres importants (ont aussi un sens pour les autres types de moteurs) :

KV : nombre de tours par Volts ou RPM/V

Souvent, les hélices adaptées sont indiquées sur la doc du moteur, en fonction de la tension

Si l'hélice et/ou la tension ne sont pas adaptées, on risque de ne pas obtenir la poussée max, tout en consommant plus, et en faisant chauffer le moteur...



## ■ Actionneurs :

- Moteurs électriques : brushed vs brushless, pas-à-pas ou non?

Moteurs brushless :

Tendances empiriques observées pour le choix d'hélice :

Plus le pas est grand, plus l'avion ira vite

Plus le diamètre est grand, plus l'avion peut être gros

Plus l'hélice a un grand pas et grand diamètre, plus le moteur doit être gros et le Kv diminue

Pour les multicopters, l'énergie nécessaire pour porter 1 gramme est entre 100 et 200 mW

Voir <http://www.ecalc.ch/> pour bien choisir ses moteurs et hélices selon les caractéristiques du robot (aérien)



## ■ Actionneurs :

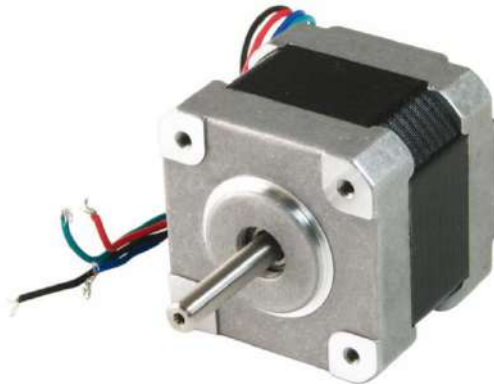
- Moteurs électriques : brushed vs brushless, pas-à-pas ou non?

Moteurs pas-à-pas :

>3 fils

Nécessite une carte de contrôle adaptée

Très précis pour se mettre à un angle donné (e.g. utilisé dans les imprimantes)



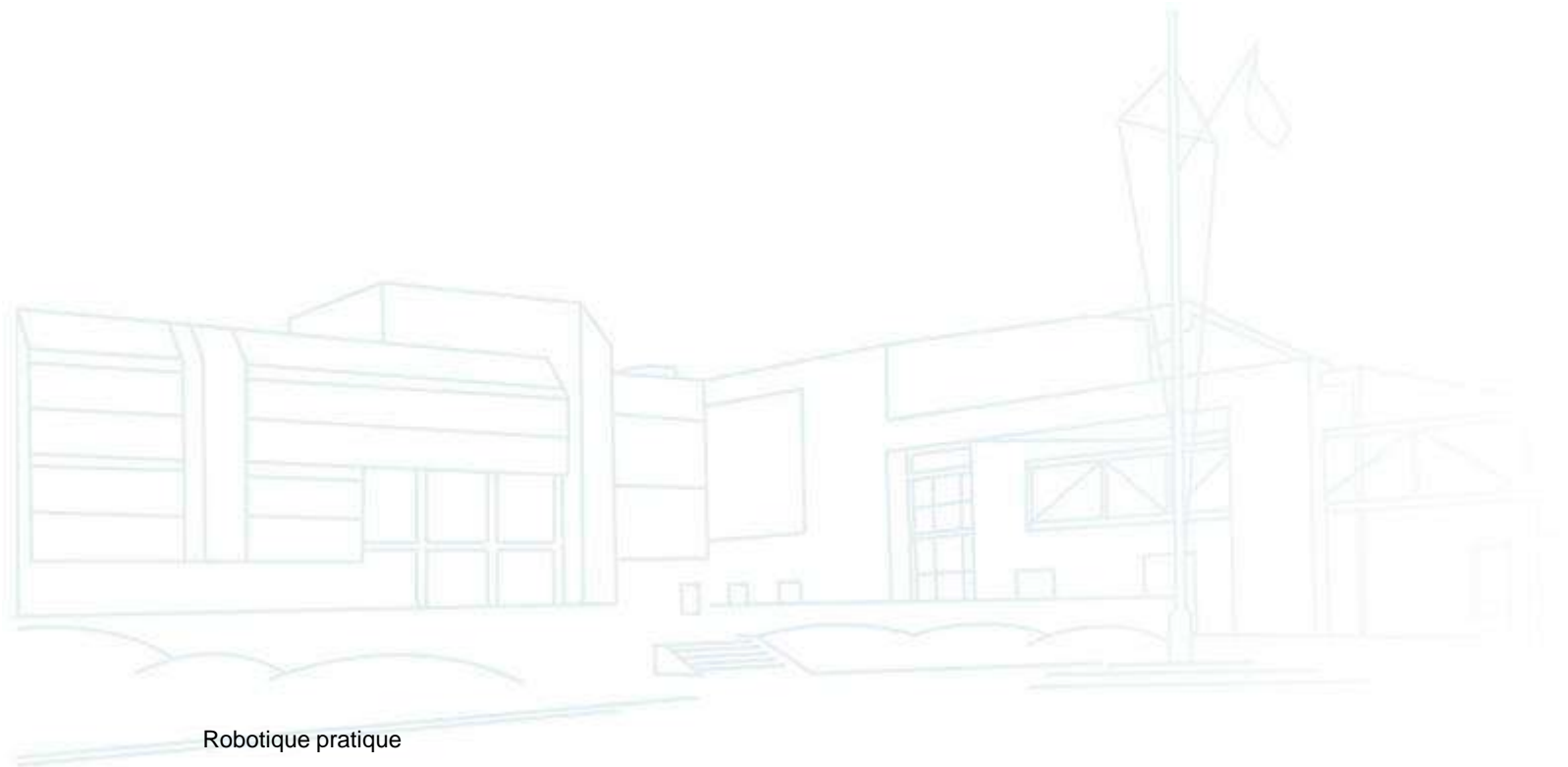
# Concevoir un robot

## ■ Actionneurs :

- Moteurs électriques

Voir cours des FIPA 1 :

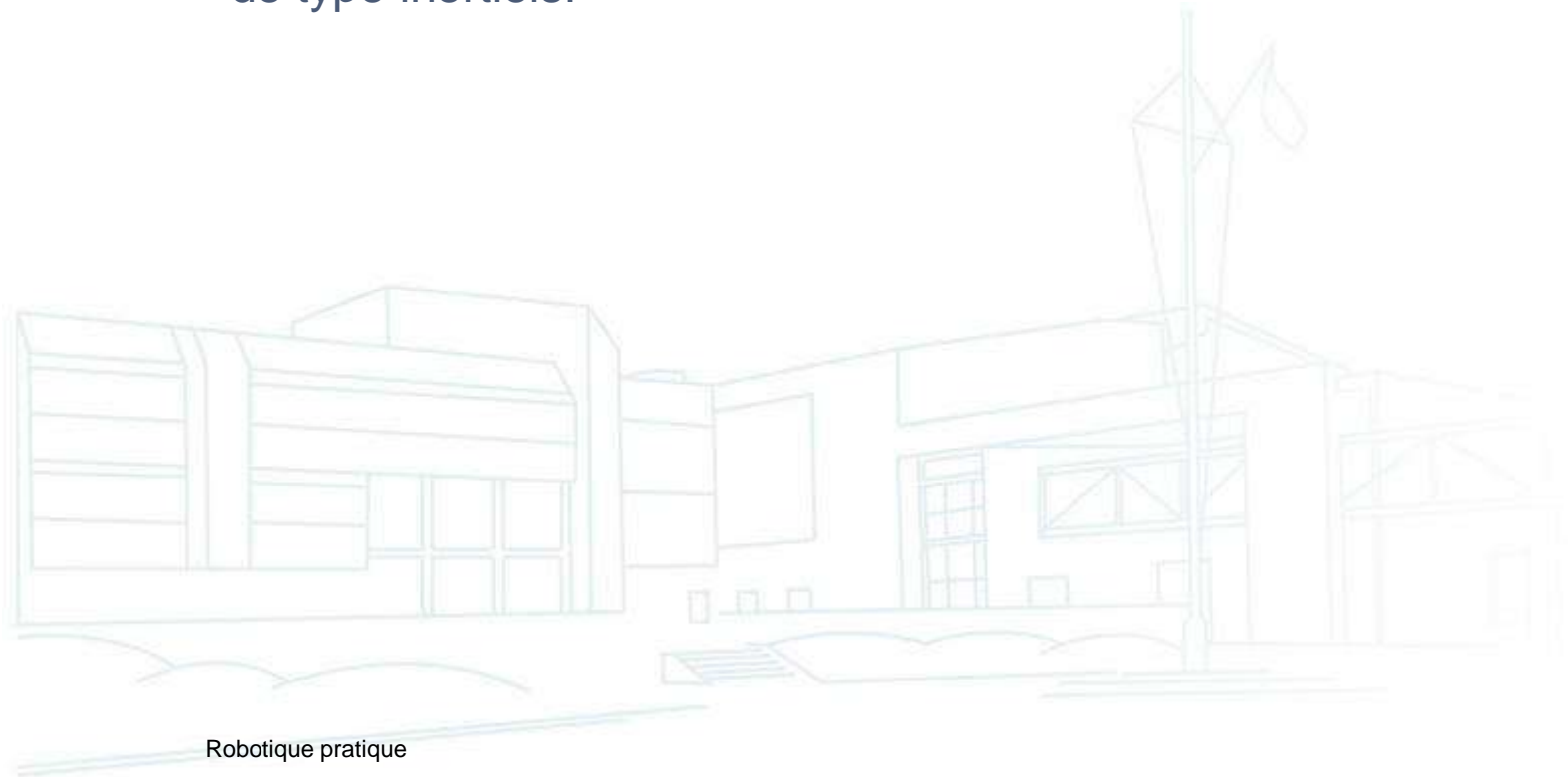
<https://moodle.ensta-bretagne.fr/course/view.php?id=1162>



## ■ Actionneurs :

- Moteurs électriques

Attention aux vibrations e.g. à cause de moteurs et hélices/roues mal fixées/équilibrées, ceci peut fortement perturber certains capteurs de type inertiels!



## ■ Actionneurs :

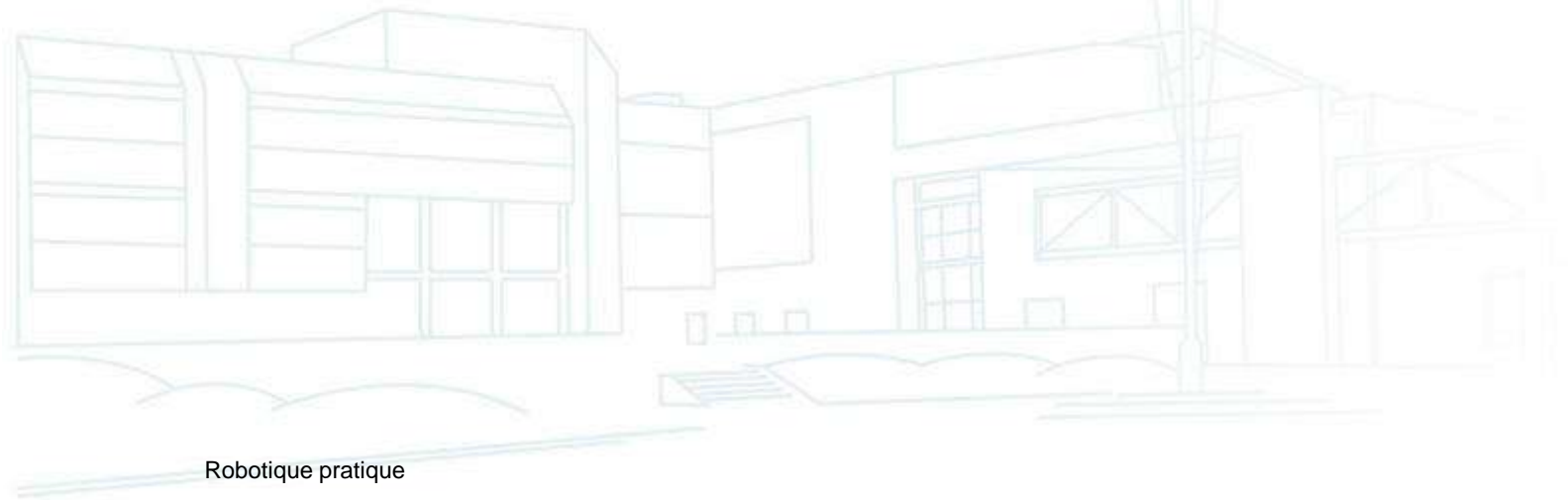
- Carte de puissance/variateur/ESC

Permet de contrôler les moteurs par des signaux de commande

Moteurs : tensions et courants élevés provenant des batteries

Signaux de commande : tensions et courants faibles venant directement ou indirectement de l'ordinateur embarqué

Exemples : signaux PWM (le plus courant en modélisme), I2C, etc.

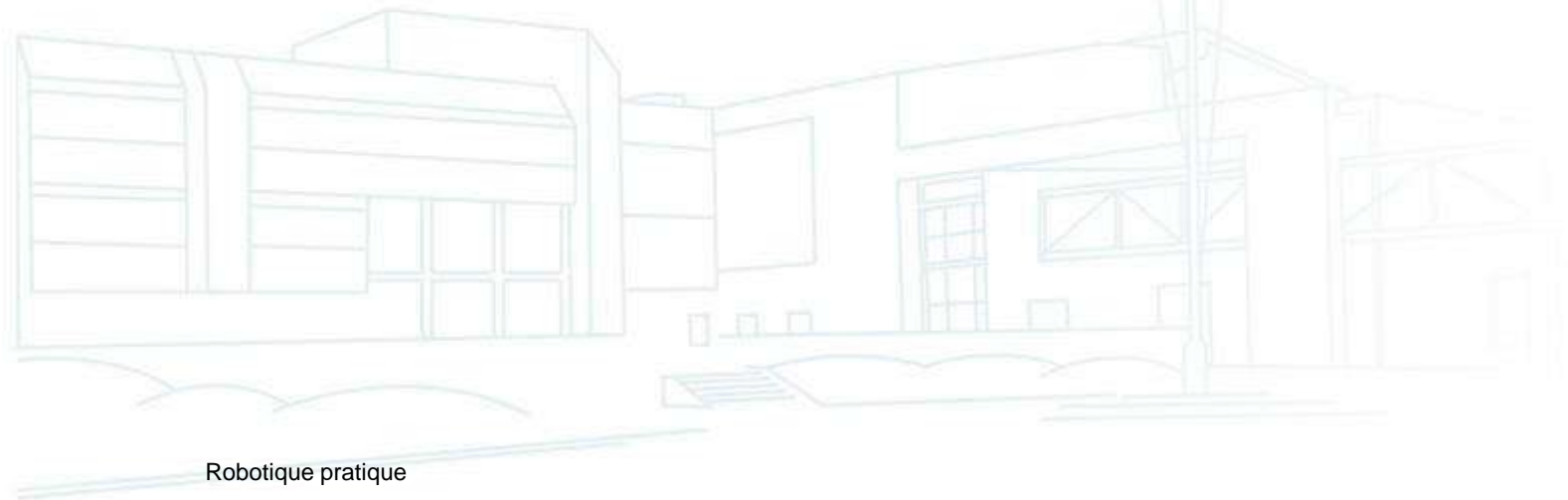


## ■ Actionneurs :

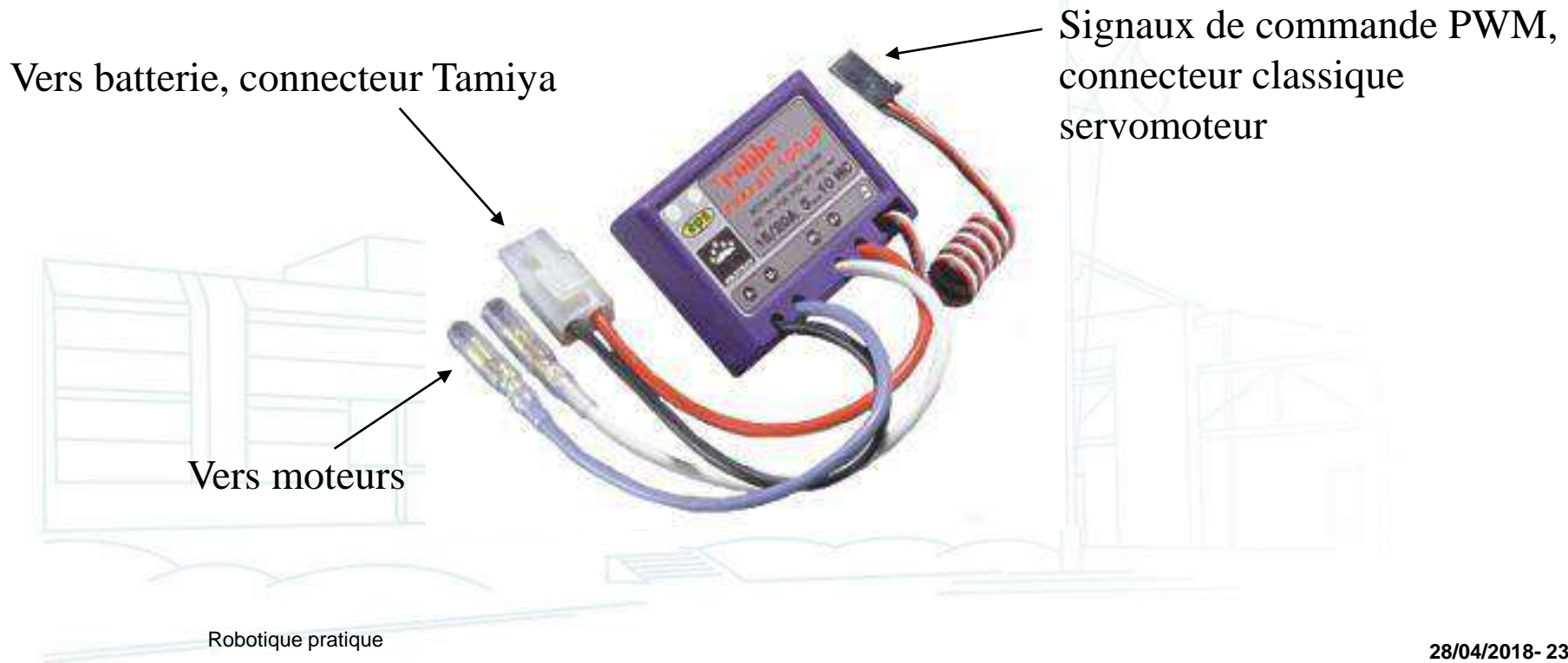
- Carte de puissance/variateur/ESC

ESC pour moteurs brushed : en déclin dans le monde du modélisme avec la baisse des prix des moteurs brushless...

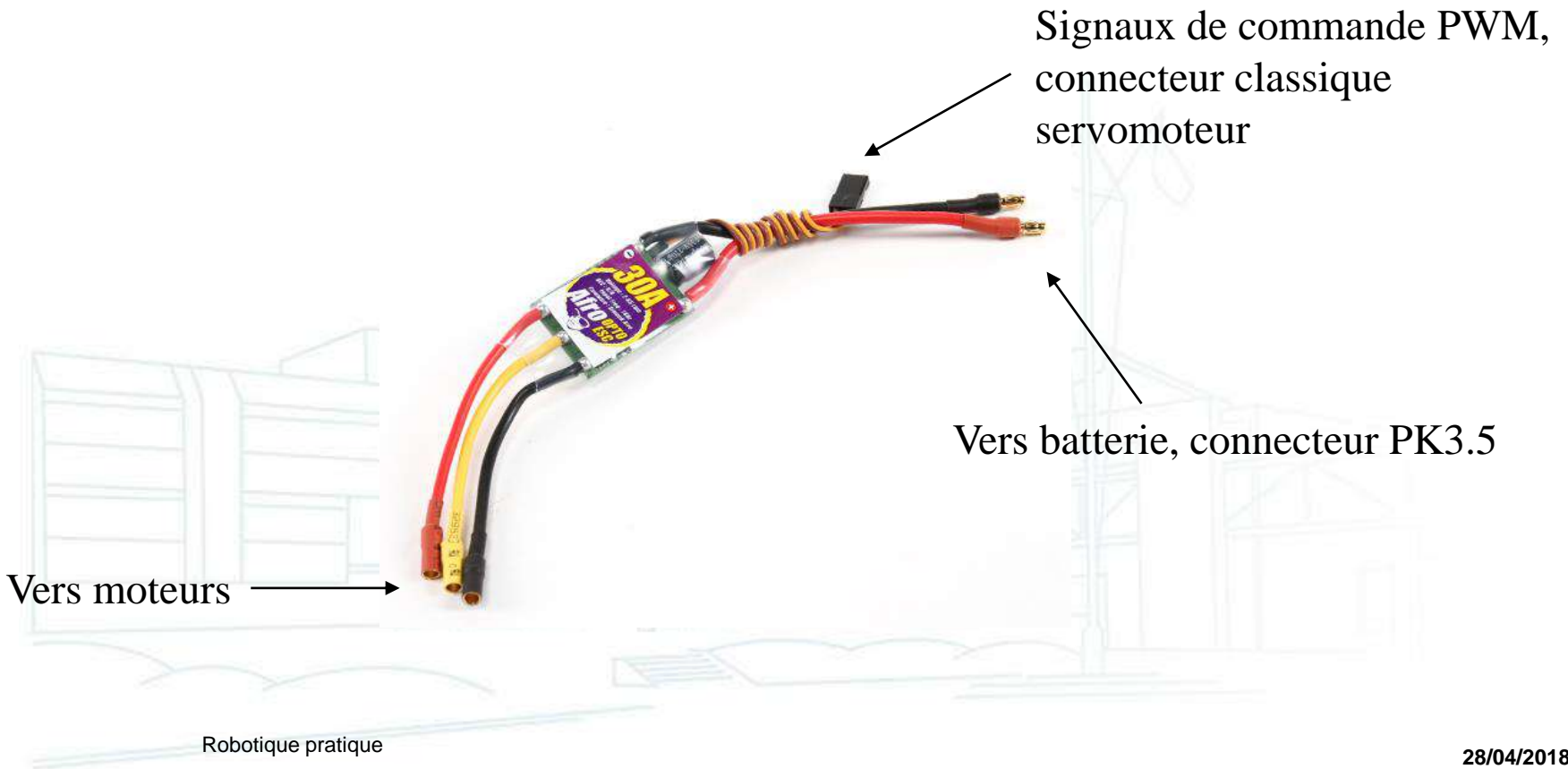
ESC pour moteurs brushless : en modélisme, souvent utilisés pour les avions, quadrirotors, hélicoptères donc il n'y a pas toujours de marche arrière...



- Actionneurs :
  - Carte de puissance/variateur/ESC  
Exemple : Robbe Rokraft (brushed motors)



- Actionneurs :
  - Carte de puissance/variateur/ESC  
Exemple : AfroESC (brushless motors)



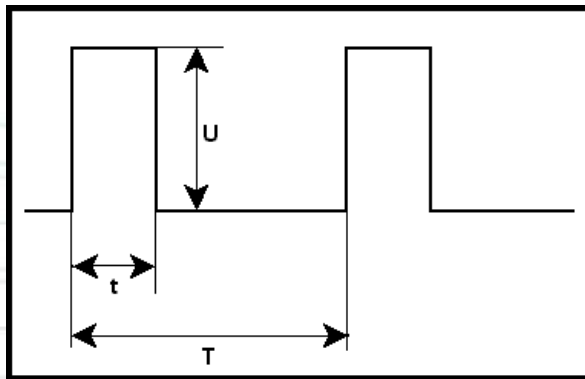


- Actionneurs :
  - Carte de puissance/variateur/ESC

Fonctionnement :

La puissance envoyée aux moteurs (et donc leur vitesse) dépend du signal de commande PWM

PWM = Pulse Width Modulation : modulation en largeur d'impulsion



**U : tension du PWM (5 V)**

**t : largeur d'impulsion (entre 1 et 2 ms)**

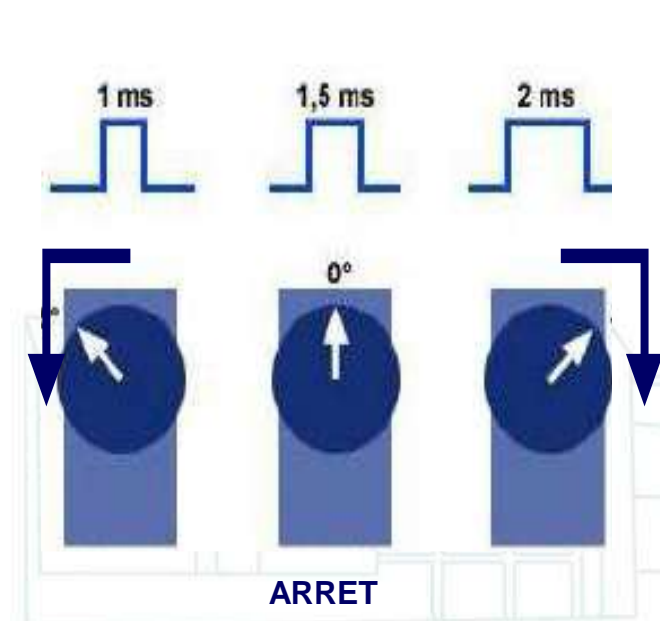
**T : période (20 ms)**

## ■ Actionneurs :

- Carte de puissance/variateur/ESC

Fonctionnement :

Correspondance largeur d'impulsion / vitesse de rotation

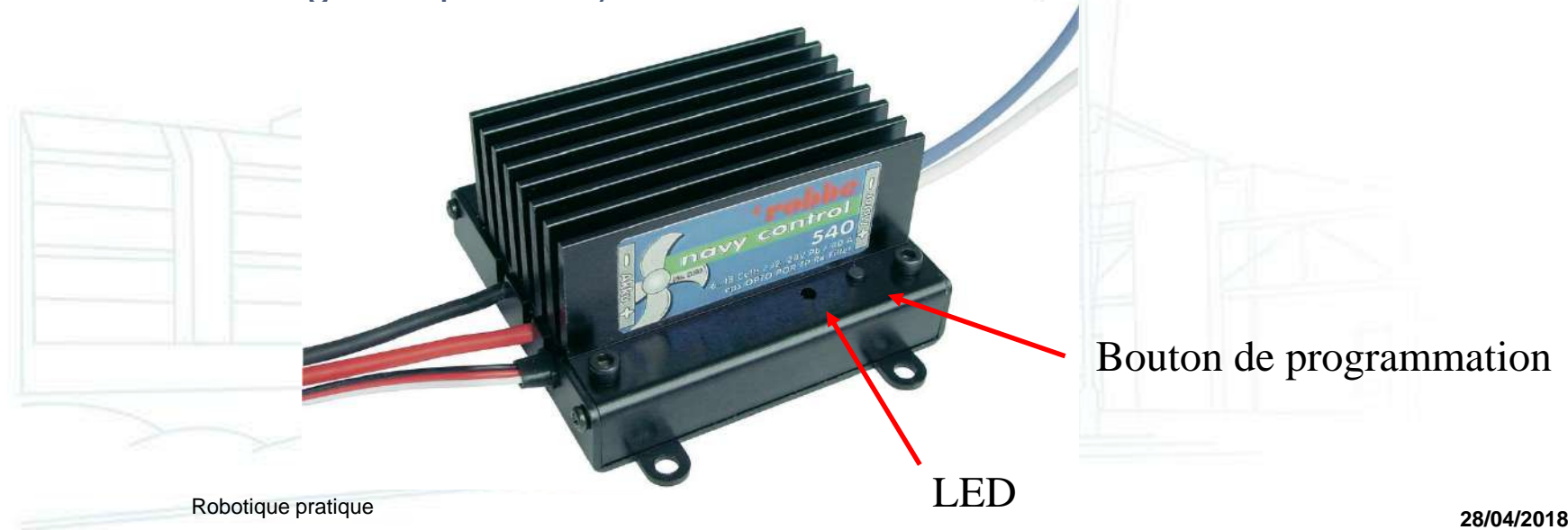


État du moteur	Largeur d'impulsion
Moteur à l'arrêt	1.5 ms
Rotation dans un sens, en accélérant	1.5 à 2.0 ms
Rotation dans le sens inverse, en décélérant	1.0 à 1.5 ms

- Actionneurs :
  - Carte de puissance/variateur/ESC

Calibration :

Certains ESC doivent être calibrés pour fonctionner : **réglage du max, du min, du neutre** en fonction du signal PWM, penser à **consulter la doc** (bouton ou procédure de démarrage spéciaux, peut nécessiter un adaptateur spécial pour connecter au PC avec un logiciel spécial...)



## ■ Actionneurs :

- Carte de puissance/variateur/ESC

BEC (Battery Elimination Circuit) :

Sur leur « **connecteur servo** », il est courant que **5 V (typ. 1-3 A)** soient fournis sur le fil rouge (générés à partir de la batterie moteurs via un **régulateur interne**)

En modélisme, le but est de pouvoir alimenter le récepteur et autres petits actionneurs sans avoir besoin d'ajouter une batterie 5 V

Certains ESC (souvent ceux avec des composants opto/photo) au contraire n'ont pas de BEC et nécessitent 5 V sur leur fil rouge pour fonctionner (ils n'utilisent pas la batterie moteurs pour leur propre alimentation)

- Actionneurs :
  - Carte de puissance/variateur/ESC  
BEC (Battery Elimination Circuit) :

**Attention : si plusieurs ESC avec BEC sont sur le robot, il ne faut pas que leurs BEC se retrouvent connectés => retirer le fil rouge du connecteur servo pour tous sauf 1 (ou tous si pas besoin de 5 V)!**

Toujours faire attention quand on a plusieurs sources d'alimentation en même temps!

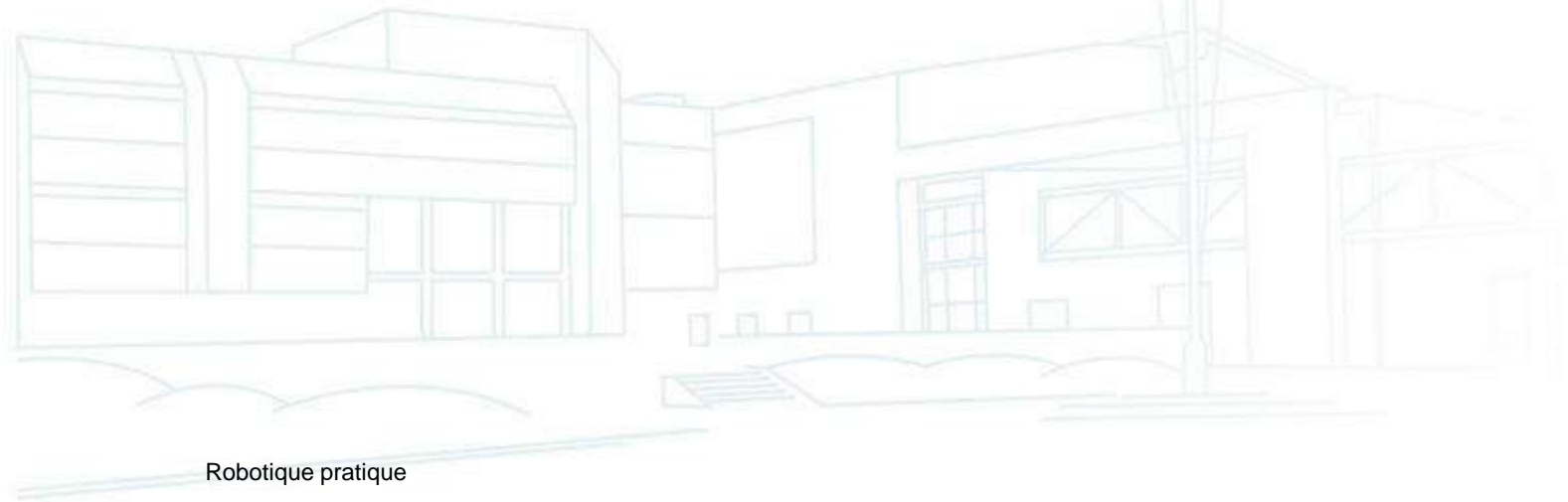
Attention aussi aux **boucles de + ou boucles de -** possibles lors de l'utilisation de SPS (interrupteurs électroniques de puissance), **vérifier** si **masse commune** possible/impossible/nécessaire, etc.

- Actionneurs :
  - Carte de puissance/variateur/ESC

Paramètres importants :

Le **courant** supporté par l'ESC doit être  $>$  à ce que le **moteur** peut demander

La **tension** supportée par l'ESC doit être  $\geq$  à celle de la **batterie** et doit être en accord avec ce qu'accepte le moteur et où il est le plus efficace (vitesse et couple max pour une conso et température acceptables)



## ■ Actionneurs :

- Servomoteur = petit moteur + carte de puissance

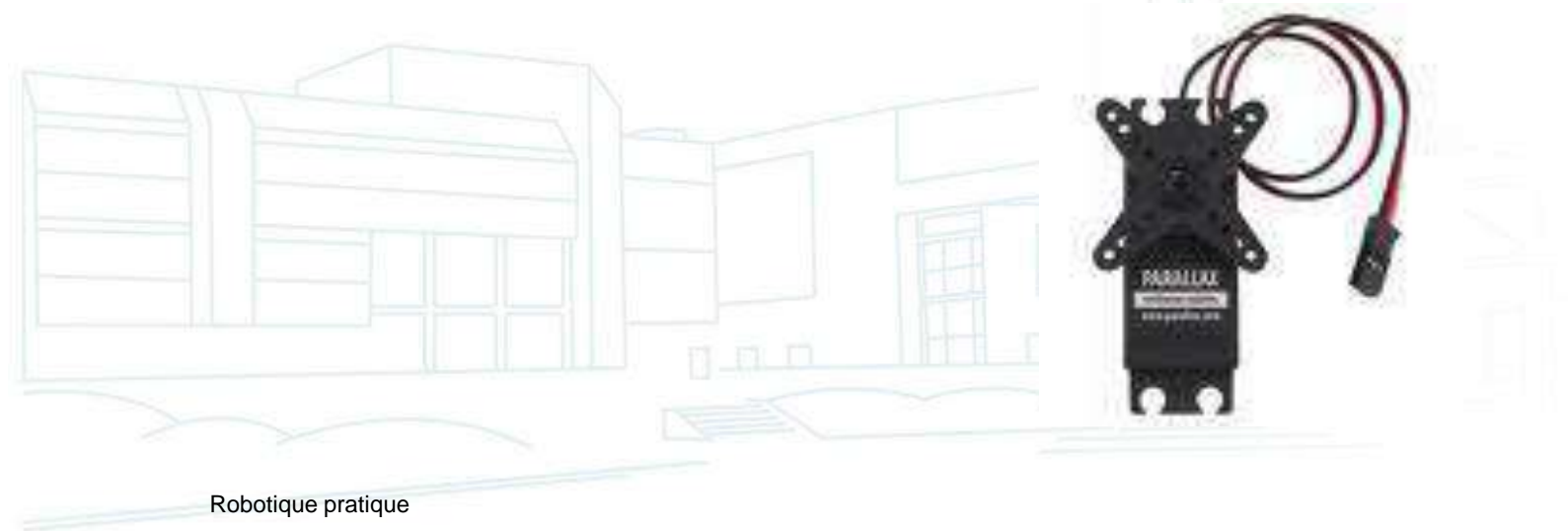
Commandé par PWM

Alimentation 5 V (< 2 A en général)

2 types de servomoteurs :

Asservis en position/angle : tournent de  $-40$  à  $+40^\circ$  par exemple

Asservis en vitesse



## ■ Actionneurs :

- Carte d'interface

Relie la partie informatique avec la partie électronique (capteurs, actionneurs)

Partie informatique : intelligence par le biais de programmes sur PC

Partie électronique : capteurs, actionneurs





## ■ Actionneurs :

- Carte d'interface

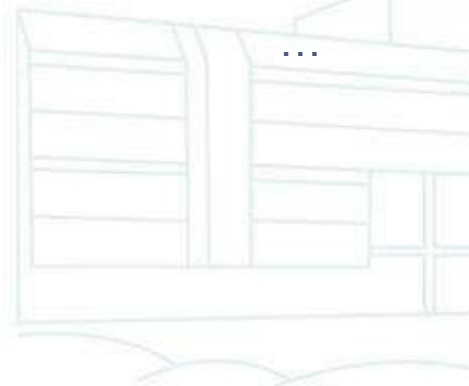
Exemple : carte IOIO pour smartphone/tablette Android, PC Windows/Linux

Se branche sur le port USB du smartphone/PC et est contrôlé par des programmes exécutés sur le smartphone/PC

Peut générer des signaux PWM, I2C

Peut générer et lire des signaux numériques

Peut lire des petites tensions (venant de capteurs analogiques tels que des télémètres, odomètres, boussoles...)

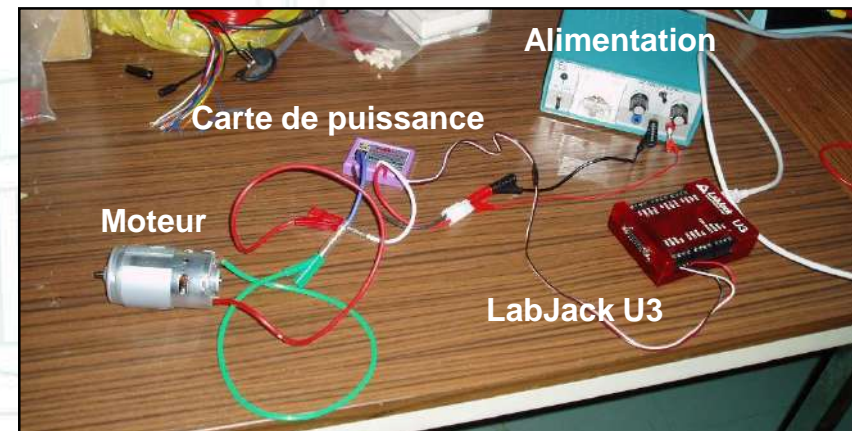
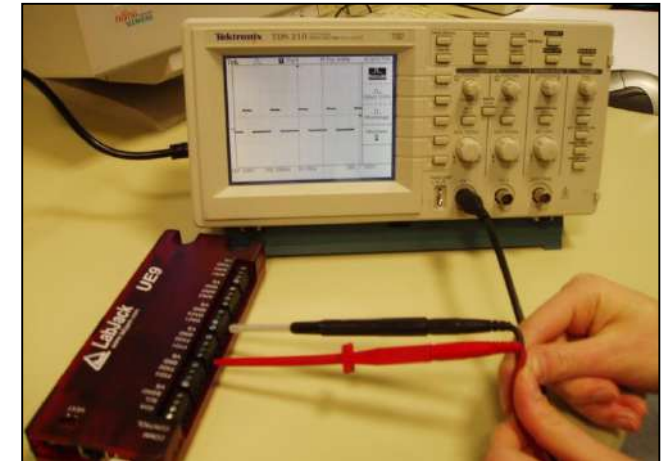


# Concevoir un robot

## ■ Actionneurs :

- Carte d'interface

Autres exemples : Cartes Pololu  
Maestro, SSC-32, LabJack pour PC

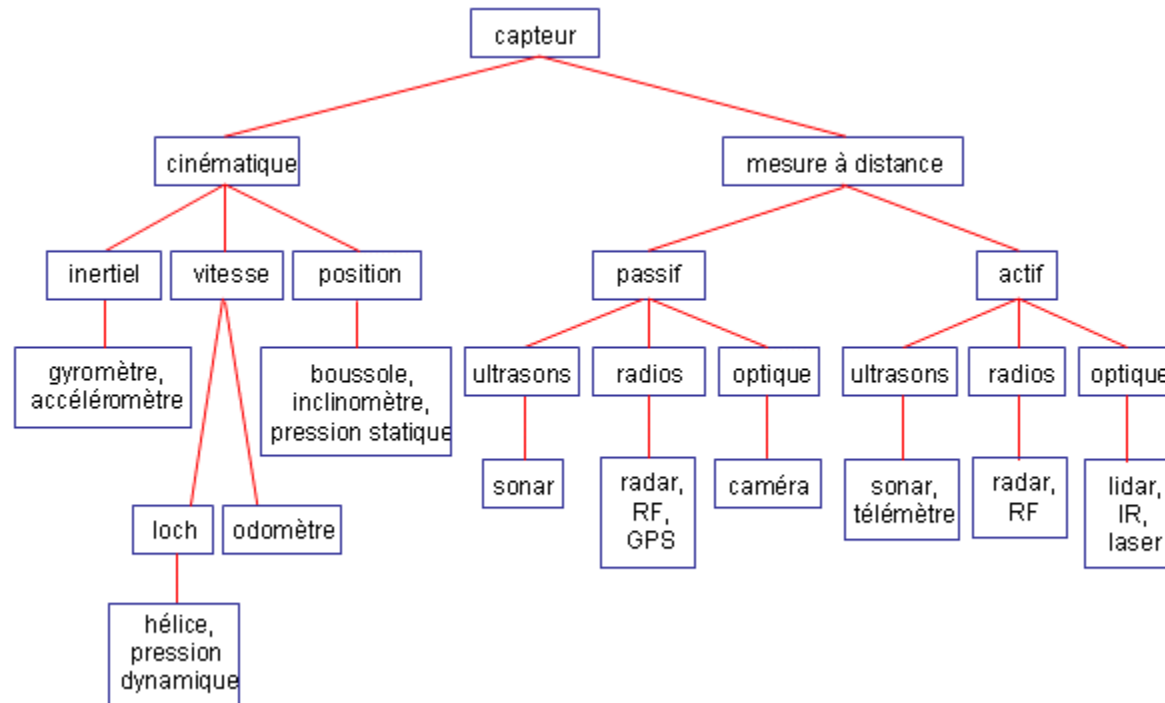


## ■ Capteurs :

- **Proprioceptifs** : fournissent une information sur l'état interne actuel du robot, par rapport à une référence interne
- **Extéroceptifs** : fournissent une information sur l'état du robot relativement à son environnement
- **Exproprioceptifs** : combinaison de proprioceptif et extéroceptif

capteur	sous-marin	aérien	proprioceptif	extéroceptif	exproprioceptif
boussole	+	+	+		
magnétomètre	+	+	+		
inclinomètre	+	+	+		
gyromètre	+	+	+		
accéléromètre	+	+	+		
GPS		+		+	
IMU/AHRS	+	+	+		
INS	+	+			+
odomètre		+	+		
altimètre barométrique		+	+		
capteur de pression	+		+		
loch hélice/pression	+	+	+		
DVL	+			+	
caméra	+	+		+	
sonar	+			+	
télémètre ultrason		+		+	
télémètre infrarouge		+		+	
télémètre laser		+		+	
radar		+		+	
lidar		+		+	
récepteur radios		+		+	

## ■ Capteurs :



## ■ Capteurs :

- **Boussole** (compass) : angle / Nord mag.
- **Magnétomètre** 3 axes / boussole 3D : mesure les 3 composantes du champ magnétique terrestre. Attention aux **perturbations** liées à pièces métalliques (même des ports USB peuvent être magnétisés), courants électriques...



**Perturbations** magnétiques dues aux **objets métalliques de l'environnement** proche : **difficile à corriger** mais on pourrait cartographier le champ magnétique

**Perturbations** dues aux **éléments constituant le robot** (peut varier selon la vitesse des moteurs...) : les **perturbations constantes** peuvent être prises en compte par une procédure de **calibration magnétique**

# Concevoir un robot

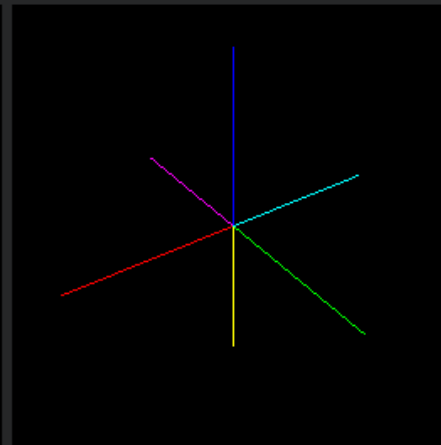
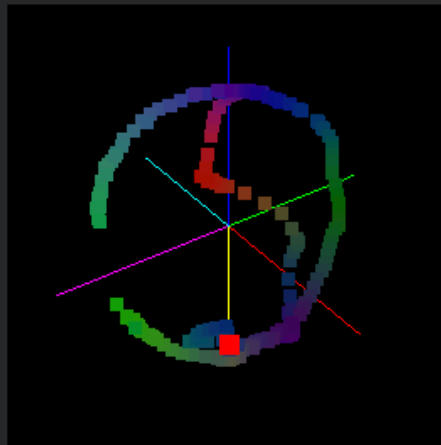
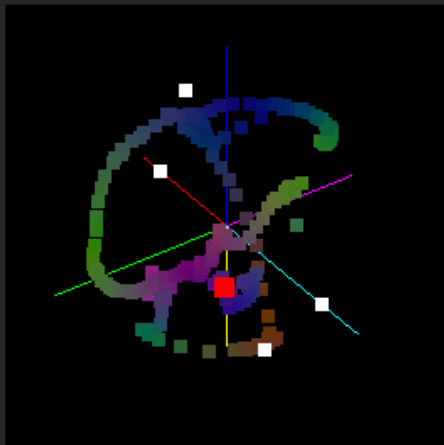
- Capteurs :
  - Calibration magnétique

Progress

Got + 679 samples  
Compass 1 error: 0.23  
Compass 2 error: 0.27  
Compass 3 error: 99  
more data needed Aim For Yellow-Green

Aim for the White dots.  
Please point the autopilot north, and rotate around  
the pitch axis until level.  
then  
Turn the autopilot 90 degrees, and rotate around the  
roll axis until level.

This method should hit every white dot.



☒ Rotate with each data point ☒ Use Auto Accept

Done





## ■ Capteurs :

- Inclinomètre : angle par rapport à l'horizontale
- Gyromètre : vitesse instantanée de rotation
- Accéléromètre : accélération instantanée en translation
- GPS : latitude, longitude, altitude (imprécis), vitesse

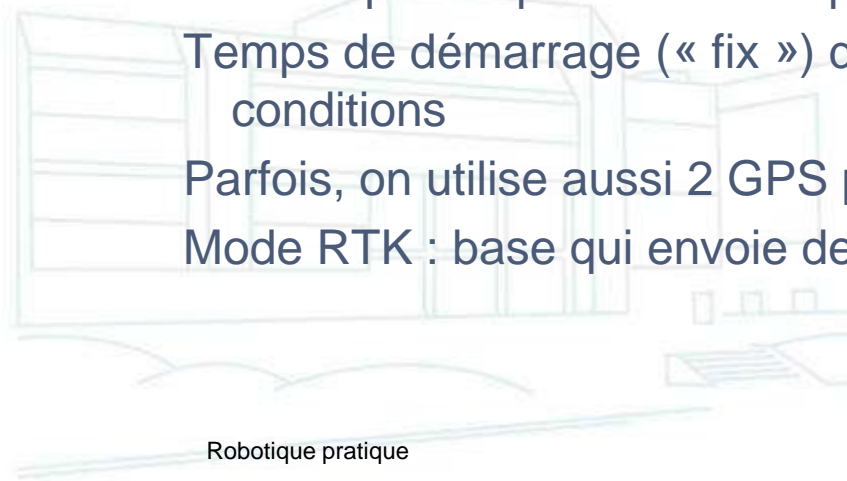
Ne fonctionne en général pas à l'intérieur ni sous l'eau (il faut qu'il ait une bonne « vue » directe des satellites dans le ciel)

Il se peut qu'il donne des positions aberrantes lorsqu'il est à la limite de ne plus capter ou si multiples réflexions sur obstacles

Temps de démarrage (« fix ») de plusieurs minutes variable selon les conditions

Parfois, on utilise aussi 2 GPS pour estimer son cap

Mode RTK : base qui envoie des corrections



## ■ Capteurs :

- IMU (Inertial Measurement Unit) : combinaison d'accéléros, gyros, magnétos
- AHRS (Attitude and Heading Reference System) : IMU avec traitement numérique global des différentes données
- INS (Inertial Navigation System) : AHRS calculant en plus positions et vitesses

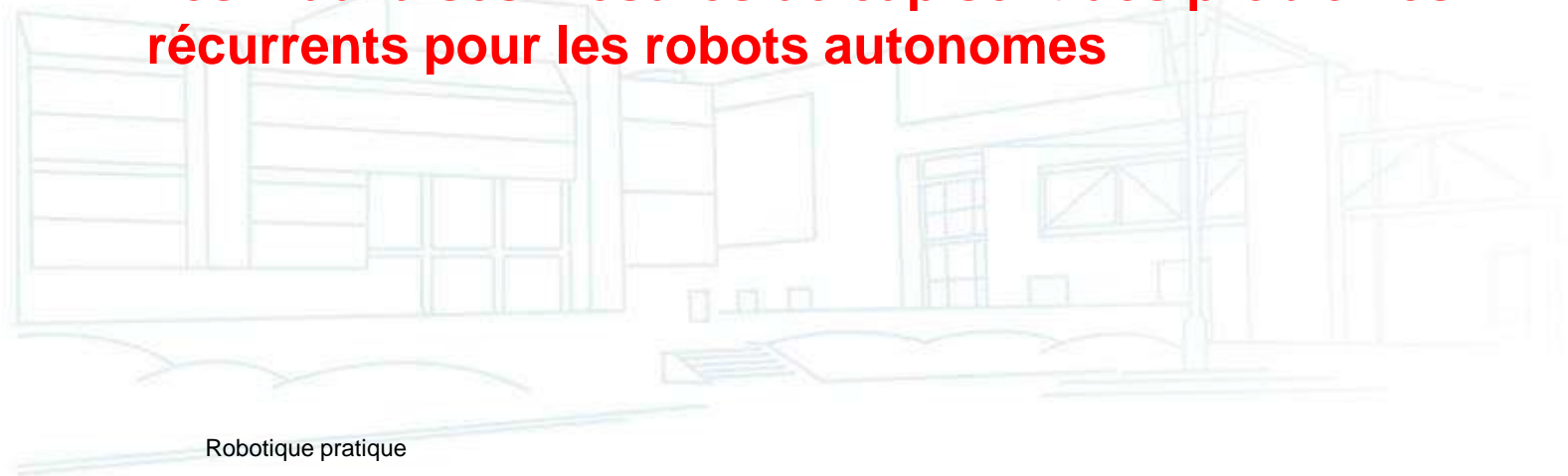
Souvent avec GPS intégré car obtenir ces valeurs à partir de données inertielles cause une accumulation des erreurs dues aux intégrations





## ■ Capteurs :

- Consulter la documentation pour voir les différentes calibrations possibles/nécessaires
- Si nécessaire, une procédure de calibration simple pour la mesure du cap avec boussole/IMU/AHRS peut être suivie : voir [http://www.ensta-bretagne.fr/lebars/these/thesis\\_2011-11-23.pdf](http://www.ensta-bretagne.fr/lebars/these/thesis_2011-11-23.pdf) (p64)
- Attention aux vibrations pour les capteurs inertiels!
- **Les mauvaises mesures de cap sont des problèmes récurrents pour les robots autonomes**



## ■ Capteurs :

- Types de communication :

Sorties analogiques (e.g. 0..5 V)

Sorties numériques :

P2P :

0 ou 1 sur un ou plusieurs bits en parallèle

Liaison série RS232/UART

Bus :

RS485/RS422

CAN

I2C/SMBus

SPI

USB

Réseau :

Ethernet, Wi-Fi

# Concevoir un robot

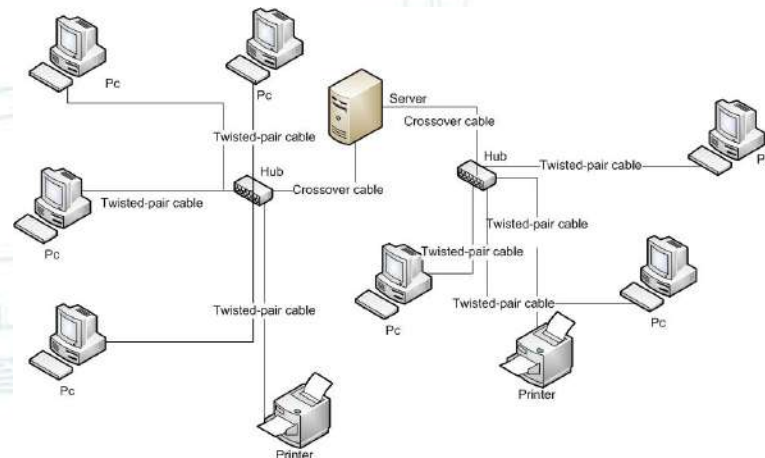
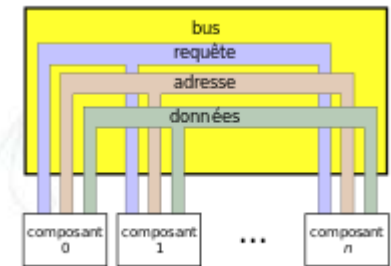
## ■ Capteurs :

- Types de communication :

**Liaison point à point (P2P)** : 2 périphériques sont en liaison exclusive

**Bus** : canal de communication partagé entre plusieurs périphériques, souvent avec des concepts de maître et esclaves

**Réseau** : plusieurs périphériques et plusieurs canaux

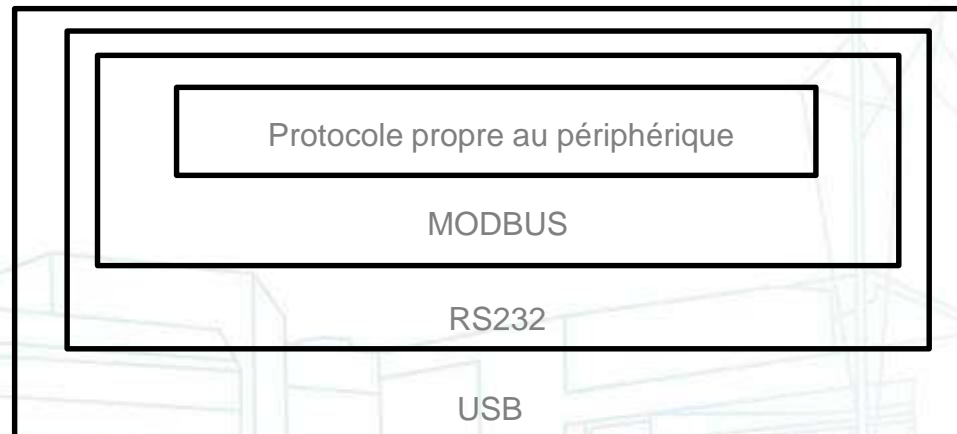


## ■ Capteurs :

- Types de communication :

Souvent couches englobantes de types de liaisons, protocoles, voir modèle OSI...

Exemple : capteur de pression Keller P33x



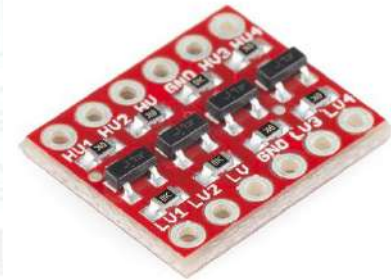
Utiliser au max les logiciels officiels fournis avec les capteurs/actionneurs pour vérifier déjà leur bon fonctionnement dans les conditions définies par le constructeur!

- Capteurs :
  - Types de communication :

**Attention : la tension max d'alimentation du capteur peut être différente de la tension max sur ses pins de communication!**

Exemple : on trouve souvent des capteurs alimentés en 5 V et avec une liaison UART avec des niveaux de tension variant entre 0 et 3.3 V

Voir <https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide>



# Concevoir un robot

## ■ Capteurs :

- Types de communication :

Liaison série RS232/UART

A l'origine, port DB9 directement sur PC

Maintenant, via convertisseurs USB

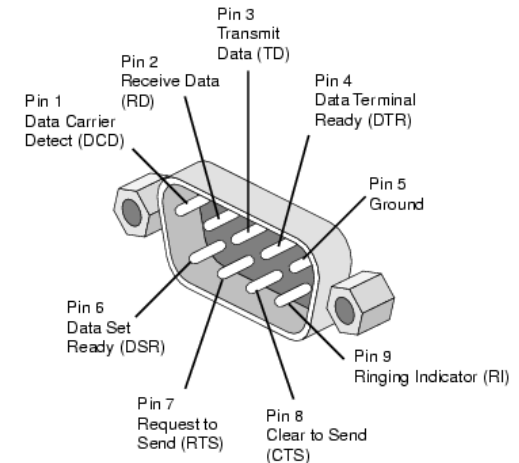
Niveaux de tensions standards PC : -12..+12 V

Sur cartes embarquées et petits capteurs :  
souvent niveaux TTL 0..5 V ou 0..3.3V, à vérifier

Adaptateurs USB-RS232 PC : FTDI, Prolific

Adaptateurs USB-RS232 TTL : FTDI

Adaptateurs RS232 PC-RS232 TTL : MAX232



Voir aussi <https://learn.sparkfun.com/tutorials/terminal-basics>

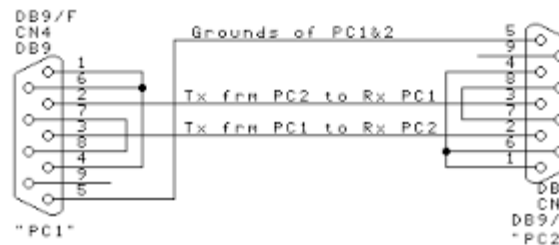
## ■ Capteurs :

- Types de communication :

Liaison série RS232/UART

Fils importants : **GND, TX (Transmit), RX (Receive)**

**Le TX du PC se met sur le RX du capteur, et inversement!**



Si le capteur ne fait qu'envoyer des données (e.g. GPS), on peut éventuellement se passer de son RX, idem pour des actionneurs avec TX

Pour les mises à jour de firmware de capteurs, les fils RTS, CTS, DTR, etc. peuvent parfois être aussi nécessaires

Inversement, parfois ces fils peuvent gêner et doivent être déconnectés...



# Concevoir un robot

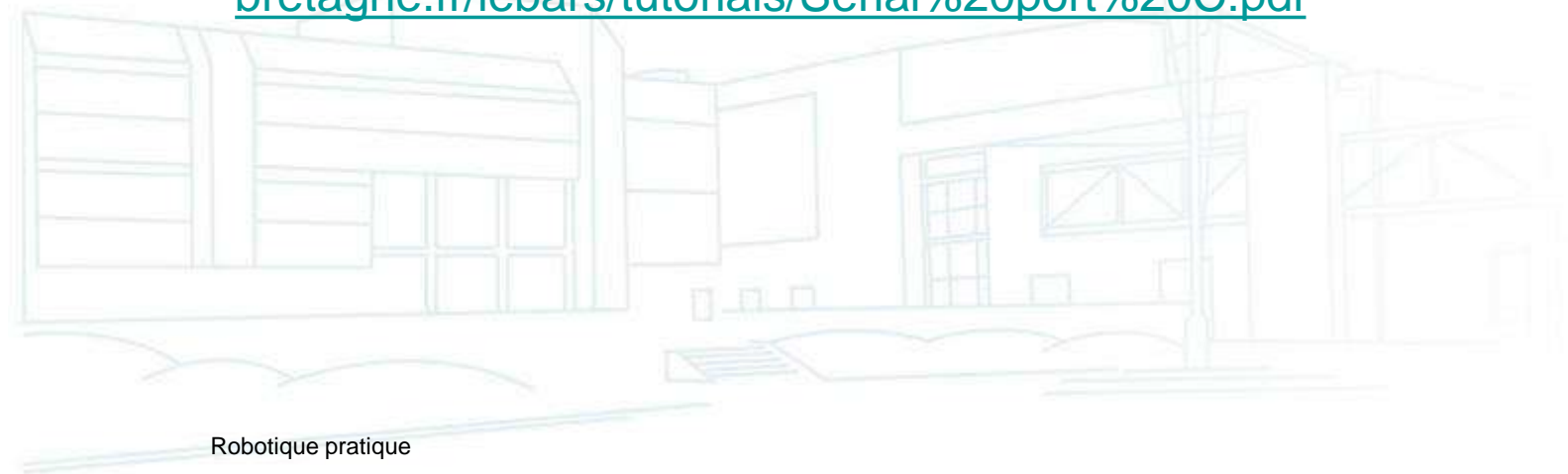
## ■ Capteurs :

- Types de communication :

Liaison série RS232/UART

Selon le protocole du périphérique, il est possible de recevoir/envoyer facilement des données avec **HyperTerminal** (Windows, voir <http://www.ensta-bretagne.fr/lebars/Share/hypertrm.zip>) ou **minicom**, **gtkterm** (Linux)

Voir aussi : <http://www.ensta-bretagne.fr/lebars/tutorials/Serial%20port%20C.pdf>





## ■ Capteurs :

- Types de communication :

Bus RS485/RS422

Fils importants : **A+**, **A-**, **B+**, **B-**, appelés parfois aussi RX+, RX-, TX+, TX-

Pas de GND car les niveaux de tensions sont mesurés entre les – et + : signal dit symétrique/différentiel, contrairement au RS232 où c'est asymétrique car la référence est GND pour le TX et le RX

**Signal différentiel** => plus robuste aux interférences

**Full-duplex** : comporte une ligne de réception et une de transmission séparée (besoin des **4 fils**)

**Half-duplex** : se sert de la même ligne pour la réception et l'envoi et doit donc commuter entre émetteur et récepteur à chaque changement (besoin de **2 fils**)

# Concevoir un robot

## ■ Capteurs :

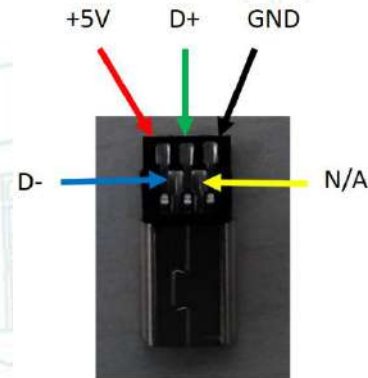
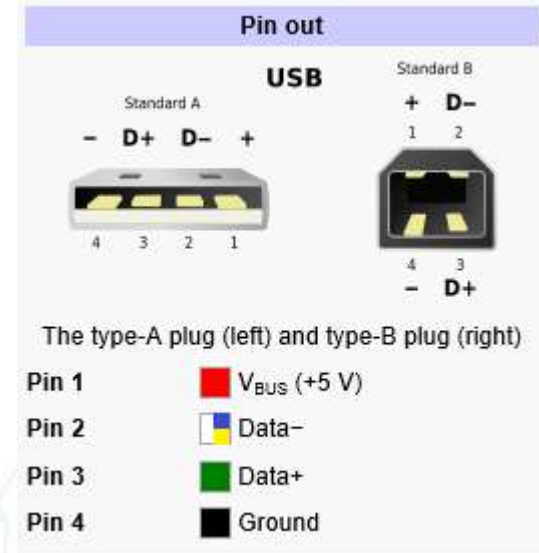
- Types de communication :

Bus USB 2.0

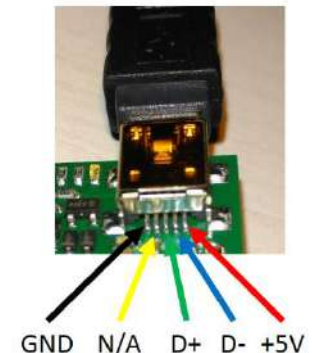
Fils : **GND, D+, D-, +5 V**

Grand nombre de **connecteurs différents**, à vérifier

Utiliser des **câbles de qualité** (blindé, etc.), éviter de couper les câbles et de passer par des connecteurs non-standards, signal très sensible en pratique...



Adafruit Slim MiniB DIY #1825



Mini USB port on  
Pololu Micro Maestro 6

## ■ Capteurs :

- Types de communication :

Bus USB 2.0

**USB host** vs **USB device** : PC=host, capteur=device en général

Smartphones/tablettes : peuvent parfois être dans un mode ou l'autre

Alimentation d'un device :

En général fournie par host

Si conso < 100 mA pas de problèmes en général

Autrement, une négociation se fait entre host et device pour obtenir jusqu'à 500 mA

Un hub USB non alimenté ne peut fournir plus de 100 mA par port

- Capteurs :
  - Types de communication :  
Bus USB 2.0

**Brancher un périphérique qui consomme trop peut endommager l'ordinateur!**

Attention : les PC ont souvent déjà des hubs intégrés, ceci a aussi des conséquences sur les débits max qu'on peut espérer (480 Mbits/s pour un « vrai » port USB 2.0)!

Attention : la rétro-compatibilité de l'USB 3.0 vers l'USB 2.0 ne marche pas toujours très bien, essayer de privilégier les ports USB 2.0 pour les périphériques USB 2.0...

## ■ Capteurs :

- Types de communication :

### Bus USB

C'est un **bus très complexe** avec ses gestions de versions, débits, alimentations, etc. => on ne code jamais directement avec son protocole, il y a toujours un protocole plus simple au-dessus et des **drivers et bibliothèques spécifiques** à utiliser (quand ce n'est pas juste une conversion vers RS232, RS485)...

e.g. : Webcams (avec OpenCV), Labjack UE9 (avec libUSB, voir constructeur), Kinect 2 (avec Kinect SDK)



## ■ Capteurs :

- Types de communication :

Bus I2C/SMBus

3 fils : **GND**, **SDA (données)** et **SCL (horloge)**

Sous Linux, un driver, des commandes, des API permettent d'utiliser ce bus si présent sur l'ordinateur



## ■ Capteurs :

- Types de communication :

Numéros de port (peuvent changer après redémarrage ou reconnexion),  
vitesse de bus, bits de start, stop, parité/correction d'erreur...

**Plusieurs drivers différents** peuvent exister pour un même périphérique selon les besoins, vérifier s'il n'y a pas une autre version en cas de problèmes

**Attention** : il n'est pas toujours possible d'utiliser **plusieurs applications pour accéder en même temps** au même périphérique, vérifier s'il n'y a pas un processus en cours oublié en cas de problèmes!

Voir aussi **Virtual Serial Port Emulator** pour certaines manipulations avec des ports série



## ■ Capteurs :

- Programmation :

**Délais de communication** sur les périphériques font que s'il y en a plusieurs, il faut essayer de paralléliser pour que la boucle de régulation tourne à une fréquence suffisante => Programmation **multithread**

Bien regarder les documentations et **codes d'erreurs retournés par les fonctions** si on doit faire un code fiable. En particulier les fonctions de type **read()**, **write()** ne retournent pas forcément le nombre d'octets demandés (et ce n'est pas forcément considéré comme une erreur), surveiller les problèmes de **timeout**, etc.

De même, il faut rechercher la bonne trame, qui ne commence pas forcément par le début des données récupérées!

Bien vérifier comment interpréter un octet (**binaire vs texte**), un paquet (**endianness**)...



## ■ Ordinateur embarqué :

- Intelligence du robot

Contient les programmes définissant le comportement du robot

En général, tous les capteurs, actionneurs, dispositifs de communication y sont relié directement ou indirectement

Sert aussi pour la sauvegarde de données, même s'il est souvent plus sûr de les envoyer à la station de contrôle distante si risque de perte du robot...

Si pas assez puissant, on peut tenter de déporter certains calculs vers la station de contrôle distante

- Par abus de langage, on confond souvent « ordinateur », « PC », « intelligence »...

# Concevoir un robot

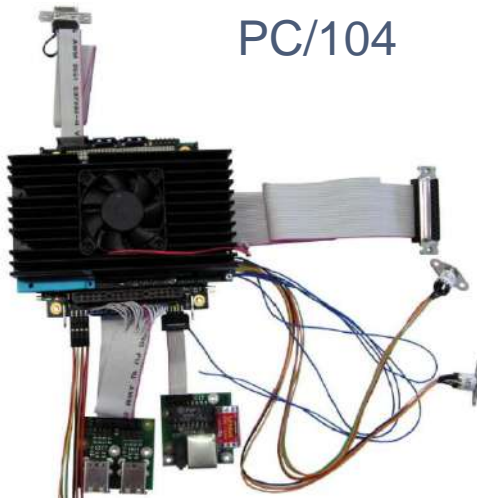
- Ordinateur embarqué :
  - Exemples : PC i.e. ordinateur avec processeur compatible Intel x86 avec BIOS/UEFI, etc.

HTPC (Home Theater PC)

EeePC 901 (netbook)

Mini ITX

PC/104



Computer form factors	
Name	Size (mm)
NUC	116.6 x 112 x 34.5
Compute Stick	103.3 x 12.5 x 37.6
Zotac Pico	66 x 19.2 x 115.2
eeePC 901	226 x 175.3 x 22.9
Mini TX	170 x 170
Nano ITX	120 x 120
Pico ITX	100 x 72
PC/104	96 x 90

# Concevoir un robot

- Ordinateur embarqué :
  - Exemples : ordinateurs avec processeur ARM
    - Smartphone / tablette
      - Smartphone Samsung Galaxy S sous Android (avec GPS, boussole, caméra, Wi-Fi, etc. déjà intégrés)
    - Raspberry Pi



# Concevoir un robot

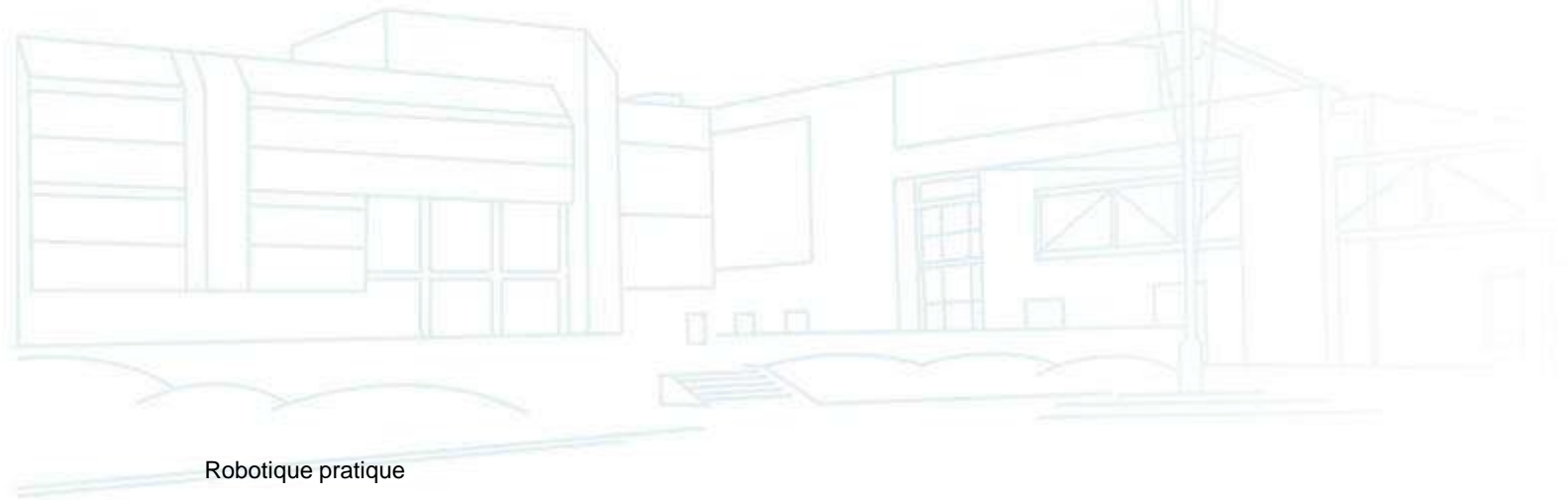
- Ordinateur embarqué :
  - Autres : Arduino, autopilotes (certaines ont souvent déjà des sorties PWM, etc.)...



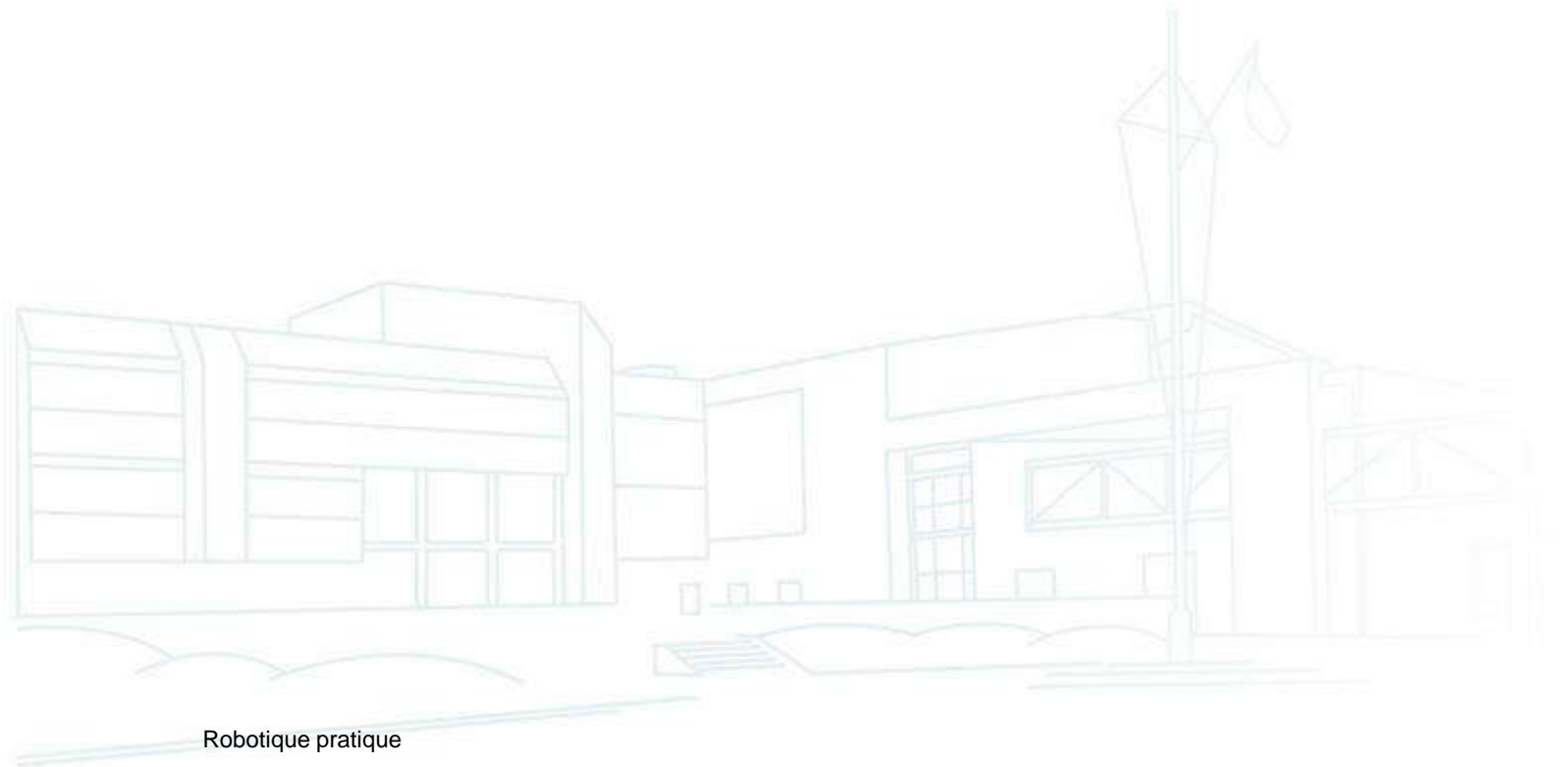
## ■ Ordinateur embarqué :

- **Attention** : même si les ordinateurs et autres cartes embarquées sont de plus en plus petits et pas chers, bien prendre en compte la **taille et le prix des connecteurs** et éventuels **adaptateurs** nécessaires!

e.g. Raspberry Pi Zero (5\$) + hub USB (5\$) + adaptateur micro USB (1\$) + adaptateur mini HDMI (5\$) + clé Wi-Fi (5\$) + clavier/souris (5\$)...



- Ordinateur embarqué :
  - Différents port pour écran : VGA, HDMI, DVI, DisplayPort, Thunderbolt à ne pas confondre avec certains port USB!



## ■ Communication :

- Commande des actionneurs

Récepteurs de modélisme côté robot, télécommande côté opérateur

Pas besoin de PC, est fait pour être relié directement aux ESC et/ou servos

Typ. 4-8 PWM

Fréquences classiques : actuellement 2.4 GHz, 5.8 GHz, 433 MHz, anciennement 27, 35, 40, 41 MHz

Types de modulation/codage variables (FM, AM, SSS, SPCM, PPM), plusieurs canaux par fréquences (vérifier quartz si applicable)





# Concevoir un robot

## ■ Communication :

- Télémétrie : infos orientation, position, vitesse, batterie, etc.

Autopilote et/ou module spécifique côté robot, module spécifique sur PC ou autre côté opérateur

Fréquences classiques : 2.4 GHz, 5.8 GHz, 433 MHz



- FPV (First Person View) : caméra à distance

Caméra avec module spécifique côté robot, lunettes ou écran côté opérateur

Fréquences classiques : 2.4 GHz, 5.8 GHz





# Concevoir un robot

- Communication :
  - Autres : communications bidirectionnelles

Wi-Fi



3G



XBee



Satellite (e.g. Iridium)



AIS (sur bateaux)



Modems acoustiques (sous-marins)



## ■ Communication :

- **Portée vs débit vs puissance d'émission vs gamme de fréquence vs taille antenne**

Plus la **fréquence** est **grande**, plus on peut espérer avoir des **hauts débits**

Plus la **fréquence** est **faible**, plus on peut espérer avoir une **grande portée** avec une **puissance d'émission faible** (donc conso énergétique plus faible)

La taille de l'antenne diminue lorsque la fréquence augmente

Utiliser des antennes directives pour maximiser portée, par contre, il faudra l'asservir précisément dans la bonne direction

## ■ Communication :

### • Perturbations :

La présence d'une antenne même non utilisée peut perturber

Un Wi-Fi puissant peut perturber un GPS, même s'ils ne sont pas à la même fréquence...

Certains types d'obstacles (grilles métalliques) peuvent être très perturbants

Plus l'antenne est loin du sol, mieux c'est...

Canaux Wi-Fi : parfois il est plus intelligent de se placer sur le **même canal** que les autres, les **protocoles sont faits pour gérer cela**, alors que si on est sur un canal différent mais proche, il y aura des interférences incohérentes...

## ■ Communication :

- Air vs eau :

Sous l'eau, les ondes électromagnétiques ne passent pas bien :

En **eau douce**, on peut quand même espérer communiquer sur **quelques m** à des fréquences de **quelques dizaines de MHz**, par contre en eau de mer, il y a beaucoup plus d'atténuation

De même, la visibilité est réduite sous l'eau donc les communications optiques sont aussi assez limitées :

En général on distingue mieux les objets jaunes ou orange car ils contrastent fortement avec la couleur de l'eau

Par contre, c'est plutôt le bleu-vert qui est le moins absorbé par l'eau, malheureusement il contraste peu avec la couleur de l'eau

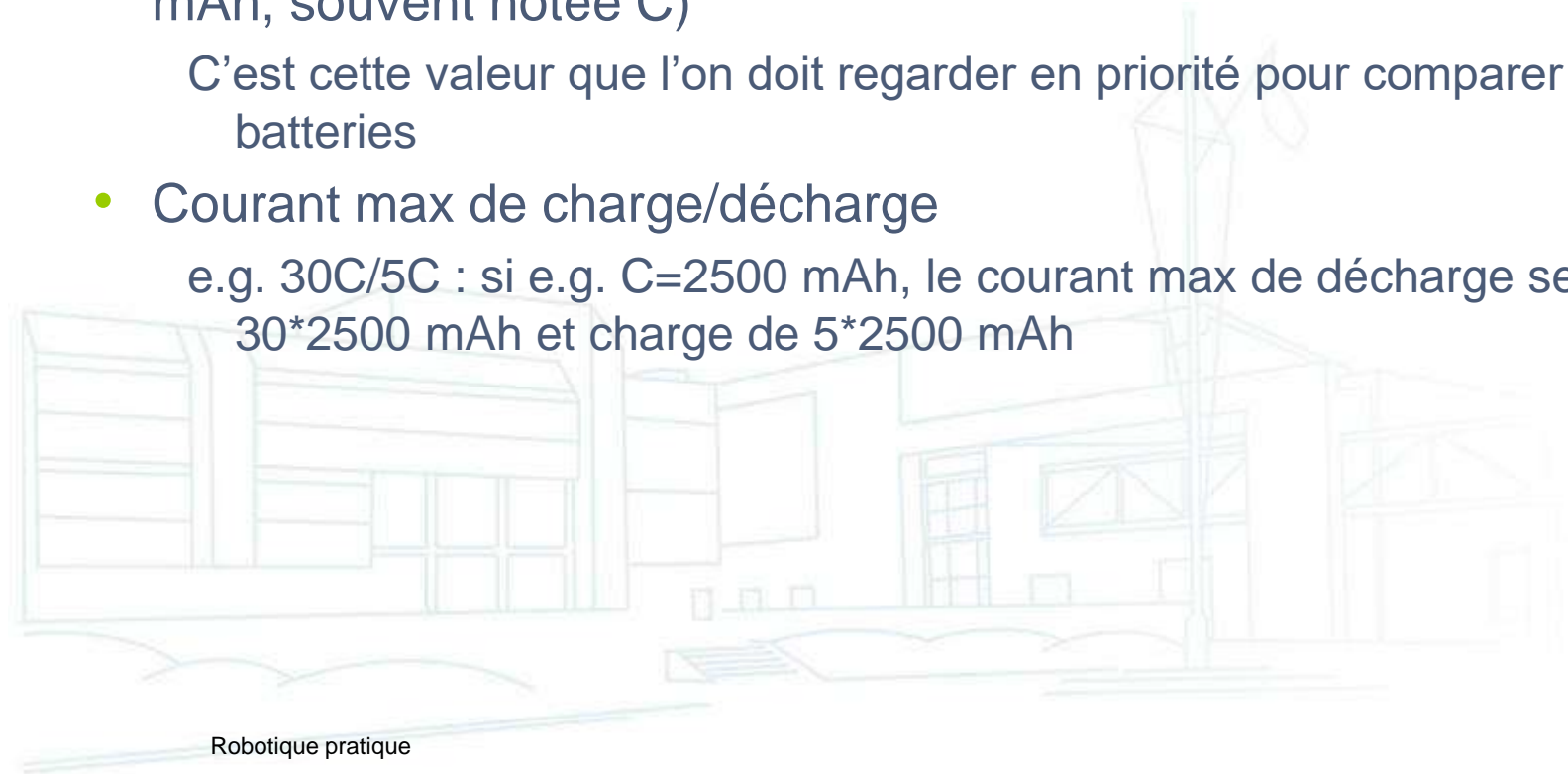
## ■ Energie :

- Type de batteries électriques classiques : **Li-Po**, Li-Ion, **NiMH**, Pb
- Energie (en Wh) = Tension (en V) \* Capacité (en Ah ou souvent mAh, souvent notée C)

C'est cette valeur que l'on doit regarder en priorité pour comparer 2 batteries

- Courant max de charge/décharge

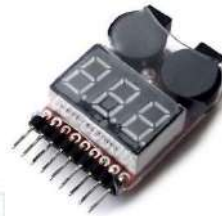
e.g. 30C/5C : si e.g. C=2500 mAh, le courant max de décharge serait 30\*2500 mAh et charge de 5\*2500 mAh



## ■ Energie :

- Cellules/éléments : 3S2P => 2 blocs parallèles de 3 éléments en série (S pour série, P pour parallèle)
- La tension d'une cellule/élément dépend du type de la batterie e.g. 3.7 V pour Li-Po, 1.2 V pour NiMH

**Attention : si la tension d'un élément descend trop bas (e.g. < 3.5 V pour Li-Po), elle peut y rester pour toujours! => Utiliser des checker/alarm!**



**Ne pas oublier de le débrancher après usage car il consomme aussi un peu d'énergie!**

- Voir [http://www.ensta-bretagne.fr/lebars/robots\\_batteries.pdf](http://www.ensta-bretagne.fr/lebars/robots_batteries.pdf)

## ■ Energie :

- La plupart des batteries doivent être chargées avec des **chargeurs intelligents** suivant une courbe de charge courant/tension qui dépend du type de batterie et de son état  
<https://youtu.be/vzAOG9ctZRU>
- **Voltmètre** ou checker : **permet d'affirmer** qu'une batterie est **déchargée**, mais n'est **pas fiable** pour dire qu'elle est bien chargée!
- En particulier, les **Li-Po** constituées de plusieurs éléments ont souvent un **connecteur d'équilibrage** qui permet au chargeur de réajuster chaque élément indépendamment
- **Ecrasement, surcharge, court-circuit = danger!**  
<https://www.youtube.com/watch?v=gz3hCqjk4yc>  
<https://www.youtube.com/watch?v=-DcpANRFrl4>



# Concevoir un robot





# Concevoir un robot

## ■ Connecteurs classiques de modélisme :

- Puissance :

T/Deans



XT60



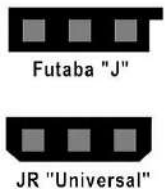
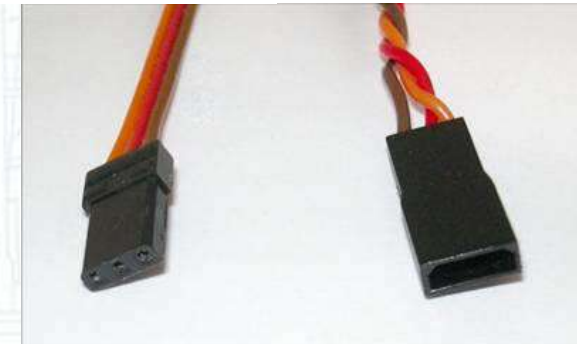
PK3.5 (conseillé pour les moteurs brushless)



- Contrôle ou faible puissance :

**Servo** (attention, parfois certaines variantes ne se connectent pas bien) : **3 fils** avec en général **noir/marron GND**, **rouge +5V**, **blanc/jaune/orange signal PWM**

**BEC JST** : 2 fils, GND et souvent +5V (vérifier)



# Concevoir un robot

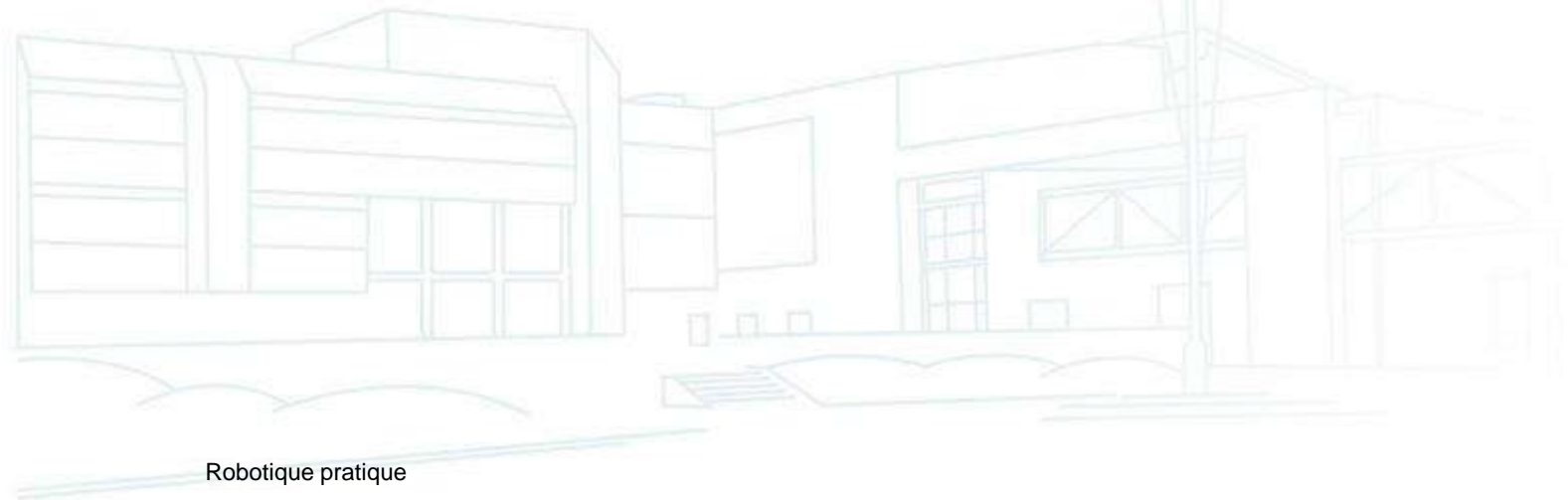
- Connecteurs classiques de modélisme :

- Conventions usuelles :

- [http://www.ensta-bretagne.fr/lebars/conventions\\_connecteurs.pdf](http://www.ensta-bretagne.fr/lebars/conventions_connecteurs.pdf)

- Voir aussi :

- <http://www.dronetrest.com/t/wires-connectors-and-current-what-you-need-to-know-as-a-drone-builder/1342>

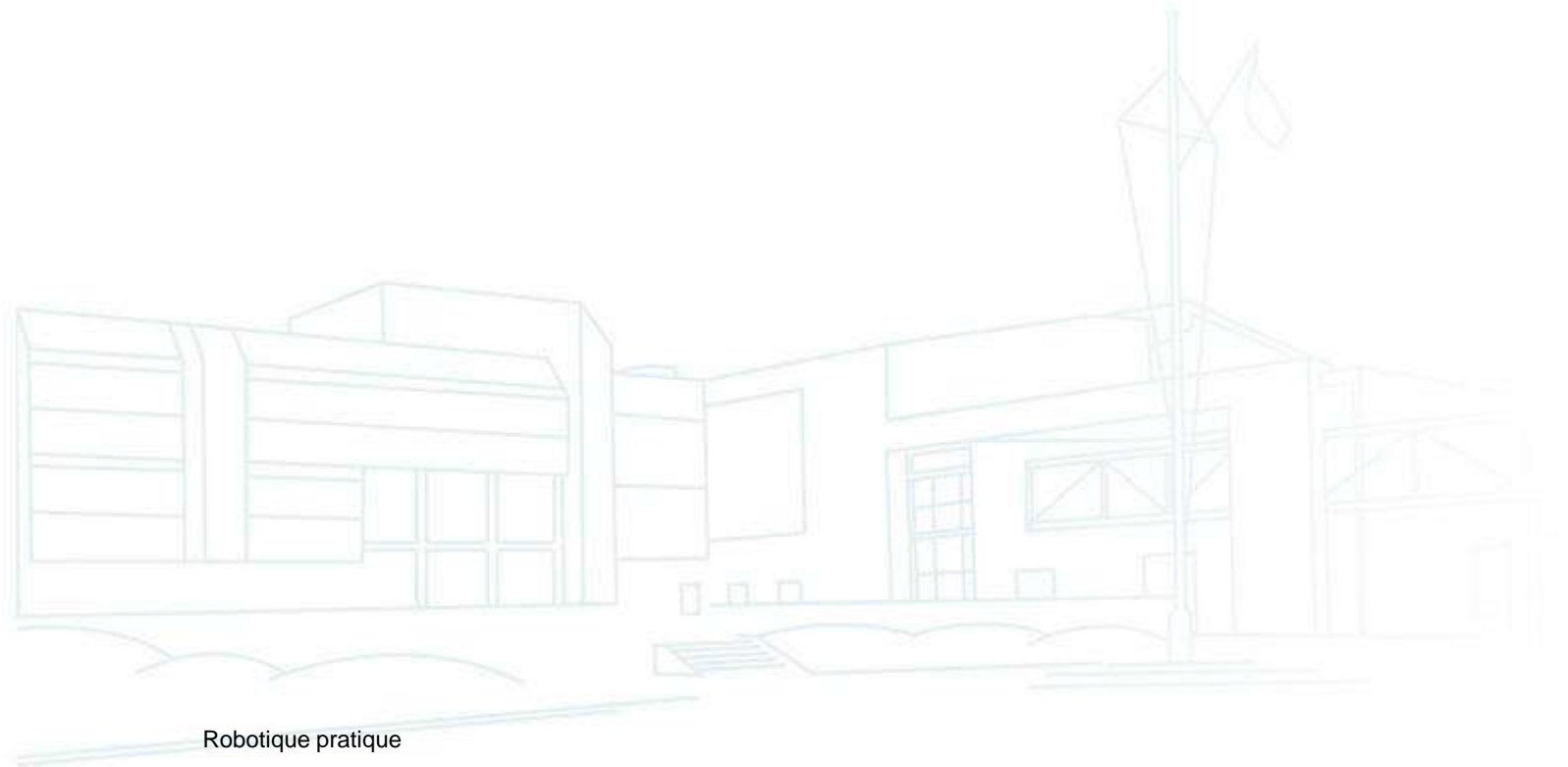


## ■ Câblage :

- Attention au **diamètre des câbles** et la **qualité des connecteurs** lorsque les **courants** deviennent **forts** (e.g.  $> 5\text{ A}$ )
- Les tensions diminuent avec la longueur des câbles (e.g.  $-1\text{ V}$  tous les 10 m selon la résistance, courant, etc.)
- Si les courants sont forts des **perturbations magnétiques** peuvent arriver au niveau de la boussole du robot => parfois **torsader** peut limiter ces perturbations
- Certains signaux numériques doivent aussi être sur des **paires torsadées** pour éviter certaines perturbations : e.g. Ethernet, RS232, PWM (câbles servos  $> 25\text{ cm}$ , on peut aussi utiliser une ferrite pour filtrer certaines perturbations)...
- Pour les **déports d'antennes**, il faut des **câbles et connecteurs coaxiaux**, vérifier quelles sont leur **fréquence max** et leur **impédance**
- Il est souvent conseillé d'avoir une **disposition des masses en étoile** plutôt qu'avec des boucles...

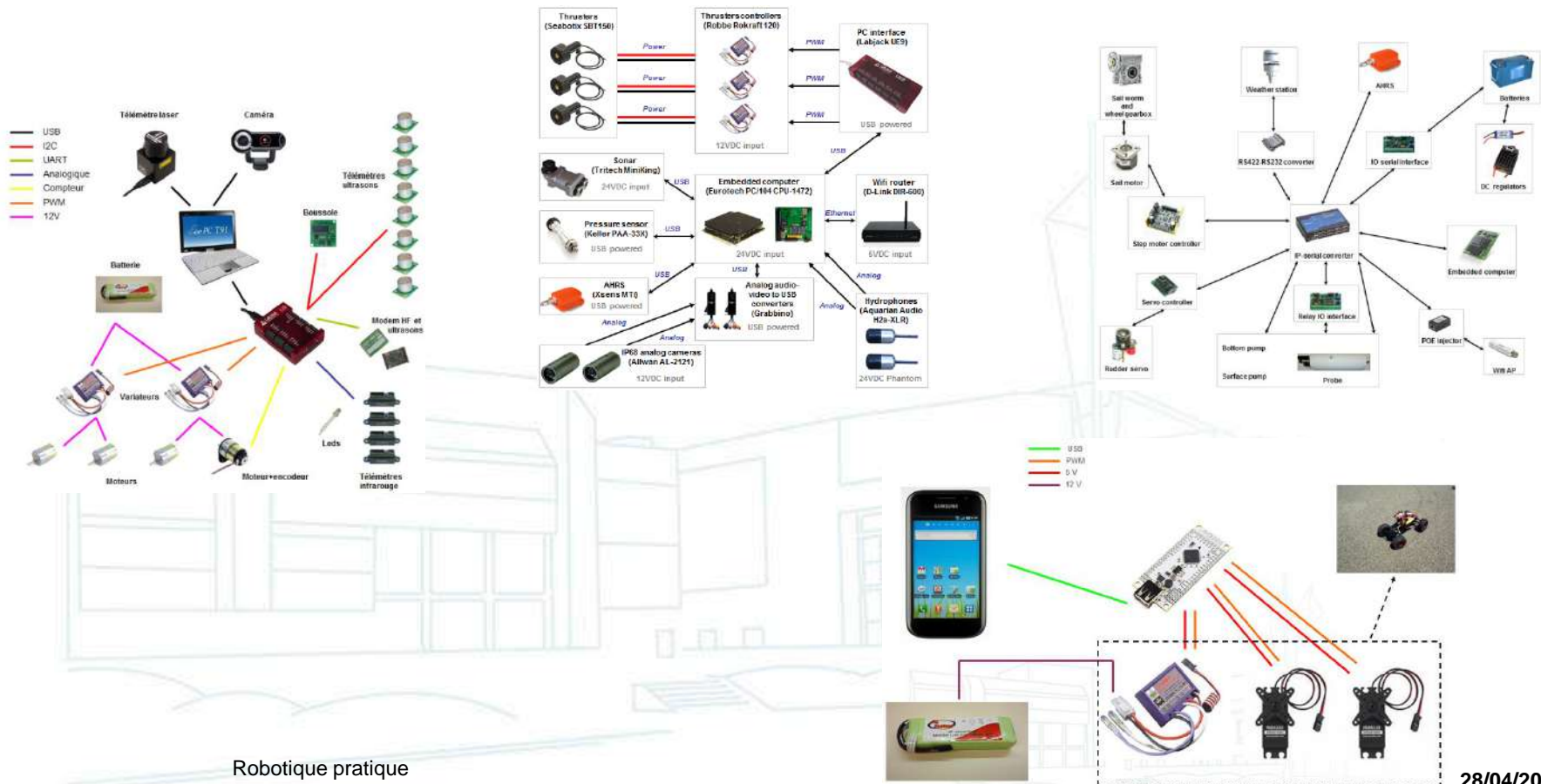
- Manipulations, interventions, tests :
  - **Débrancher la batterie** en 1er dès qu'on fait une **intervention technique**!
  - **Ne rebrancher que le minimum** quand on n'est pas sûr de ce qu'on fait/de l'état du matériel **pour limiter les dégâts** en cas de problème!
  - Attention si l'espace de travail contient des **objets métalliques**: **risques de court-circuit**!
  - Précautions hélices à enlever, protections sur banc de test en cas de projections...
  - **Filmer** et prendre des **photos** à chaque étape du montage/démontage/test
  - Faire checklist d'utilisation de robot, de départ en test en mer, etc.

- Boucle de conception de robot :
  - En particulier pour les drones (mais pas seulement), problèmes interdépendants pour le poids, surface alaire, motorization, hélice, batterie, etc.



# Concevoir un robot

## ■ Exemples d'architectures pour différents types de robots





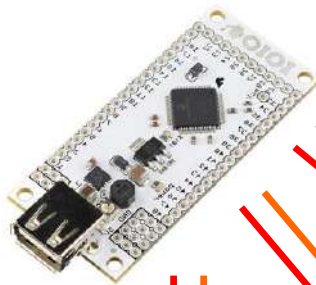
# AGV Buggy (2012)



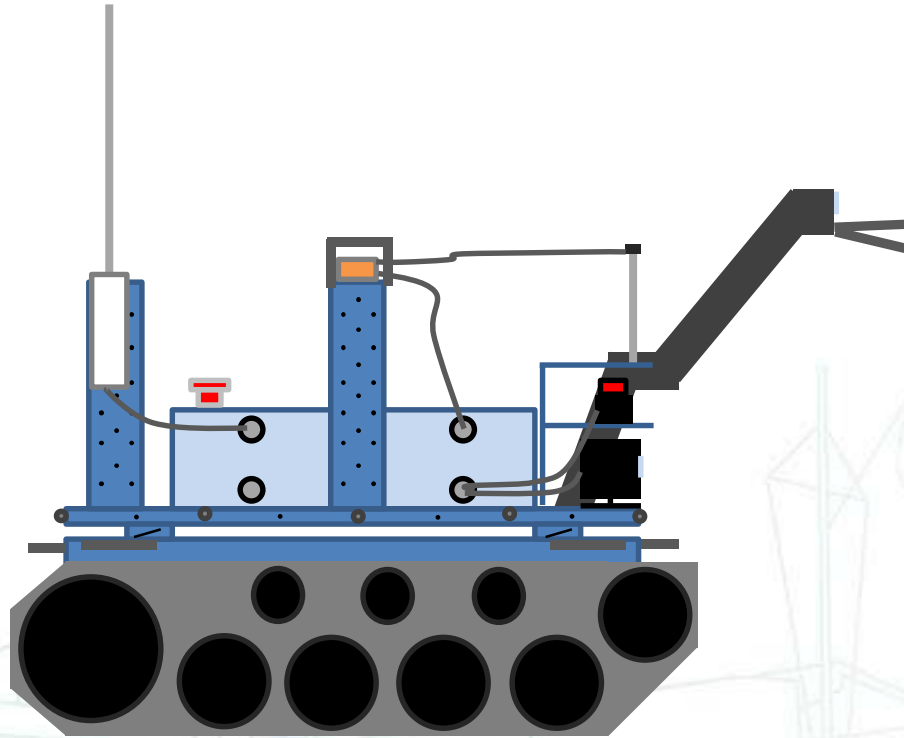


# AGV Buggy (2012)

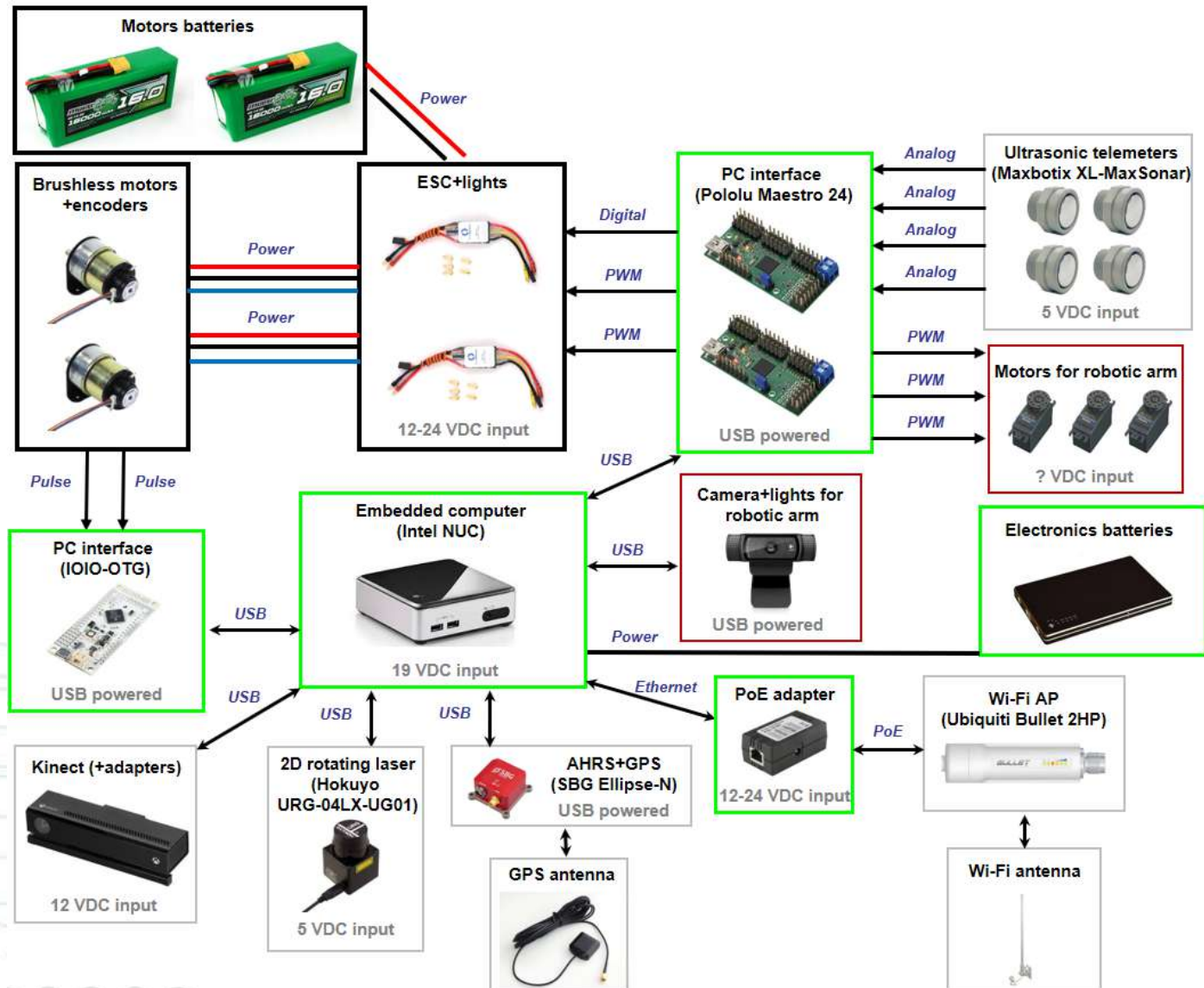
- USB
- PWM
- 5 V
- 12 V



# AGV euRathlon (futur)



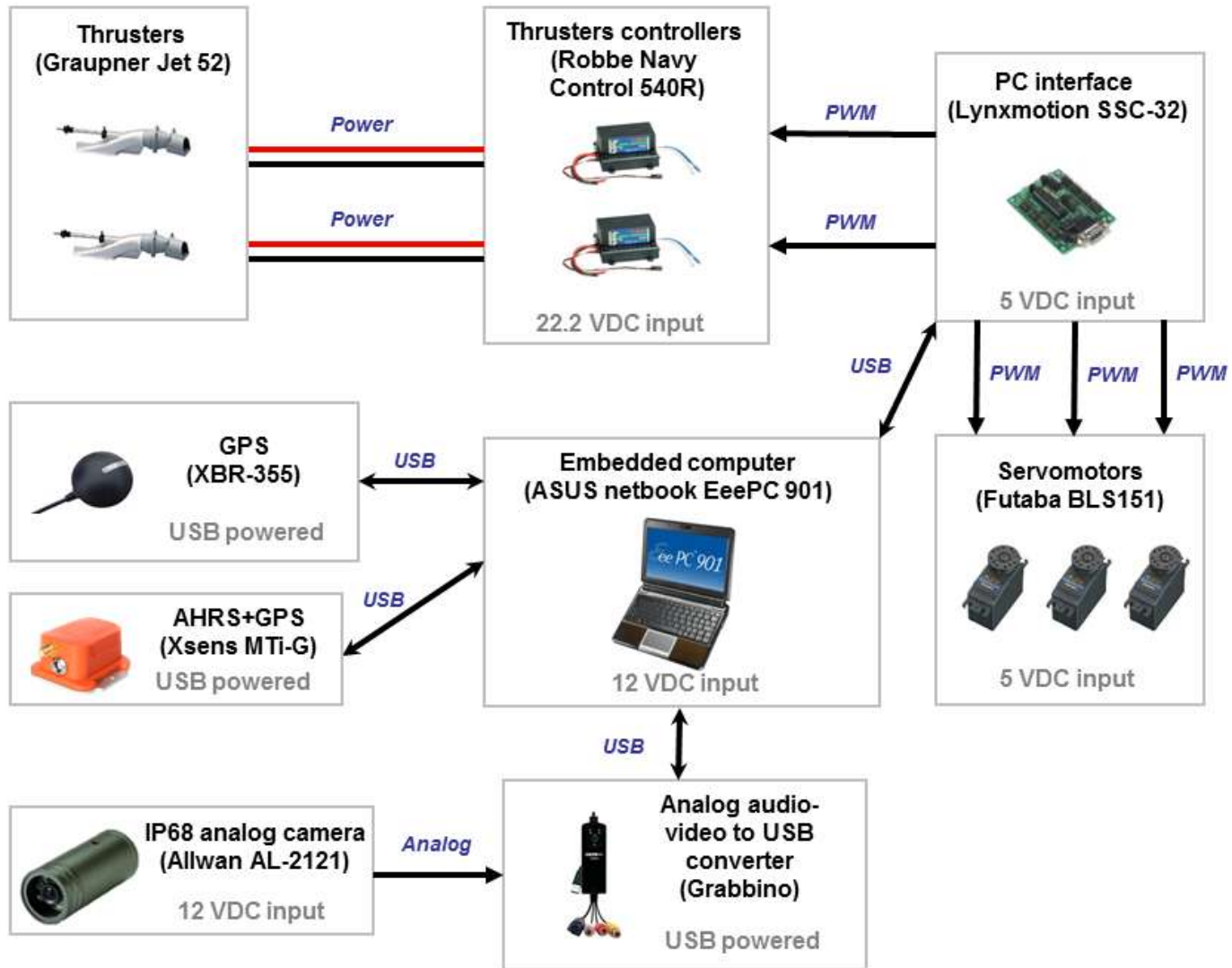
# AGV euRathlon (futur)



# ASV Motorboat (2013)



# ASV Motorboat (2013)

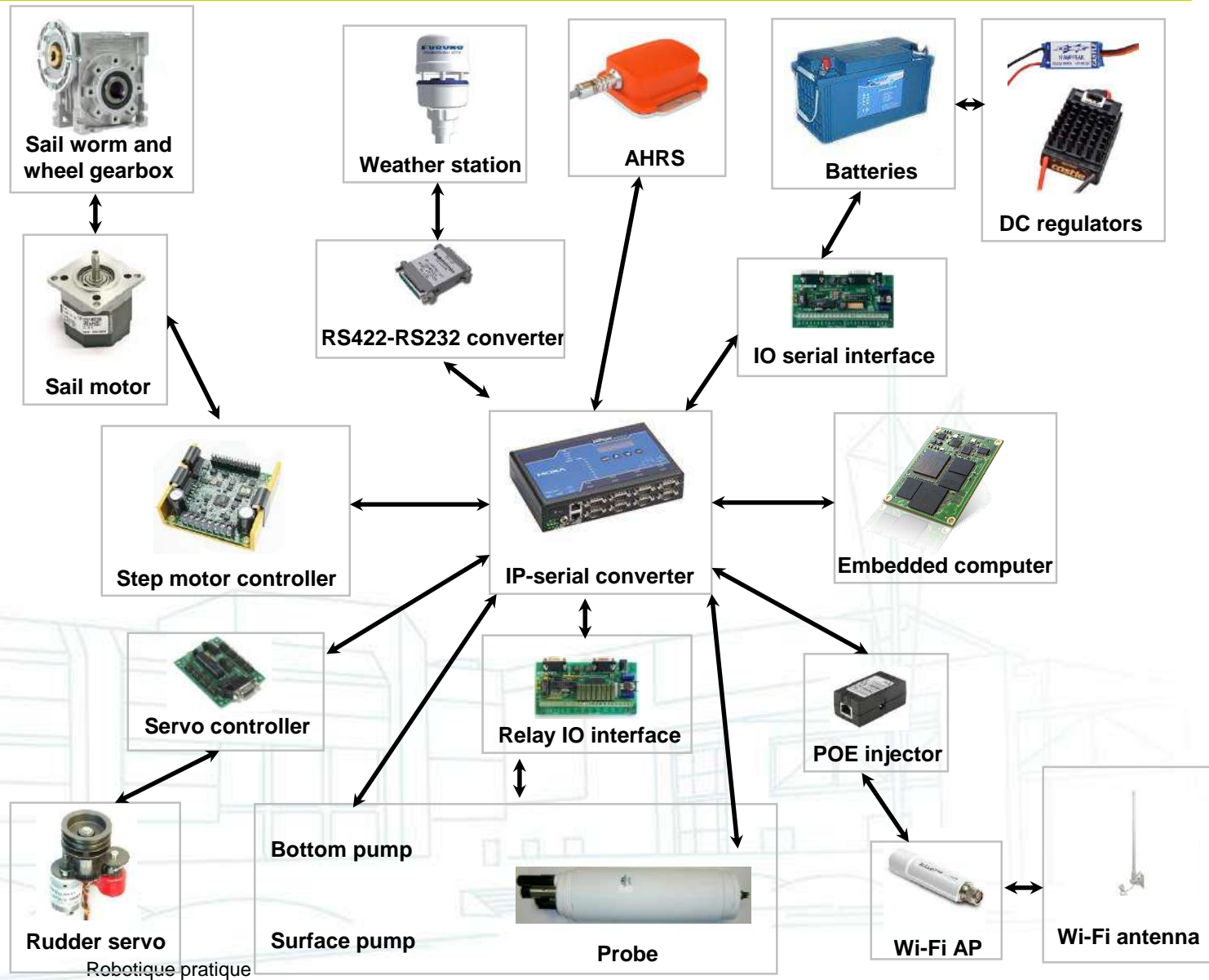




# ASV VAIMOS (2011)



# ASV VAIMOS (2011)

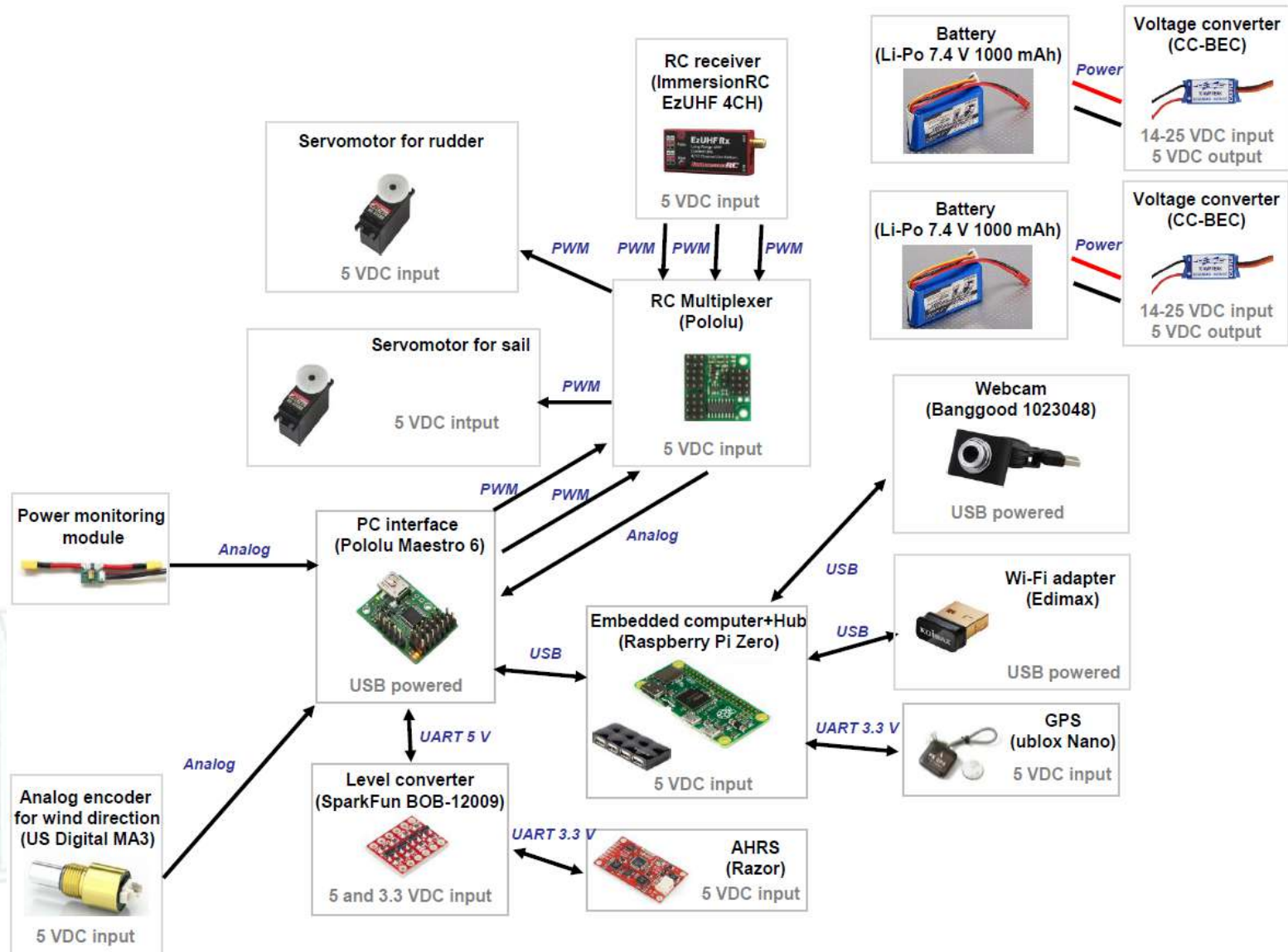




# ASV White Sailboat (2016)



# ASV White Sailboat (2016)

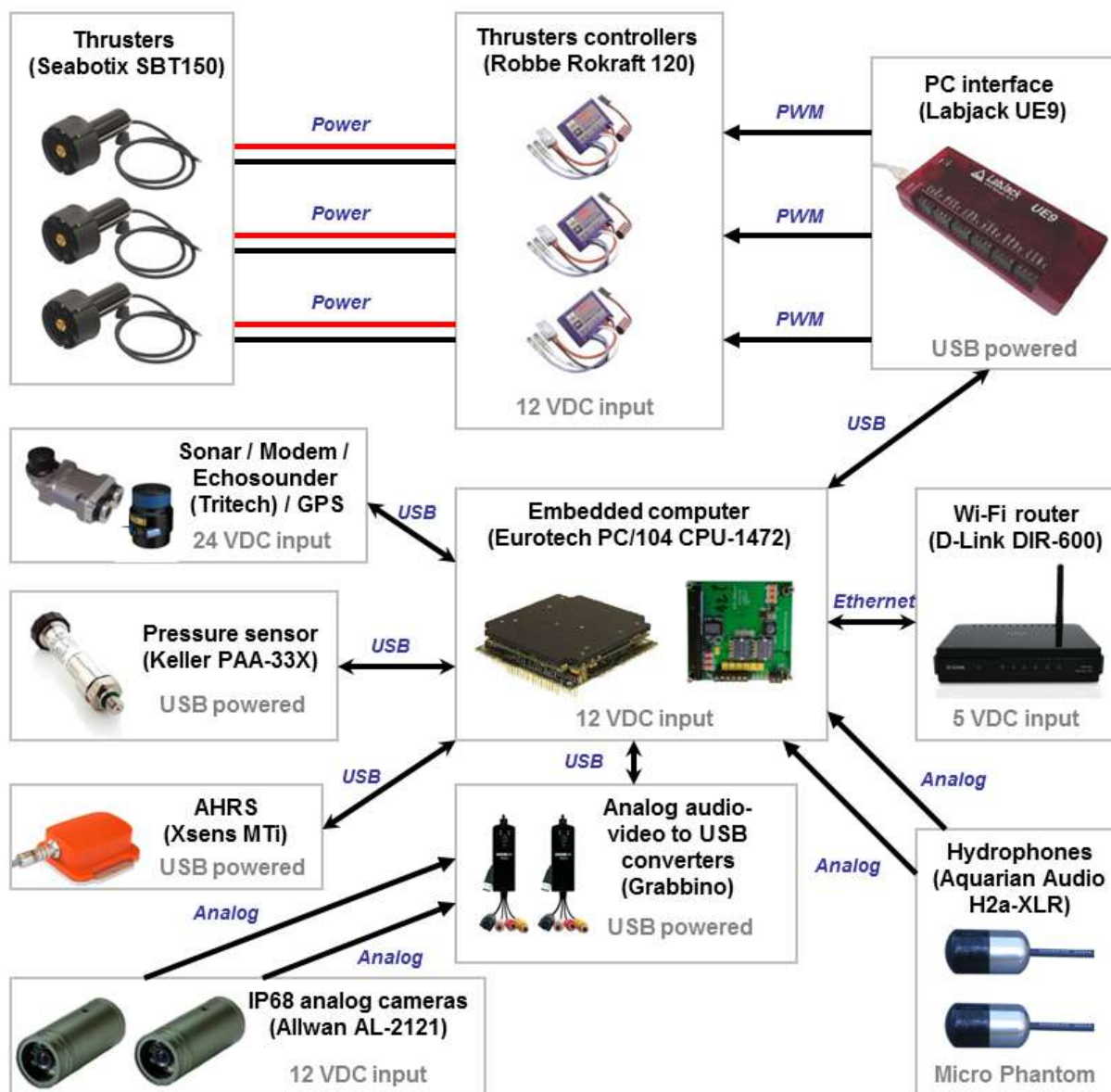




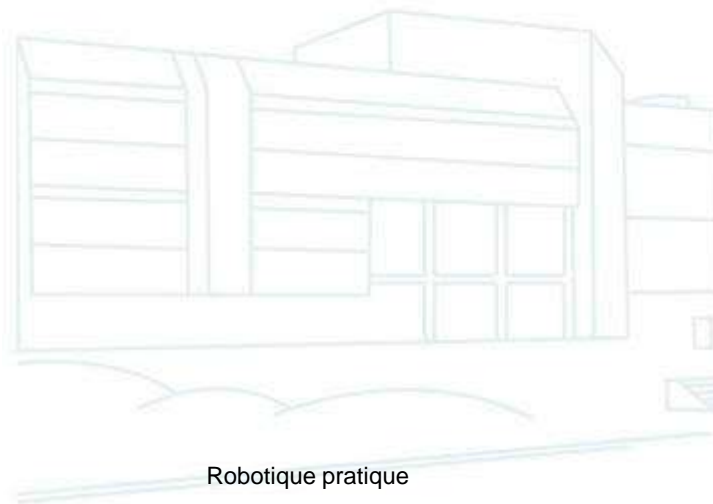
# AUV SAUC'ISSE (2014)



# AUV SAUC'ISSE (2014)

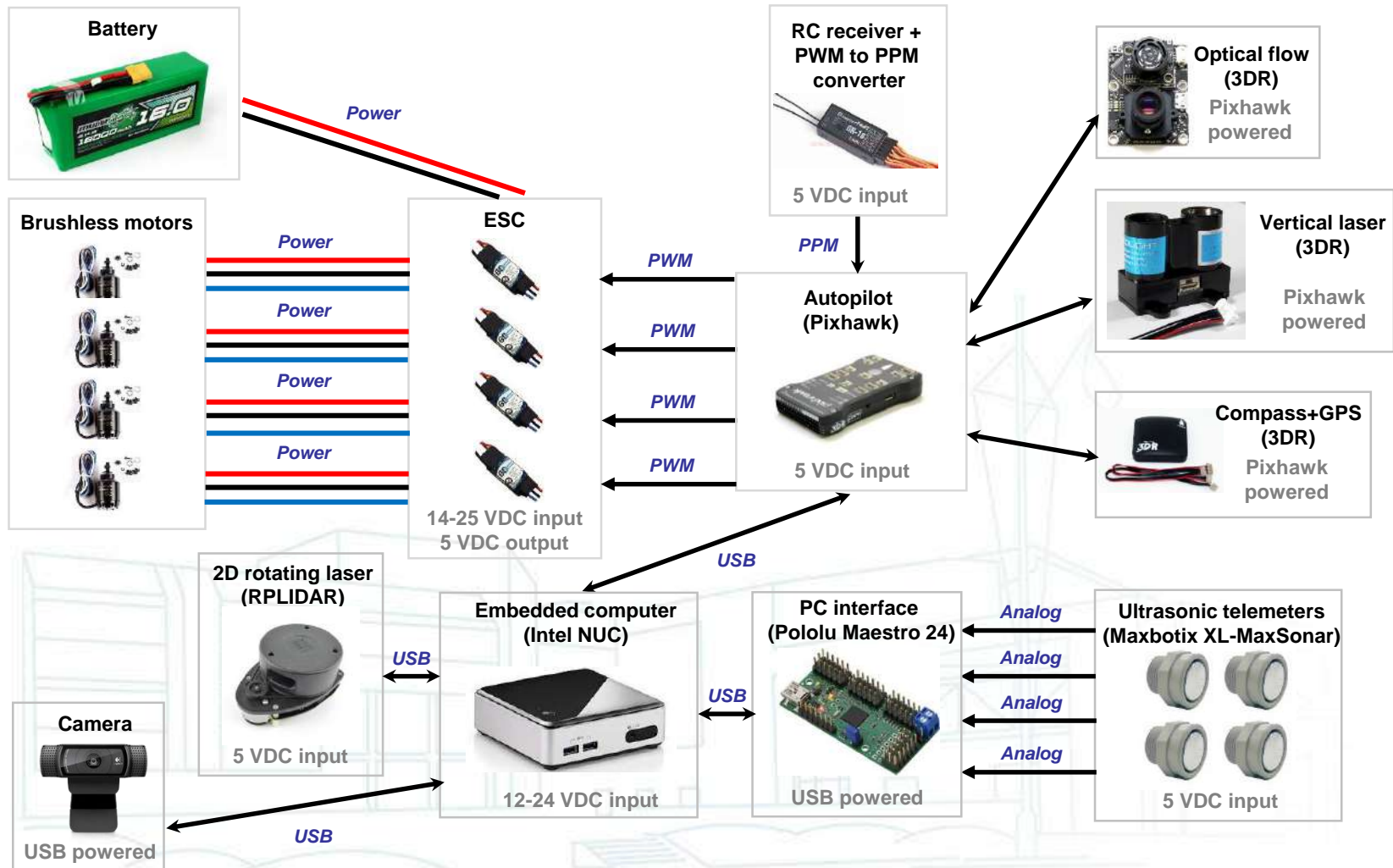


# Indoor Quadrotor euRathlon (futur)

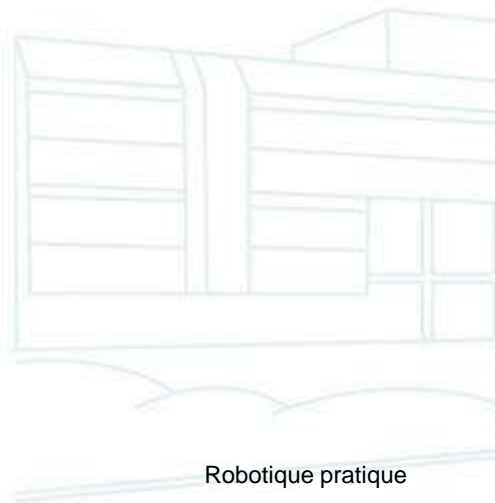
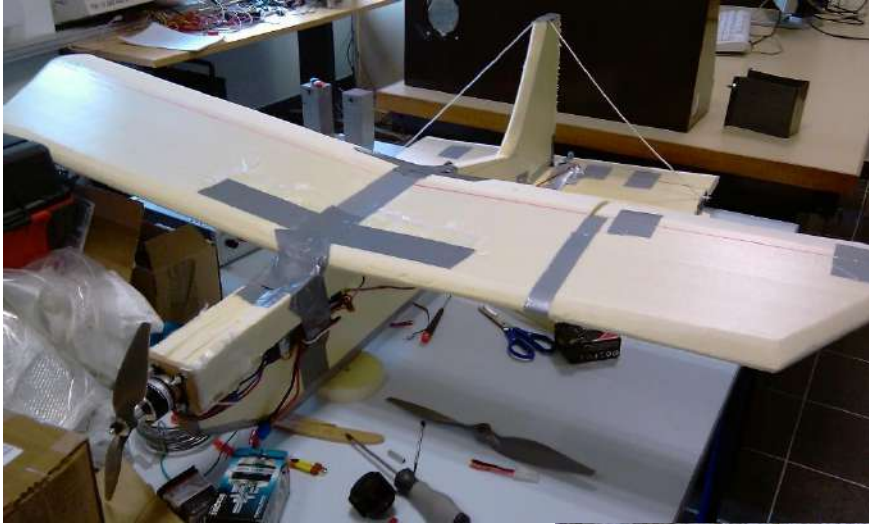




# Indoor Quadrotor euRathlon (futur)

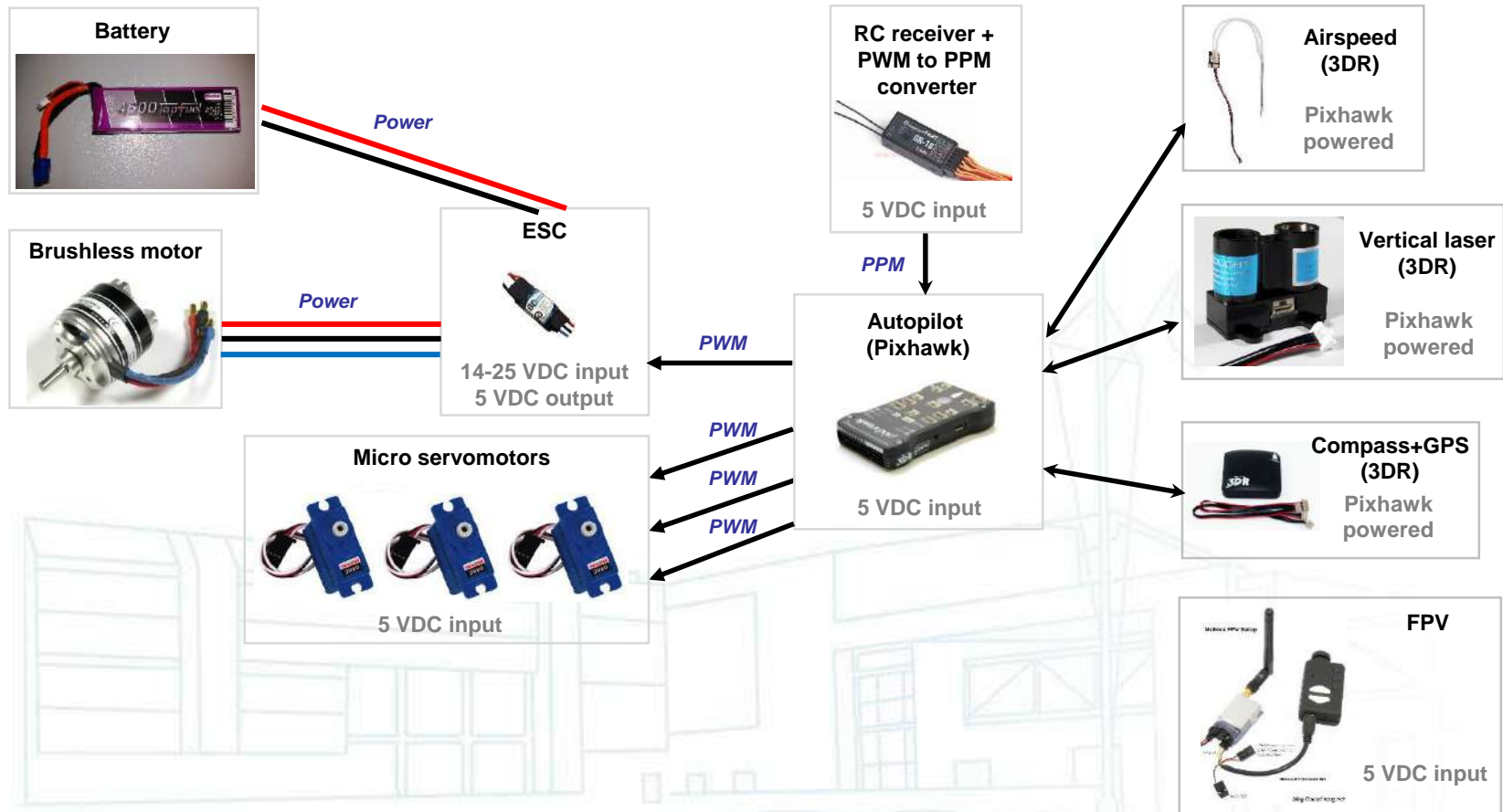


# Plane (futur)





# Plane (futur)

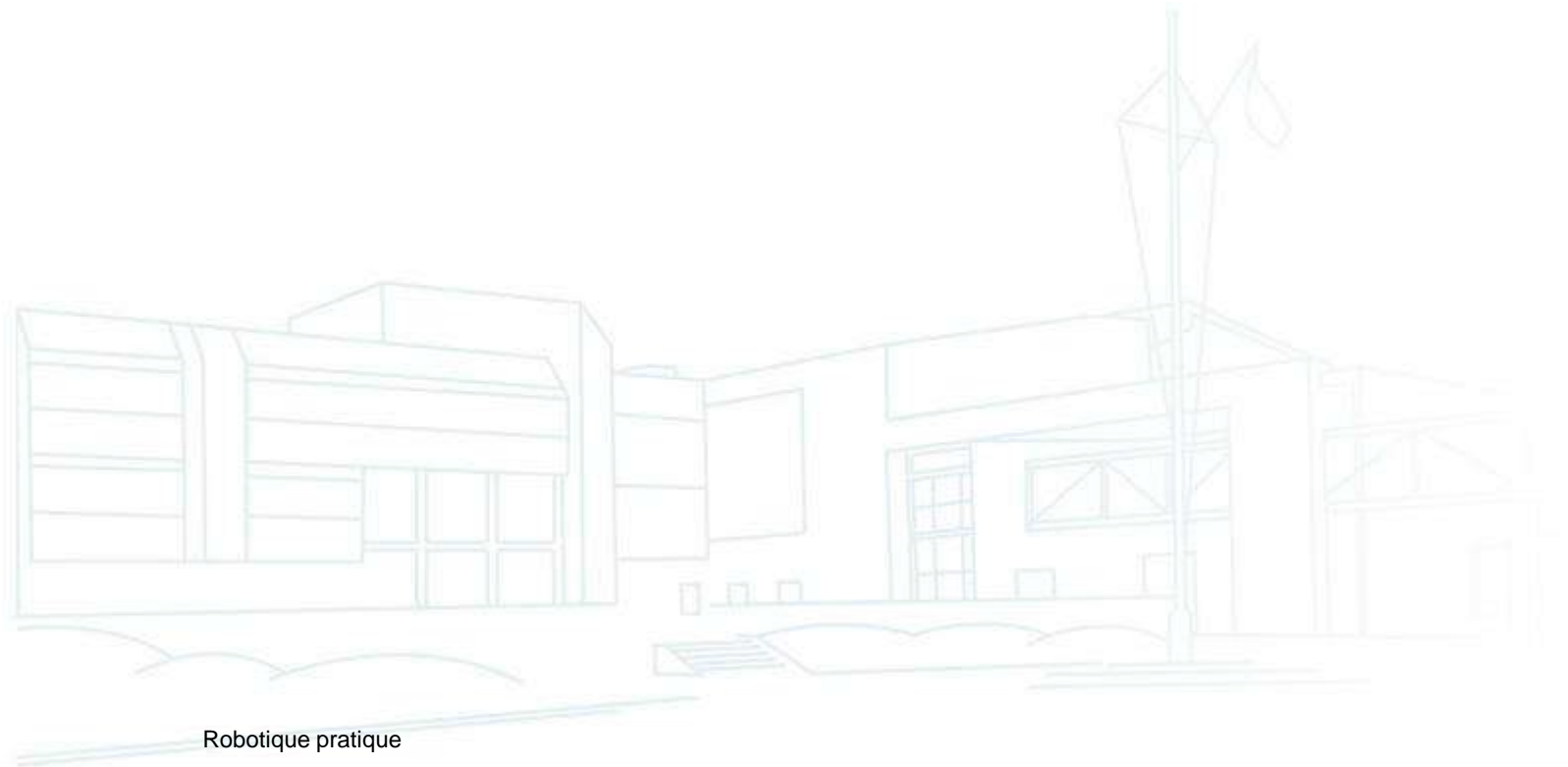




# Communiquer, contrôler à distance un système informatique autonome

# Communiquer, contrôler à distance un système informatique autonome

- But :
  - Savoir installer un PC et les périphériques réseaux nécessaires pour pouvoir le contrôler à distance lorsqu'il sera embarqué sur un robot



# Communiquer, contrôler à distance un système informatique autonome

- Montage d'un PC :
  - Un PC est en général constitué de :  
Carte mère : carte sur laquelle tout le reste est raccordé



Processeur (CPU)



Mémoire vive (RAM)



SSD, mémoire flash, disque dur



# Communiquer, contrôler à distance un système informatique autonome

## ■ Montage d'un PC :

- Un PC est en général constitué de :

Carte graphique (GPU) : souvent intégrée dans CPU ou carte mère



Carte audio : souvent intégrée dans la carte mère

Cartes réseaux Ethernet, Wi-Fi



Des dispositifs de refroidissement passifs ou actifs : sur CPU, GPU

Alimentation (PSU) : e.g. format ATX



Boîtier : contient tout et est muni de boutons (Power, Reset), LEDs (statut de l'alimentation, activité disques) et ports additionnels (USB, audio)

# Communiquer, contrôler à distance un système informatique autonome

## ■ Montage d'un PC :

- Bus et connecteurs internes classiques :

PCI Express : bus de données très rapide pour cartes d'extension (graphique, audio, Wi-Fi, LAN/Ethernet), remplace à la fois PCI et AGP)

SATA : disques durs, SSD, lecteurs DVD, remplace ATA/PATA/IDE

ExpressCard : bus de données très rapide pour cartes d'extension de PC portables, remplace PCMCIA

SMBus : capteurs de tension, courant, température

- Pour des vérifications de compatibilité diverses, voir <https://pcpartpicker.com/>





# Communiquer, contrôler à distance un système informatique autonome

## ■ Montage d'un PC :

- Bien rechercher des pièces de PC, robot, etc. :

Exemple : carte graphique puissante pour les simulations ou le traitement des données de certains capteurs (e.g. caméra stéréo ZED)

Recherche Google "carte graphique puissante" donne par exemple dans ses 1ers résultats "ASUS GeForce GTX 1070 - TURBO-GTX1070-8G" sur amazon.fr avec juste le prix mais sans description des caractéristiques

=> Rechercher sur le site du constructeur (NVIDIA) la documentation officielle et éventuellement comparer avec les autres de la gamme, rechercher à nouveau sur Google d'autres revendeurs et utiliser leurs fonctions de tri

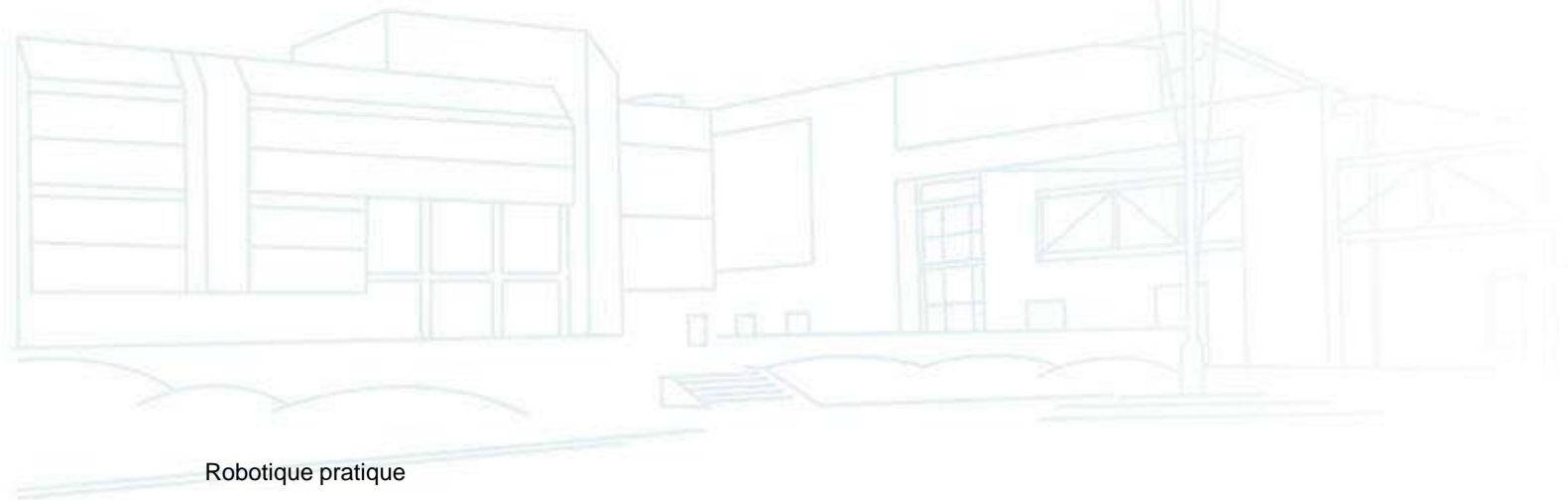


# Communiquer, contrôler à distance un système informatique autonome

## ■ Montage d'un PC :

- Bien rechercher des pièces de PC, robot, etc. :

Faire la différence entre le **constructeur du composant principal** (NVIDIA), le **modèle du composant principal** (GTX 1070), le **constructeur de la carte graphique complète** (ASUS), le **modèle de la carte graphique complète** (si plusieurs variantes existent) et le **distributeur/revendeur** (amazon.fr)



# Communiquer, contrôler à distance un système informatique autonome

## ■ Montage d'un PC :

- Bien rechercher des pièces de PC, robot, etc. :

Rechercher aussi sur Google des **tests** et **reviews** sur des sites spécialisés, parfois des **commentaires** par des acheteurs, certains revendeurs font aussi leurs propres tests indépendants (attention, dans tous les cas il est courant que les commentaires soient biaisés voire faux!)

Une fois qu'on a fait son choix, ne pas hésiter à faire une recherche Google du type "problème ASUS GeForce GTX 1070 - TURBO-GTX1070-8G" pour vérifier les **problèmes courants** rencontrés par les utilisateurs et évaluer à quel point il y a un risque d'être impacté

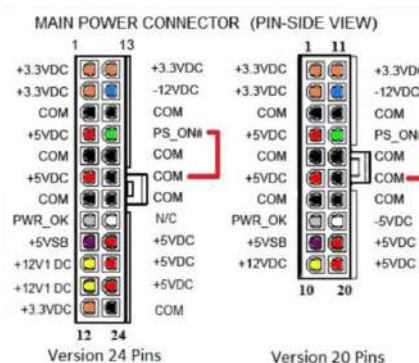


# Communiquer, contrôler à distance un système informatique autonome

## ■ Montage d'un PC :

- Tests unitaires matériels :

Une alimentation au format ATX peut être testée de cette manière



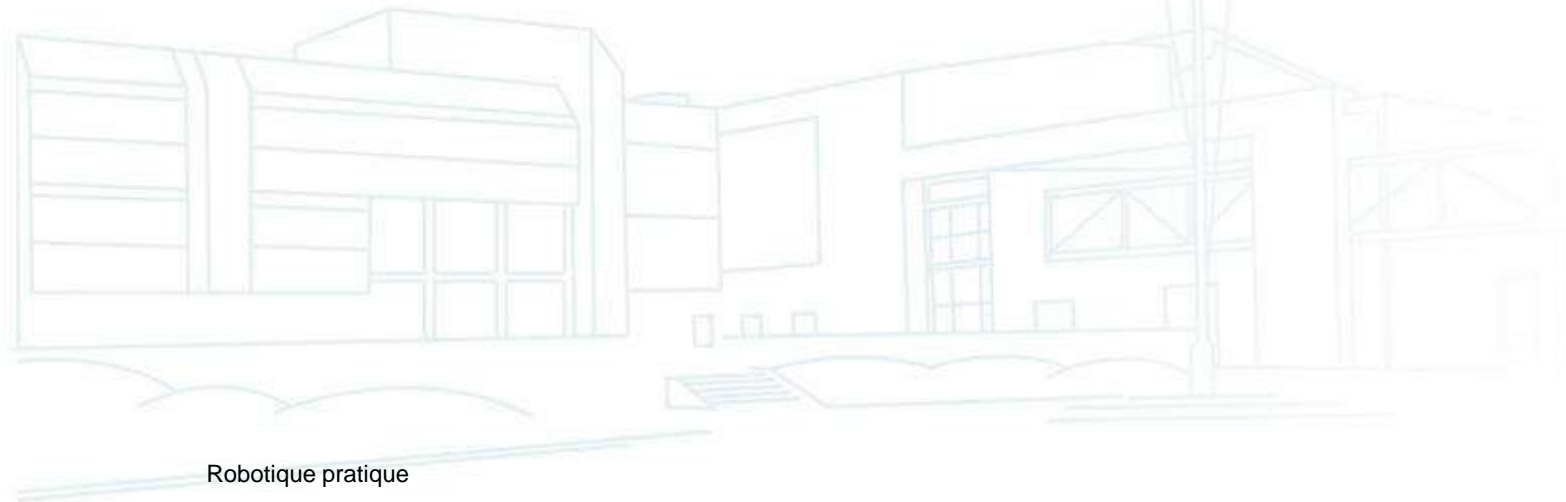
Carte mère : nécessite alimentation+bouton Power du boîtier (on peut remplacer l'effet du bouton Power en faisant un contact temporaire sur les pins correspondantes). Si on rajoute clavier, carte graphique (si non intégrée) et écran on peut souvent déjà commencer à interagir (limité)

# Communiquer, contrôler à distance un système informatique autonome

- Montage d'un PC :
  - Tests unitaires matériels :

**Les dispositifs de refroidissements adéquats sont indispensables dès qu'on rajoute le processeur : risque de surchauffe!**

Dès qu'on rajoute en plus la RAM, on peut en général commencer à configurer la majorité du BIOS/UEFI



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

### • BIOS/UEFI :

Utilitaire de configuration de la carte mère, intégré dedans

Si le BIOS/UEFI ne fonctionne plus (e.g. suite à de mauvais réglages ou à sa mise jour) :

Rester appuyé sur le bouton Power 10 s puis débrancher 5-10 min

Rallumer 5 fois (faire comprendre à la carte mère qu'elle a raté 5 démarrages de suite peut parfois la faire remettre ses paramètres par défaut)

Inverser 10 s le jumper CCMOS de la carte mère (hors tension)

Enlever/remplacer la pile Lithium de la carte mère (hors tension)

Consulter la documentation de la carte mère si aucune des solutions ne fonctionne (e.g. combinaison de touches spéciales, etc.)

SAV

**=> Etre prudent lors de la modification du BIOS/UEFI!**

# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- BIOS/UEFI :

Touches d'accès classiques :

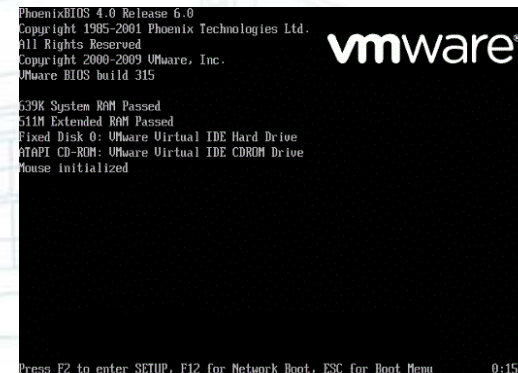
**DEL, F2 Enter Setup** : page principale de réglages

**F7 Update BIOS** : mise à jour à partir d'un fichier sur clé USB

**ESC, F8, F10, F11, F12 Boot Menu Override** : démarrage sur un périphérique temporaire

**F12 Network Boot**

**CTRL+I** : configurer un RAID (multiples disques durs en parallèle)



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

### • BIOS/UEFI :

Paramètres à vérifier :

**Load optimized settings** : souvent un bon point de départ

**Date, heure** : peut perturber l'OS si incorrect (mettre UTC), nécessite une carte mère avec pile ou supercapacité

**Halt on no error** : e.g. il ne faut pas que le PC du robot considère que l'absence d'un clavier est une erreur

**APM\AC loss auto restart, Power on if power failure** : sur un robot, on n'a pas toujours accès au bouton Power du PC, il faut que celui-ci démarre dès qu'il est alimenté



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- BIOS/UEFI :

Paramètres à vérifier :

**Boot order** : mettre en premier le périphérique contenant l'OS pour être sûr que si une clé USB est présente au démarrage, il ne cherche pas à démarrer dessus, de plus une touche permet en général de temporairement démarrer sur un autre périphérique si besoin

**WOL (Wake On Lan)** : peut éventuellement servir pour démarrer à distance

**Watchdog** : désactiver car souvent il va essayer de restaurer des paramètres par défaut lors de pertes d'alimentation brutales, ce qui arrive souvent sur un robot sur batteries, l'inconvénient est que si un jour on règle des paramètres qui font planter le BIOS, on ne pourra pas compter sur cette sécurité...

# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

### • BIOS/UEFI :

Paramètres à vérifier :

**AHCI/RAID/IDE, Compatible** : si SSD ou disque dur **SATA** et OS moderne, choisir AHCI, si l'OS n'arrive pas à accéder au disque, choisir IDE, Compatible (très légère perte de performance) ou consulter la documentation de la carte mère pour charger les bons drivers au cours de l'installation de l'OS (peut être compliqué)

**OS choice** : si l'option existe, un choix d'OS incohérent peut conduire à l'échec de l'installation ou des performances réduites (préréglages souvent liés aux paramètres d'alimentation, stockage, etc.)

**Certains changements** peuvent être **interdépendants** et nécessiter **plusieurs redémarrages** pour apparaître ou être pris en compte!

# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :



- Installation d'un système d'exploitation (OS) :

Penser à **activer tous les périphériques** potentiellement utiles **avant l'installation** (dans le BIOS, via interrupteurs, etc.)

Un CD, DVD ou clé USB doit être préparé de manière spéciale pour que le PC puisse démarrer dessus et lancer l'installation

Si **PC 32 bit**, seul un **OS 32 bit** est possible, si **PC 64 bit**, on a le **choix** en général (certains réglages dans le BIOS peuvent être nécessaire selon le choix)

Il vaut mieux utiliser la fonction **Boot Menu Override** du BIOS (menu temporaire de démarrage) pour démarrer temporairement sur un périphérique, de cette manière aux prochains redémarrages il ne cherchera pas à relancer le début de l'installation

# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Installation d'un système d'exploitation (OS) :

## **Boot Menu Override :**

Les périphériques peuvent parfois être **listés par catégories** et un **menu séparé** peut exister pour choisir l'ordre selon la catégorie e.g. certaines clé USB peuvent être listées dans la catégories Hard disk aux cotés du disque dur interne, il faut donc penser à changer aussi l'ordre de démarrage dans cette catégorie si nécessaire...

Les noms de catégories ou classements peuvent parfois porter à confusion, bien vérifier si on ne trouve pas le périphérique souhaité...



# Communiquer, contrôler à distance un système informatique autonome

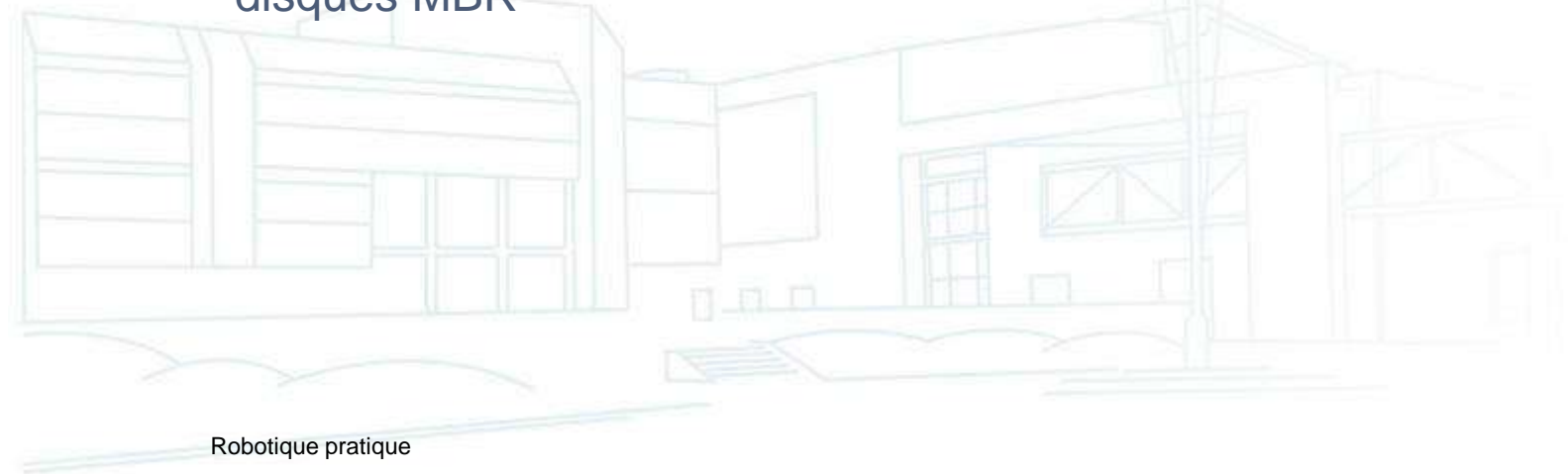
## ■ Installation d'un PC :

- Installation d'un système d'exploitation (OS) :

### **UEFI boot (GPT) vs Legacy boot (MBR) :**

Les PC munis de **BIOS** (avant 2010, en général 32 bit) démarrent sur des disques durs ou SSD avec un **MBR** (contient un peu de code et la table des partitions, avec des limitations de taille)

Les PC munis d'**UEFI** (après 2010) démarrent différemment (**GPT**) et sont notamment compatibles avec des **disques > 2 To** et peuvent aussi proposer de se comporter comme un BIOS avec disques MBR



# Communiquer, contrôler à distance un système informatique autonome

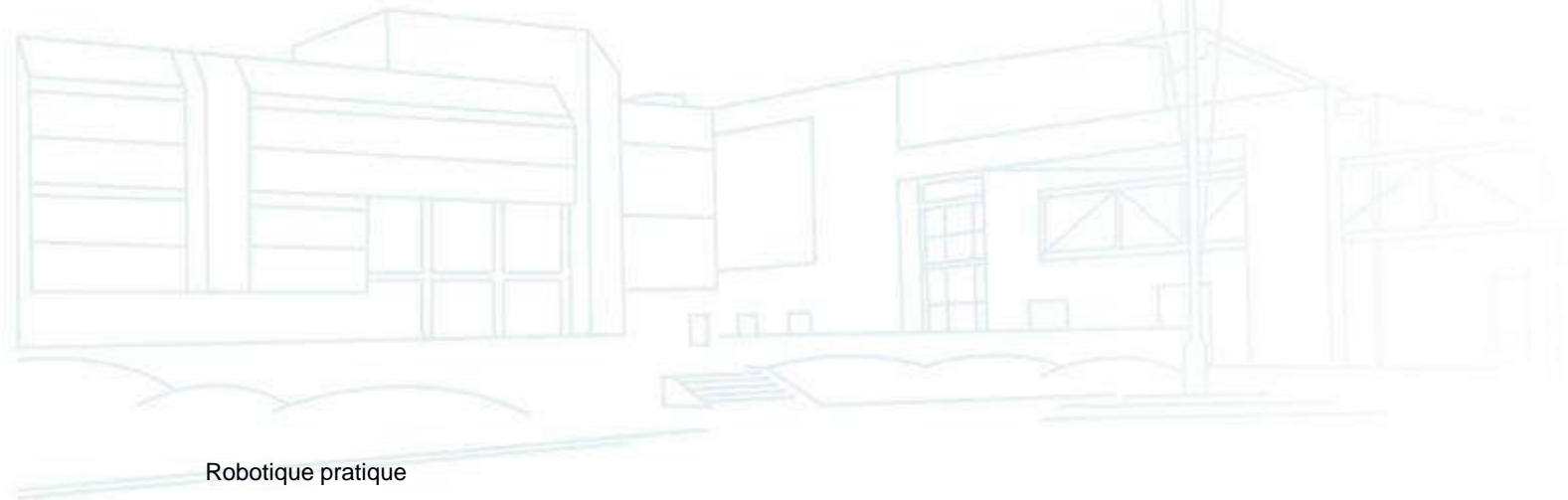
## ■ Installation d'un PC :

- Installation d'un système d'exploitation (OS) :

### **UEFI boot (GPT) vs Legacy boot (MBR) :**

**BIOS/MBR** : Permet de définir 4 partitions sur un disque dur de max 2 To. Une partition peut être « principale » (normale) ou « étendue » (contient des « lecteurs logiques » qui vont se comporter comme des partitions principales, l'idée étant de surmonter la limite des 4 partitions qu'on peut définir dans le MBR).

**UEFI/GPT** : Jusqu'à 128 partitions de 18 millions de To.



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Installation d'un système d'exploitation (OS) :

### **UEFI boot (GPT) vs Legacy boot (MBR) :**

Le choix de l'un ou l'autre des modes a en général peu d'importance pour un robot (laisser par défaut), sauf s'il a un disque dur ou SSD > 2 To, ou que le ou les OS à installer ne supporteraient pas l'un ou l'autre des modes

Attention, cela a une influence sur l'organisation des partitions...

=>

Recommandations pour Windows :

UEFI/GPT : [https://msdn.microsoft.com/fr-fr/library/windows/hardware/dn898510\(v=vs.85\).aspx](https://msdn.microsoft.com/fr-fr/library/windows/hardware/dn898510(v=vs.85).aspx)

BIOS/MBR : [https://msdn.microsoft.com/fr-fr/library/windows/hardware/dn898504\(v=vs.85\).aspx](https://msdn.microsoft.com/fr-fr/library/windows/hardware/dn898504(v=vs.85).aspx)



# Communiquer, contrôler à distance un système informatique autonome

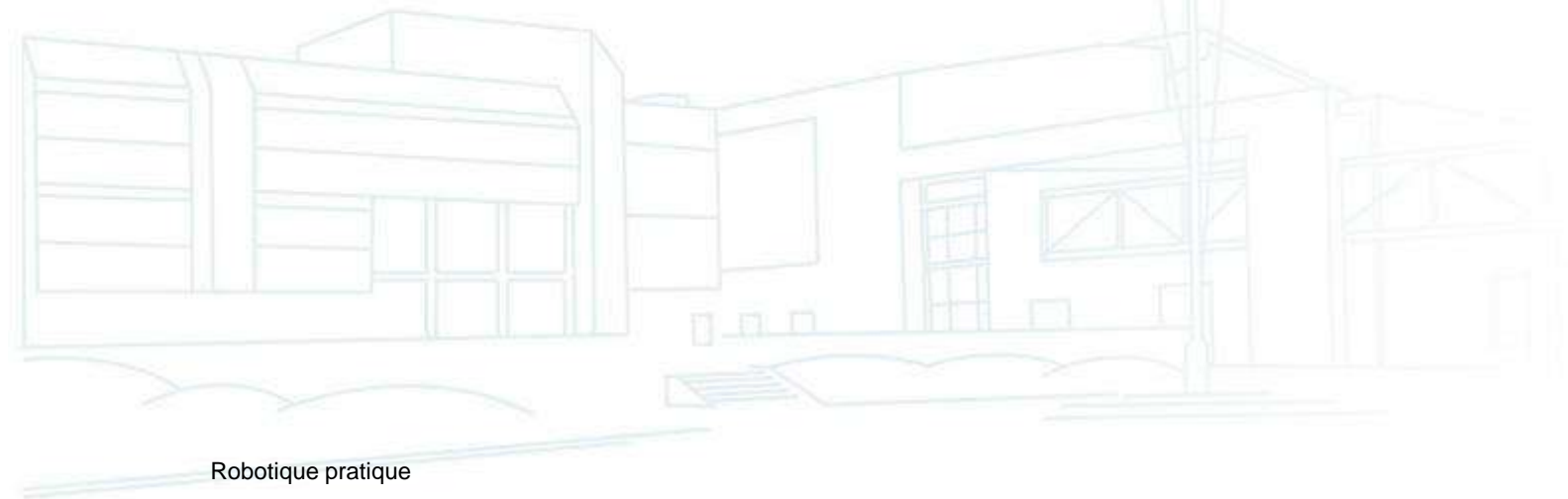
## ■ Installation d'un PC :

- Installation d'un système d'exploitation (OS) :

### **UEFI boot (GPT) vs Legacy boot (MBR) :**

Certains PC peuvent avoir un **mélange inhabituel d'UEFI et processeurs 32 bit ou 64 bit** (e.g. ordinateurs à base d'Intel Atom), dans ce cas certains OS nécessitent une **installation spéciale...**

Voir <http://liliputing.com/2014/10/run-ubuntu-zotac-zbox-pico-mini-pc-kind.html>



# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Systèmes de fichiers :

Un disque dur contient en général plusieurs partitions, chaque partition pouvant contenir un système de fichier spécifique

Un **système de fichiers** définit une **organisation de dossiers, fichiers, droits d'accès** sur une partition

**Image de disque** : fichier correspondant à une copie bit par bit d'un disque (disque dur, SSD, DVD, etc.)

**Montage de partition** : attribution d'un chemin d'accès permettant de visionner les fichiers et dossiers d'une partition

**Racine d'une partition** : premier niveau de l'arborescence de dossiers sur une partition

# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Systèmes de fichiers :

Manipulations sur des partitions déjà créées et remplies de données :

Il est **très difficile de modifier** l'adresse de **début d'une partition** (en effet, cela implique de déplacer toutes ces données et il faut a priori avoir de l'espace en plus disponible pour pouvoir faire les manipulations nécessaires)

**On peut réduire la taille d'une partition** si elle n'est pas remplie mais ceci prend un temps dépendant des données qui se trouvaient physiquement en fin de partition car elles seront à déplacer

**On ne peut** en général **pas changer le système de fichier** d'une partition sans effacer ses données

Cependant, il est en général **simple et immédiat d'agrandir une partition** s'il y a de l'espace non partitionné derrière

# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Systèmes de fichiers :

Windows :

**NTFS** : le meilleur choix si on n'utilise que Windows, fonctionne aussi de mieux en mieux sous Linux

**exFAT** : peu utilisé, mise à jour du FAT32

**FAT32** : la meilleure portabilité, mais limite de 4 Go pour les fichiers et pas de gestion de droits d'accès

**FAT (FAT16)** : bonne portabilité avec certains appareils électroniques (e.g. carte micro SD sur Arduino) car simple à gérer si on doit écrire soi-même un driver sur microcontrôleur

Linux :

**ext4** : choix par défaut pour les Linux modernes, pas accessible sous Windows

**ext2-3** : bon choix si besoin de partage avec Windows car des drivers tiers existent (vérifier les limitations : <http://www.fs-driver.org/> )

**vfat** : FAT32, on ne peut pas installer Linux dessus mais bon choix pour des partages de données avec Windows

**swap** : spécifique mémoire virtuelle, on peut en général s'en passer si on a assez de RAM

# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Démarrage d'un système d'exploitation (OS) :

Etapes :

Mise sous tension du PC

Le **BIOS/UEFI** démarre, fait diverses vérifications puis examine le début des disques durs, SSD, clés USB présents dans l'ordre configuré et **lance le premier MBR/GPT valide trouvé**

Le MBR/GPT contient un logiciel très court (512-8192 octets) pouvant commencer à proposer des choix à l'utilisateur, ou exécuter le reste du logiciel sur le début d'une autre partition (**VBR**, sur la **partition active** du disque si MBR)



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Démarrage d'un système d'exploitation (OS) :

Etapes :

Des fichiers vont commencer à être exécutés sur la partition et le **boot loader** (son code est réparti entre le début du disque (MBR), le début d'une partition (VBR), et des fichiers sur cette partition) va alors pouvoir se charger complètement et commencer à rechercher le système d'exploitation ou proposer des choix

Les fichiers du **système d'exploitation** choisi (éventuellement sur une autre partition ou disque) vont alors pouvoir commencer à lancer son **noyau**, charger des **drivers** lancer des **services** et afficher le **bureau** et les **applications**



# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Démarrage d'un système d'exploitation (OS) :  
Boot loaders :
    - Linux : GRUB, LILO
    - Windows XP : NTLDR
    - Windows Vista, 7, 8, 10 : BOOTMGR

Toutes les parties des boot loaders ne sont pas visibles dans les explorateurs d'ordinateur car certaines parties ne sont pas des fichiers (e.g. MBR, VBR), il faut utiliser des outils de lecture bas niveau (e.g. **dd**) pour accéder au code et éventuellement le sauvegarder sous forme de fichiers

# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Démarrage d'un système d'exploitation (OS) :

Des commandes ou fichiers de configuration spécifiques permettent de régler les options

**/boot/grub/menu.lst** ou **/etc/default/grub** et **update-grub**, **efibootmgr**, **grub-customizer**, **boot-repair** pour **GRUB**

**bcdedit**, **bcdboot**, **bootsect**, **EasyBCD** pour **BOOTMGR**

Il est parfois possible de lancer les boot loaders les uns après les autres

Plus d'infos : [http://www.ensta-bretagne.fr/lebars/tutorials/TD\\_wifi\\_rtk\\_multiboot\\_rpi.pdf](http://www.ensta-bretagne.fr/lebars/tutorials/TD_wifi_rtk_multiboot_rpi.pdf)

```
default N
timeout sec
color couleur1 couleur2

# la configuration pour l'OS dont le Grub est installé

title Le libellé d'OS
root (hd<disque>,<partition>)
kernel /boot/vmlinuz-2.x.x.xx root=/dev/hdLN options
initrd /boot/initrd.img-2.x.x.xx

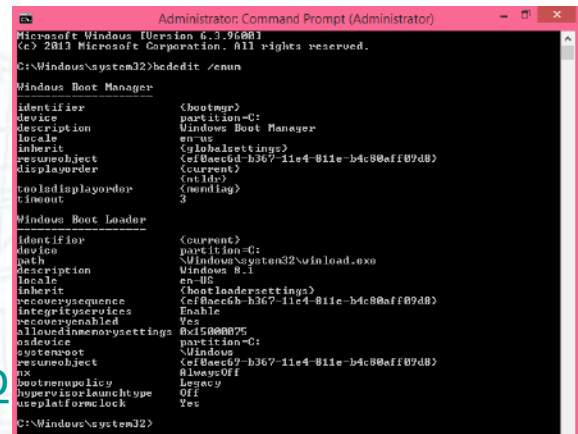
# à partir d'ici à éditer pour les autres OS

# Pour la grande famille GNU/Linux
# pour chaque OS à ajouter dans Grub il faut écrire le bloc suivant

title Le libellé d'OS supplémentaires
root (hd<disque>,<partition>)
kernel /boot/vmlinuz-2.x.x.xx root=/dev/hdLN options
initrd /boot/initrd.img-2.x.x.xx
rootnoverify (hd<disque>,<partition>)

# Pour la famille Windows

title Le libellé d'OS (à mettre ce que tu veux)
root (hd<disque>,<partition>)
chainloader +1
```



```
Microsoft Windows [Version 6.0.6002]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32\bcdedit /enum

Windows Boot Manager
identifier {bootmgr}
device partition=C:
description Windows Boot Manager
locale en-US
inherit {efib6c9-b367-11e4-811e-b4c80aff09d8}
displayorder {current}
valid {all}
tooldisplayorder {bootmgr}
timeout 3

Windows Boot Loader
identifier {current}
device partition=C:
path \Windows\system32\winload.exe
description Windows 8.1
locale en-US
inherit {efib6c9-b367-11e4-811e-b4c80aff09d8}
recoverysequence {efib6c9-b367-11e4-811e-b4c80aff09d8}
integrityservices Enable
recoveryenabled Yes
alleudminmenosettings 0x15000075
osdevice partition=C:
systemroot \Windows
resumeobject {efib6c9-b367-11e4-811e-b4c80aff09d8}
bootmanupolicy Windows
hypervisorlaunchtype Off
securebootplatformlock Yes

C:\Windows\system32>
```

# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Démarrage d'un système d'exploitation (OS) :

=> Lorsqu'un PC ne démarre pas correctement, il faut essayer d'identifier à quelle étape les problèmes commencent pour savoir comment les résoudre!

=> Plus d'infos : <https://neosmart.net/wiki/mbr-boot-process/> ,  
<https://neosmart.net/wiki/easybcd/uefi/>



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

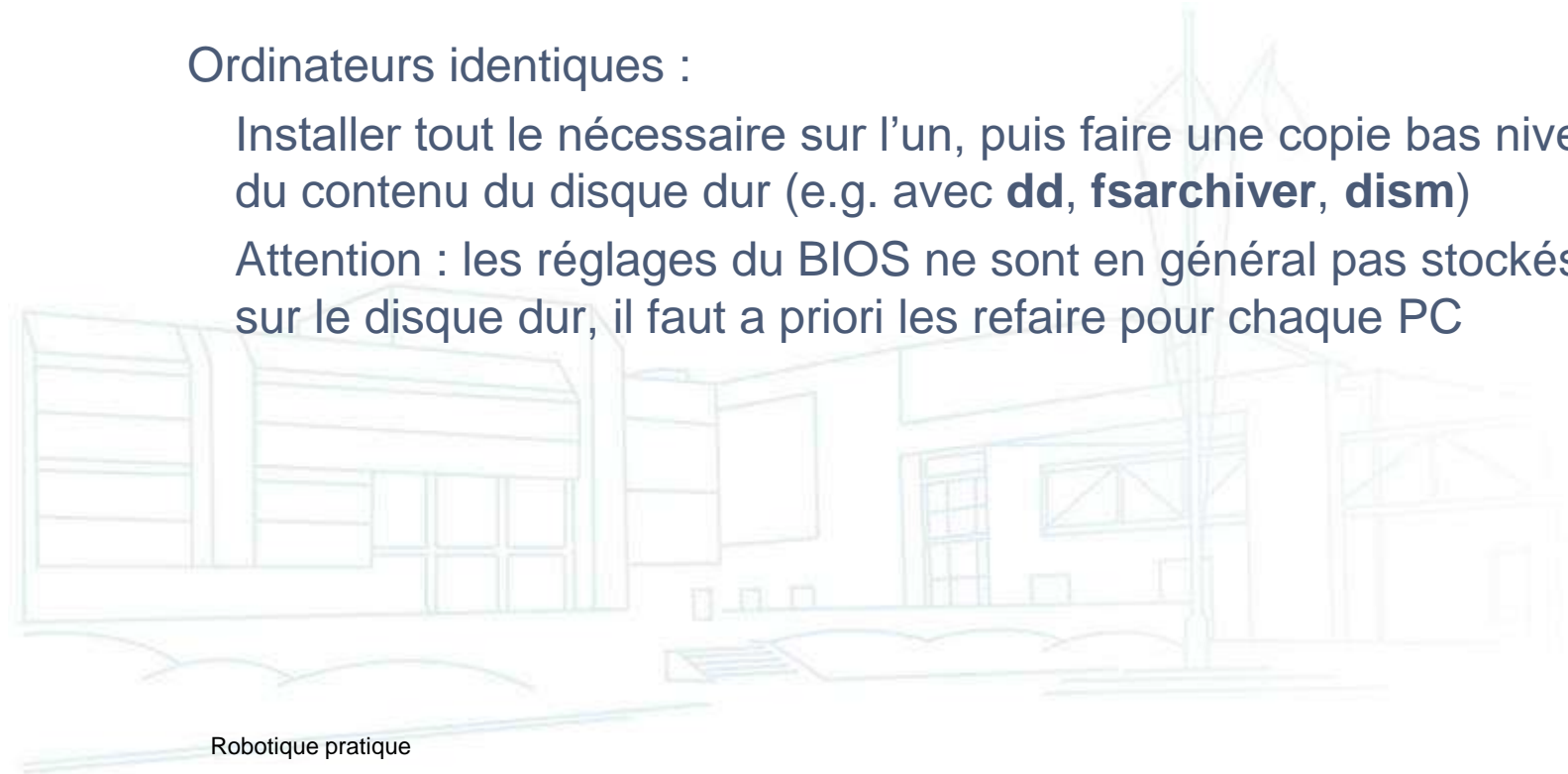
- Automatisation de l'installation d'OS, installations en masse :

Pour une meute de robots, il se peut qu'on ait plusieurs ordinateurs à installer

Ordinateurs identiques :

Installer tout le nécessaire sur l'un, puis faire une copie bas niveau du contenu du disque dur (e.g. avec **dd**, **fsarchiver**, **dism**)

Attention : les réglages du BIOS ne sont en général pas stockés sur le disque dur, il faut a priori les refaire pour chaque PC



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Automatisation de l'installation d'OS, installations en masse :

Ordinateurs hétérogènes :

Une installation automatisée de l'OS et de la plupart des applications peut être préparée et lancée en parallèle sur chaque PC

**Windows** : fichier **autounattend.xml**, voir **Windows PE**, Windows ADK, Windows AIK et documentation des installeurs d'applications classiques : **Windows Installer**, **Inno Setup**, **InstallShield**, Nullsoft **NSIS**, etc. En dernier recours voir **Autolt** pour **simuler clics de souris**. **Mots-clés** : **unattended**, **silent**, **quiet**, **passive** installations.

**Ubuntu** : fichier **preseed.cfg**.

Les version modernes de Windows et Ubuntu peuvent dans certaines conditions aussi s'adapter lors du premier démarrage si on a essayé d'utiliser la méthode de la slide précédente, mais ce n'est pas vraiment supporté...

# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Automatisation de l'installation d'OS, installations en masse :

Ordinateurs hétérogènes :

Ubuntu garde dans **/var/cache/apt/archives** les paquets .deb téléchargés, ceci peut être utile quand on veut installer les mêmes paquets sur d'autres ordinateurs ayant une connexion Internet limitée...

Utiliser **sudo apt-get clean** et **sudo apt-get autoremove** si besoin de les supprimer et économiser de l'espace disque





# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Lancement automatique de programmes au démarrage de l'OS

Linux :

**cron**

Fichiers dans **/etc/init.d**

Configurer l'**auto login** et fichier **~/.bashrc** ou similaire

Windows :

Task Scheduler

Création d'un service, e.g. avec l'outil **srvany.exe**

Configurer l'**auto login** et

**C:\ProgramData\Microsoft\Windows\Start  
Menu\Programs\StartUp**

# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Démarrage/arrêt/configuration de services

Linux :

Redémarrage du réseau :

**sudo /etc/init.d/networking restart**

ou

**sudo service network-manager restart**

Windows :

Arrêt du service de mise à jour :

**sc stop wuauserv**

Désactivation du service de mise à jour (attention : peut perturber l'installation de certaines applications) :

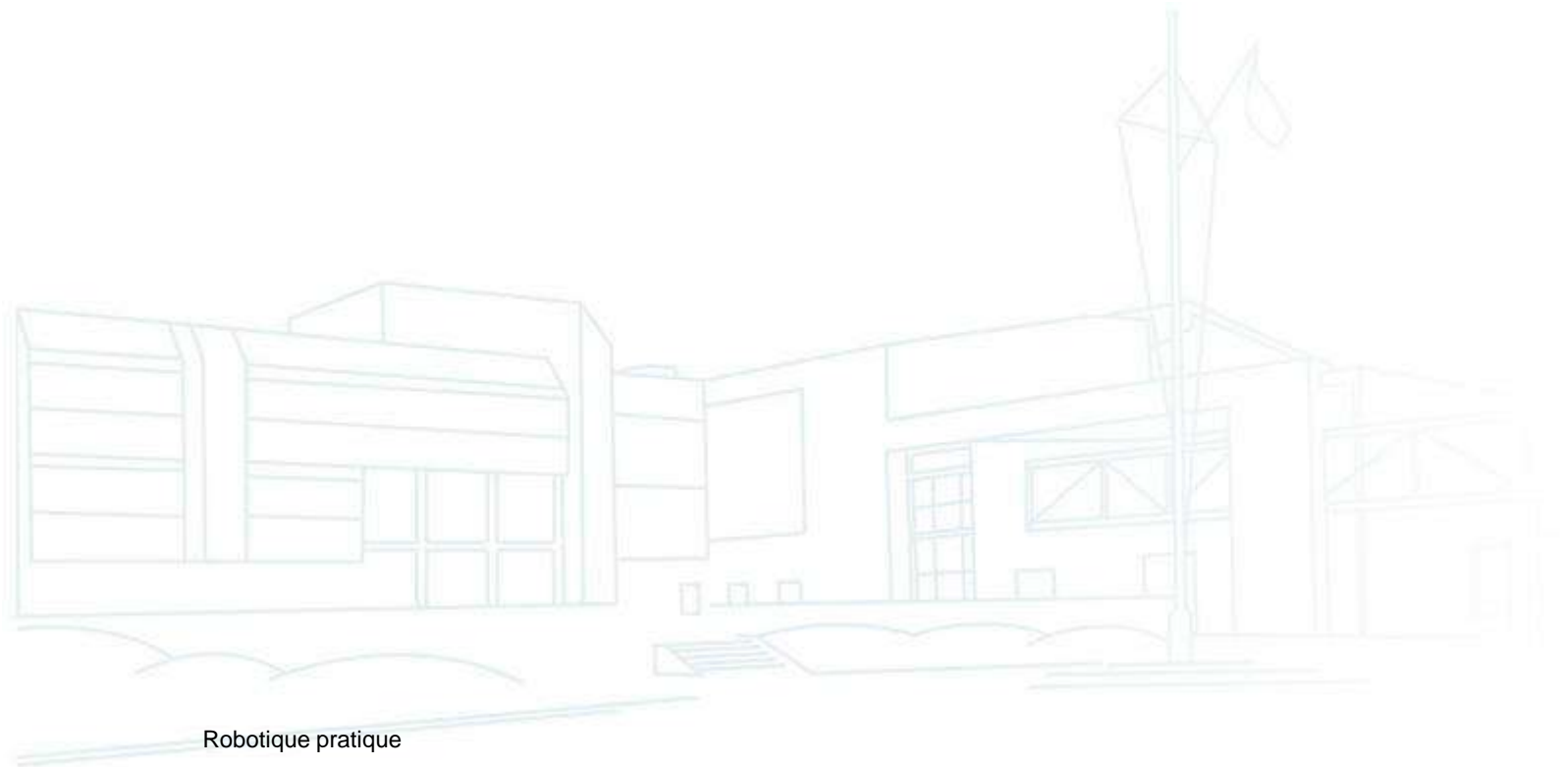
**sc config wuauserv start= disabled**

# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :

- Gestion des mises à jour

Désactiver le téléchargement et installation automatique sur un système autonome...



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Configuration du réseau pour l'accès à distance

Prérequis/Rappels :

**Interface réseau** : **carte Wi-Fi** (réseau sans fil), **carte Ethernet** (réseau avec câbles à prises RJ45)

**Adresse MAC** : identifiant unique au monde d'une interface réseau  
e.g. 00-AB-2D-F9-28-CC

**Adresse IP** : identifiant d'une interface réseau sur un réseau IP, unique sur ce réseau IP particulier

e.g. pour IPv4 : 172.20.25.154

**Masque de sous-réseau** : combiné avec l'adresse IP, permet de savoir combien de périphériques peuvent être mis sur ce réseau IP

e.g. pour IPv4 : 255.255.0.0

# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Configuration du réseau pour l'accès à distance

Prérequis/Rappels :

**DHCP** : protocole de configuration automatique d'adresse IP

**Serveur DHCP** : périphérique fournissant des adresses IP aux périphériques compatibles

Adresse IP **statique** vs **dynamique** : statique implique de spécifier soi-même au moins l'adresse IP et le masque de sous-réseau, dynamique nécessite qu'un serveur DHCP soit présent sur le réseau pour les configurer automatiquement



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Configuration du réseau pour l'accès à distance

Prérequis/Rappels :

**SSID** : nom d'un réseau Wi-Fi

**AP** (Access Point/Point d'accès) : périphérique générant un réseau Wi-Fi en mode infrastructure

**AP client** : périphérique se connectant à un réseau Wi-Fi généré par un AP

**Wi-Fi ad-hoc vs infrastructure : le mode infrastructure est le plus courant**, le mode ad-hoc n'a pas de concept d'AP avec plusieurs AP client connectés dessus, tous les périphériques sont à égalité

**Wi-Fi Mesh 802.11s** : chaque périphérique peut en plus servir de relais (voir [http://www.ensta-bretagne.fr/lebars/Share/mesh\\_network.txt](http://www.ensta-bretagne.fr/lebars/Share/mesh_network.txt) pour un exemple de configuration pour un Ubiquiti Bullet M2)



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Configuration du réseau pour l'accès à distance

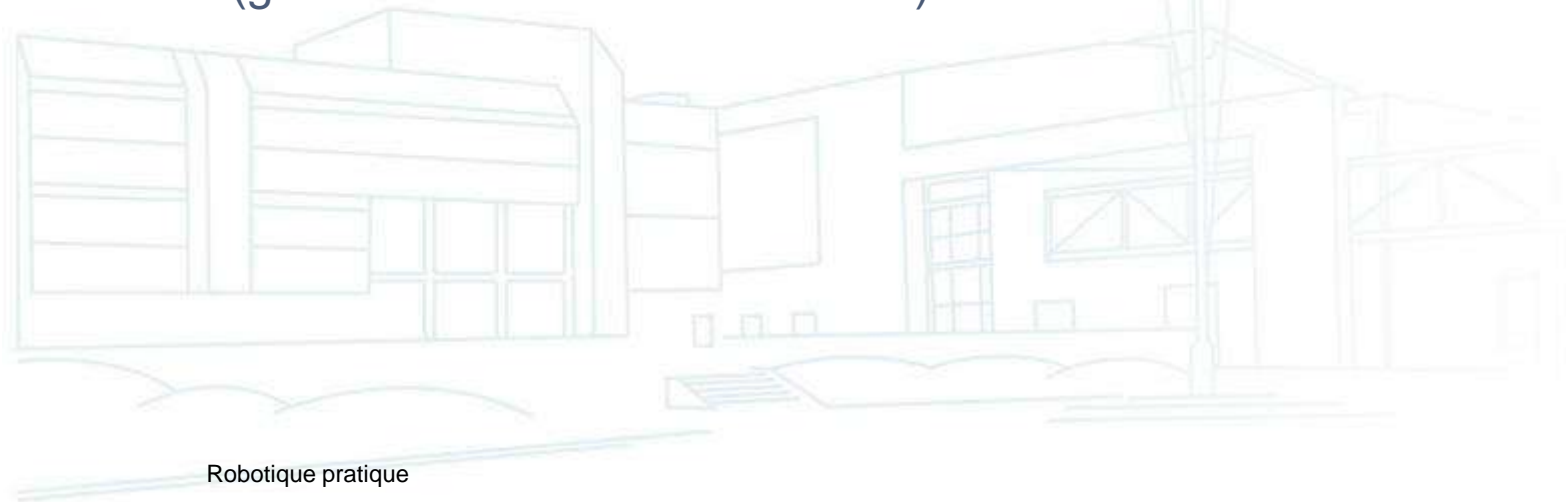
Prérequis/Rappels :

**Pont/Bridge** : similaire à un switch à 2 ports

**Commutateur/Switch** : périphérique utilisé pour relier plusieurs périphériques sur un même réseau

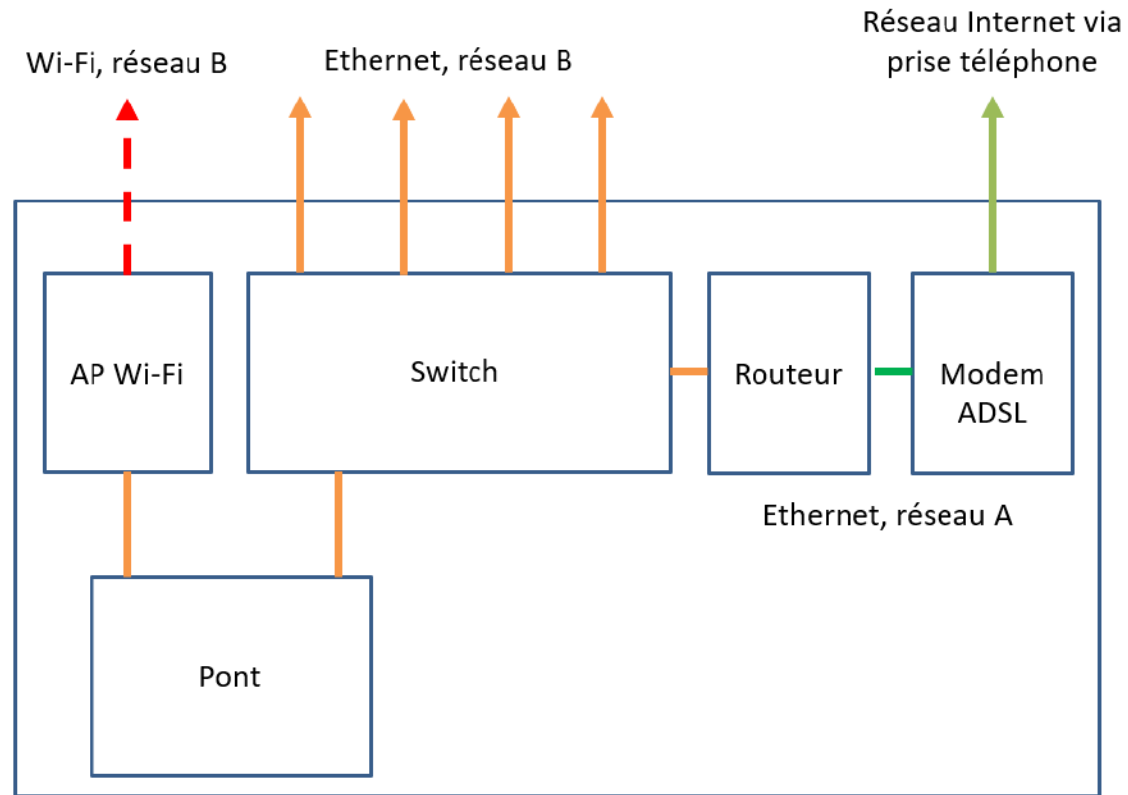
**Hub** : similaire à un switch, en moins intelligent...

**Routeur** : périphérique utilisé pour relier des réseaux IP différents (gammes d'adresses différentes)



# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Configuration du réseau pour l'accès à distance  
Exemple : Livebox



Livebox

# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Configuration du réseau pour l'accès à distance

Blocages possibles :

**Pare-feu/Firewall** : blocage de port réseaux TCP/UDP spécifiques ou blocage de l'accès réseau pour certaines applications

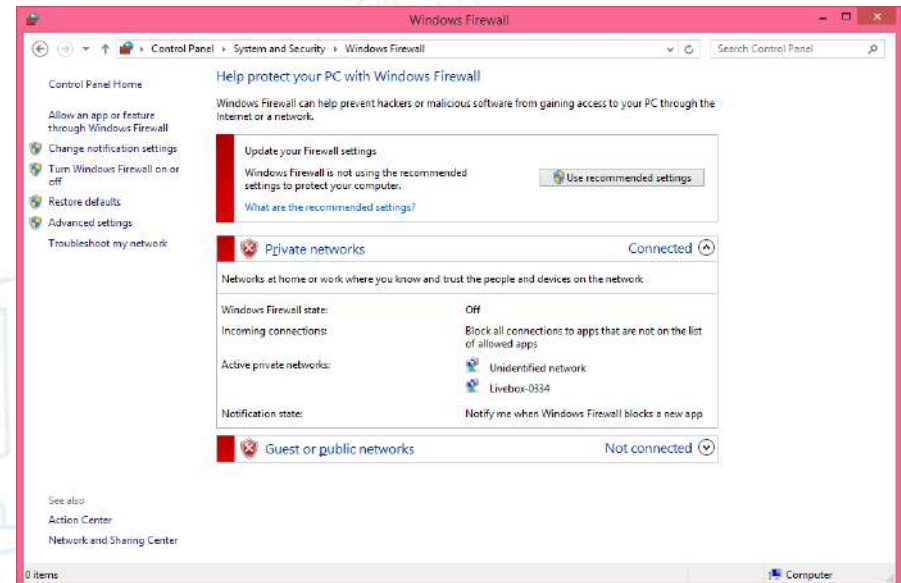
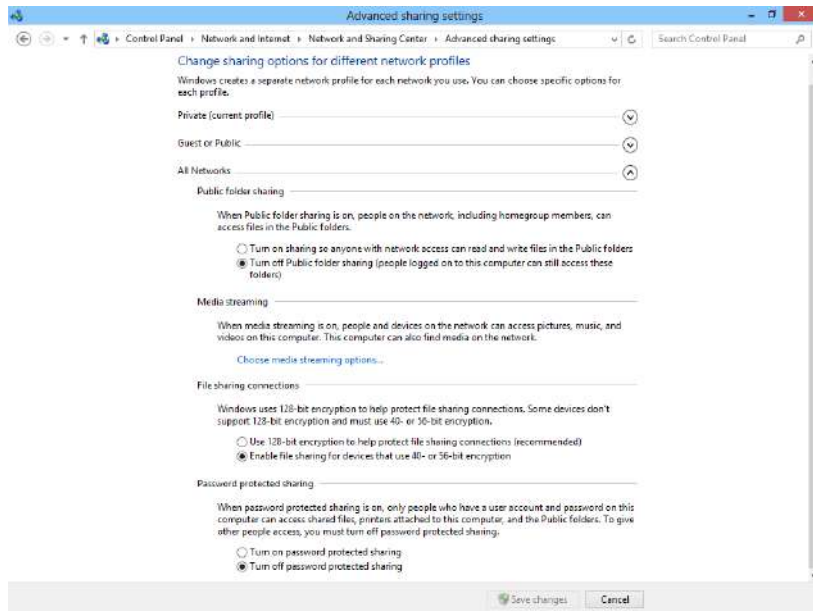
**Antivirus** : blocage de certaines applications

**Profils réseaux** : Windows (ou des logiciels tiers) est capable de reconnaître le réseau sur lequel on est connecté et d'appliquer des paramètres spécifiques (e.g. paramètres du pare-feu) à ce réseau, ces paramètres peuvent changer si le réseau change (voir `secpol.msc\Network List Manager Policies`)



# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
    - Configuration du réseau pour l'accès à distance
- Blocages possibles :  
Profils réseaux Windows 8 :  
2 types définis : **Public** ou **Private**



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

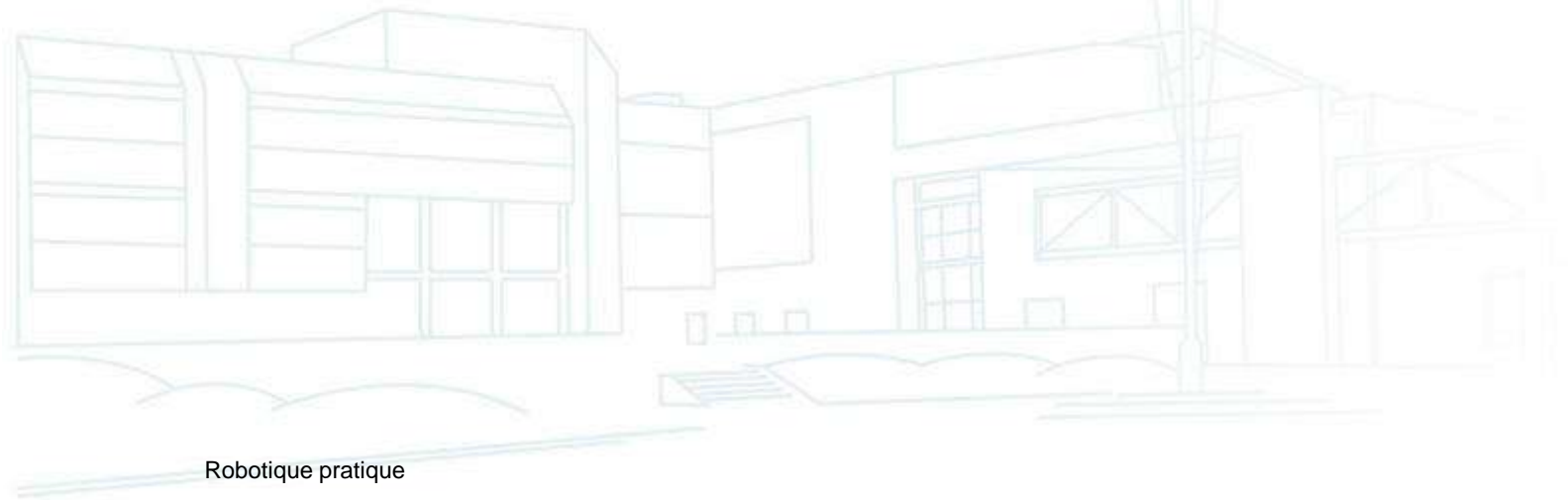
- Configuration du réseau pour l'accès à distance

Blocages possibles :

Profils réseaux Windows 8 :

2 types définis : **Public** ou **Private**, pour régler on peut e.g. utiliser **regedit** et mettre **Category** à **1** (Private) ou **0** (Public) dans les réseaux listés dans

**HKLM\SOFTWARE\Microsoft\Windows  
NT\CurrentVersion\NetworkList\Profiles**



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Configuration du réseau pour l'accès à distance

Blocages possibles :

Périphérique réseau muni de pare-feu : e.g. les routeurs ont souvent des réglages de pare-feu

**Proxy** : les accès réseaux (typiquement Internet) ne se font qu'en passant par l'intermédiaire d'un périphérique spécifique, qui peut ou non accepter de laisser passer certaines informations

Exemple : configuration du proxy de l'école pour divers usages

<http://www.ensta-bretagne.fr/lebars/Share/apt.conf>

Commandes Linux

```
export http_proxy=http://192.168.1.17:8080
export https_proxy=https://192.168.1.17:8080
export ftp_proxy=ftp://192.168.1.17:8080
export socks_proxy=socks://192.168.1.10:822
```

Proxy socks : voir e.g. Proxifier (Windows) pour pouvoir l'utiliser avec tout protocole basé sur les sockets (contrairement à juste http, https, ftp)



# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
    - Configuration du réseau pour l'accès à distance
- Tests, diagnostics :
- ipconfig, ifconfig
  - ping, ping -4, ping -6
  - Look@LAN, nmap



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Configuration du réseau pour l'accès à distance

Tests, diagnostics :

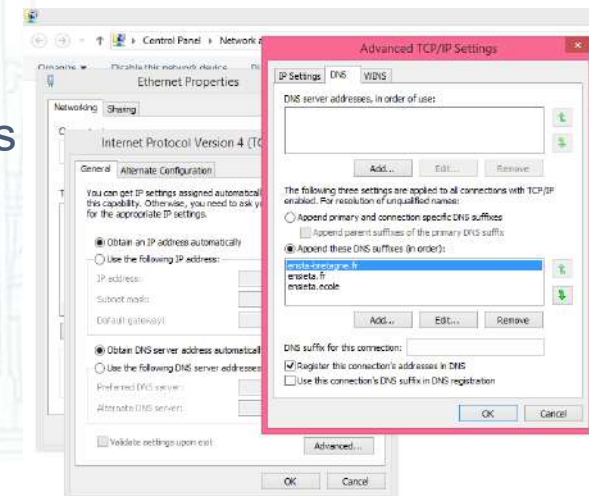
Cohérence adresses IP, DNS, adresses MAC

nslookup, arp -a

Délais possibles lors de changements...

**Nom de domaine / groupe de travail, nom de l'ordinateur** : Concepts assez présents dans les réseaux d'entreprises ou écoles sous Windows

Suffixe DNS pour pouvoir juste taper **moodle** et non **moodle.ensta-bretagne.fr** ou **portail** au lieu de **portail.ensieta.ecole**...



# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Configuration du réseau pour l'accès à distance

Concept de serveur / client :

**Serveur** : programme (ou par abus de langage ordinateur avec un ou des programmes de ce type) attendant des connexions avec un protocole spécifique

**Client** : programme (ou par abus de langage ordinateur avec un ou des programmes de ce type) se connectant à un serveur quand il en a besoin

Exemple : un **serveur web** est un programme attendant des connexions de clients sur le port réseau TCP 80. Il gère toutes les pages web et autres données du site. Lorsqu'un **client navigateur web** (Google Chrome, Internet Explorer, Mozilla Firefox sont des exemples) s'y connecte et demande une page web en particulier, il va la lui envoyer et le client va l'afficher à l'utilisateur. Typiquement un serveur web est sur un ordinateur puissant capable de gérer des milliers de connexions d'ordinateurs clients.

# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Configuration du réseau pour l'accès à distance

Concept de serveur / client :

Exemple avec le protocole **http** : pour demander la page web par défaut du serveur accessible par l'URL <http://checkip.dyndns.org/> (l'adresse IP correspondante est 216.146.38.70), un client peut envoyer (e.g. via des sockets TCP/IPv4 sur le port 80) :

```
GET http://checkip.dyndns.org/ HTTP/1.1\r\nHost: 216.146.38.70\r\nAccept: text/html\r\n\r\n\r\n
```

Le serveur répondra :

```
HTTP/1.1 200 OK\r\nContent-Type: text/html\r\nServer: DynDNS-CheckIP/1.0\r\nConnection: close\r\nCache-Control: no-cache\r\nPragma: no-cache\r\nContent-Length: 105\r\n\r\n\r\n<html><head><title>Current IP Check</title></head><body>Current IP Address: 86.229.255.67</body></html>\r\n\r\n\r\n
```

# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Configuration du réseau pour l'accès à distance

Concept de serveur / client :

Exemple avec le protocole **smtp** : pour demander à un serveur mail (e.g. smtp.orange.fr, 80.12.242.10) d'envoyer un mail, un client peut envoyer (e.g. via des sockets TCP/IPv4 sur le port 25) :

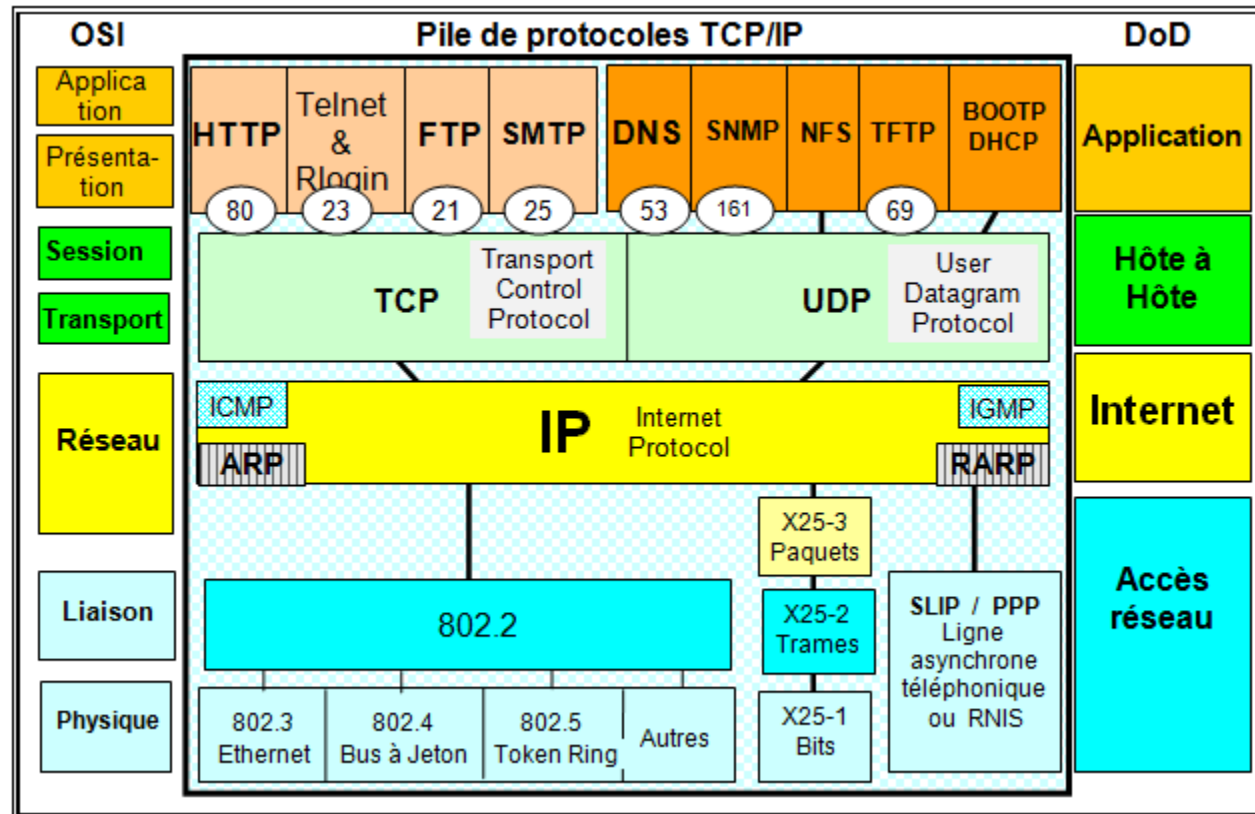
```
HELO 86.229.255.67\r\n
MAIL FROM:<sender_example@ensieta.fr>\r\n
RCPT TO:<receiver_example@ensieta.fr>\r\n
DATA\r\n
From: "Sender EXAMPLE" <sender_example@ensieta.fr>\r\n
To: "Receiver EXAMPLE" <receiver_example@ensieta.fr>\r\n
Subject: Test e-mail\r\n
\r\n
Message content test.\r\n
\r\n
.\r\n
QUIT\r\n
```

Le serveur répondra :

```
220 mwinf5d37 ME ESMTP server ready\r\n
250 mwinf5d37 hello [86.229.255.67], pleased to meet you\r\n
250 2.1.0 <sender_example@ensieta.fr> sender ok\r\n
250 2.1.5 <receiver_example@ensieta.fr> recipient ok\r\n
354 enter mail, end with "." on a line by itself\r\n
250 2.0.0 QLf61u00D1U1h4N03Lf8Vb mail accepted for delivery\r\n
221 2.0.0 mwinf5d37 ME closing connection\r\n
```

# Communiquer, contrôler à distance un système informatique autonome

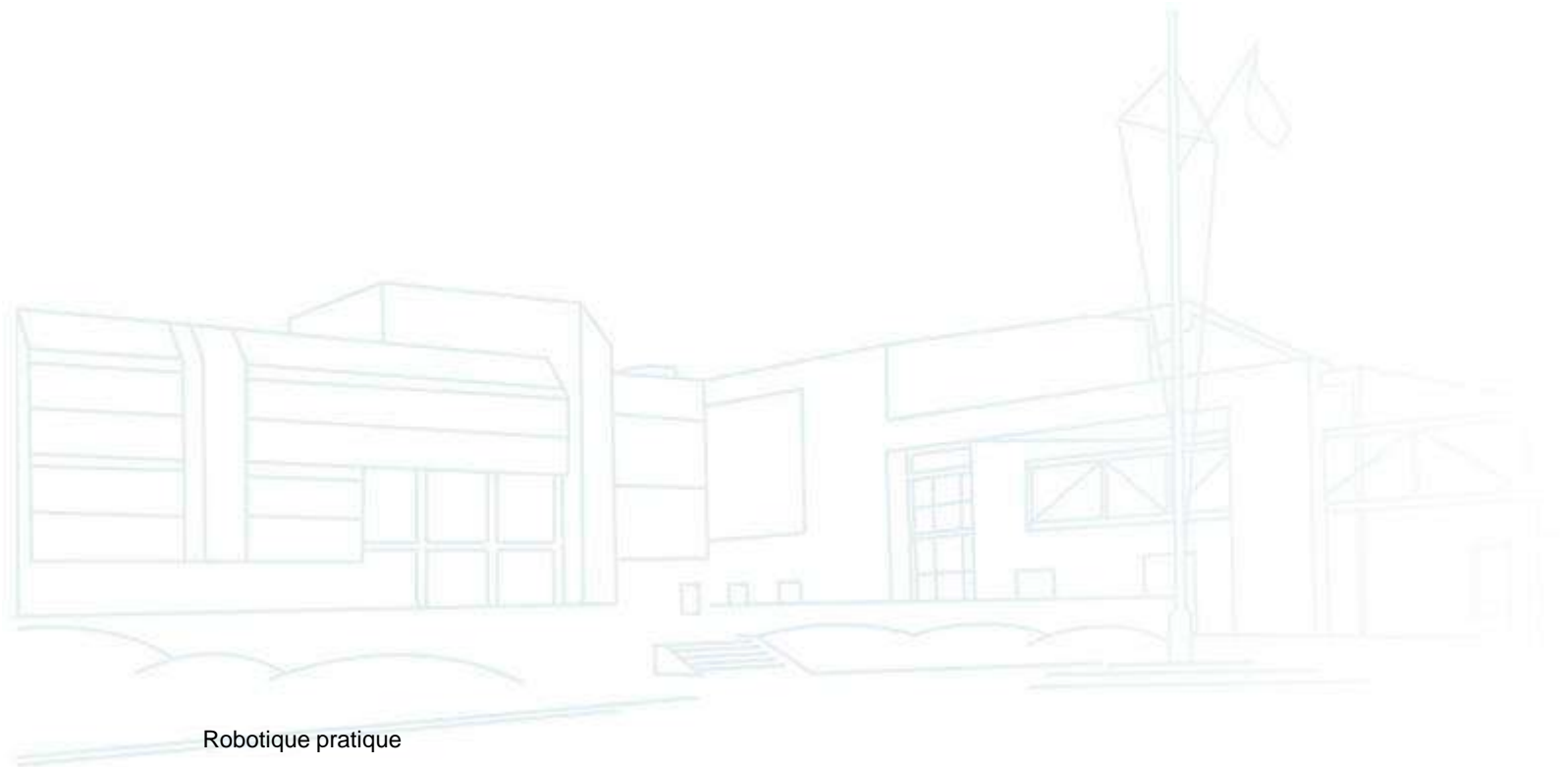
- Installation d'un PC :
    - Configuration du réseau pour l'accès à distance
- Concept de pile de protocoles :





# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
  - Configuration du réseau pour l'accès à distance  
Exécuter des commandes à distance :  
**SSH**, telnet



# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Configuration du réseau pour l'accès à distance

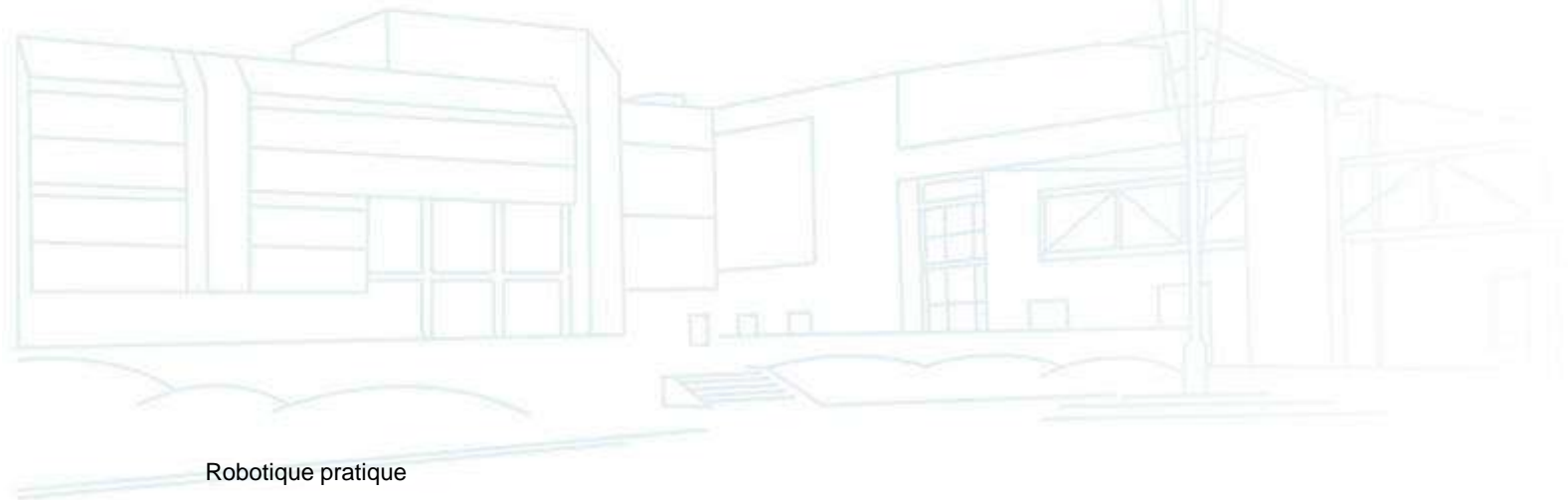
Transmettre des fichiers à distance :

**Lecteurs réseaux** (samba/smb), **SFTP**, FTP, SCP

Exemple : lecteurs réseaux école, utilisation des imprimantes (\\print3.ensieta.ecole), utilisation depuis l'extérieur avec VPN (<https://vpn2.ensta-bretagne.fr>)

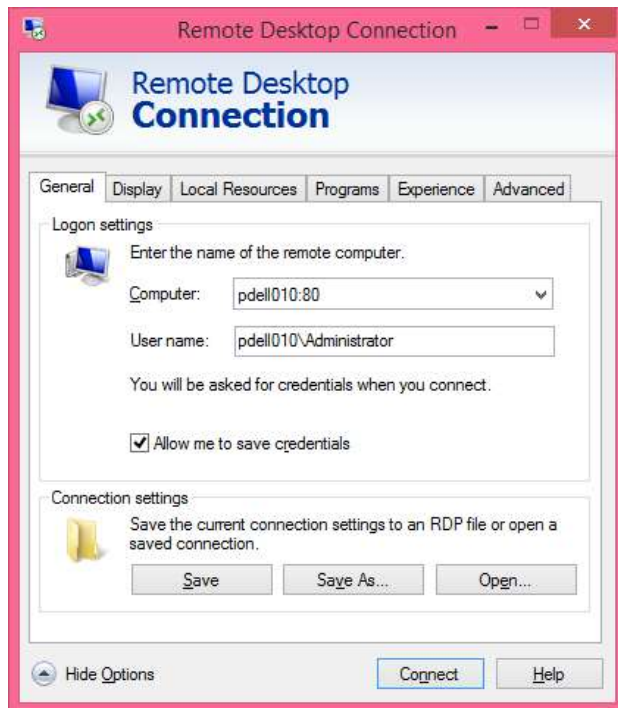
<http://www.ensta-bretagne.fr/lebars/Share/Network%20drives%20ENSIETA.bat>

[http://www.ensta-bretagne.fr/lebars/Share/remote\\_ensieta.sh](http://www.ensta-bretagne.fr/lebars/Share/remote_ensieta.sh)



# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
    - Configuration du réseau pour l'accès à distance
- Bureau à distance :
- RDP, NoMachine, Team Viewer, VNC, ssh+X**



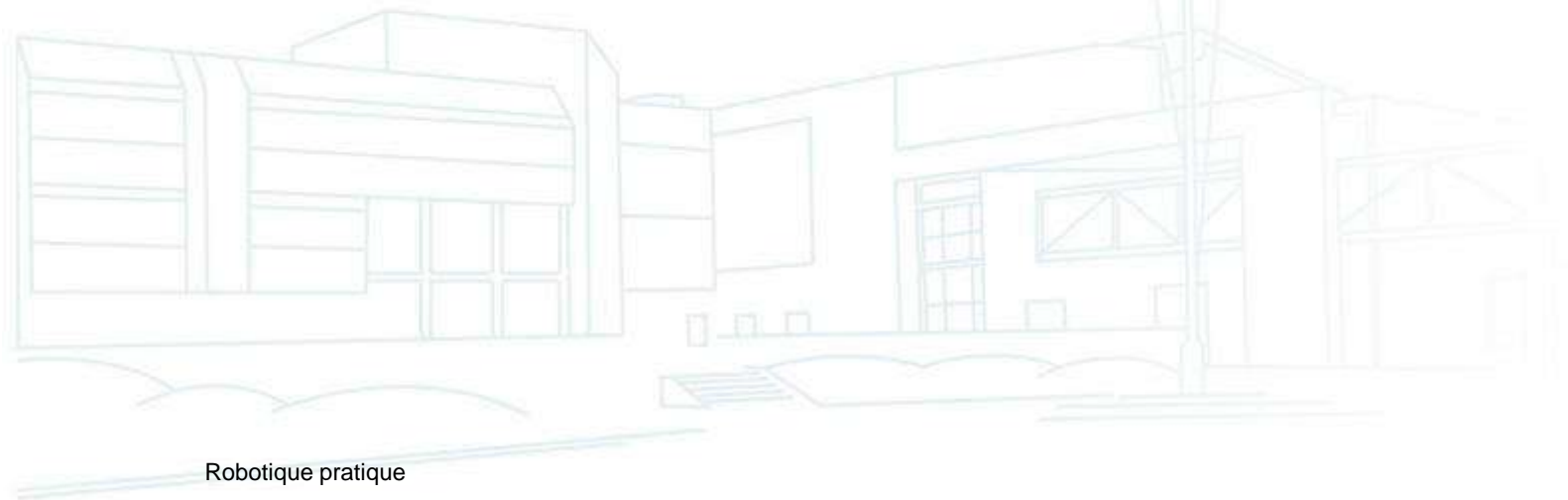
Note : Sur certains PC pour **mstsc /admin** il faut préciser le nom de domaine (nom du serveur si pas dans un domaine)...

# Communiquer, contrôler à distance un système informatique autonome

## ■ Installation d'un PC :

- Problèmes de droits d'accès

Faire **sudo** si message type « permission denied » sous Linux, sous Windows l'équivalent est clic-droit « **Run as administrator** », autrement faire **chmod -R 777 foldername** ou **chmod 777 file** pour donner tous les droits d'accès sur des fichiers ou dossiers particuliers sous Linux, sous Windows l'équivalent s'obtient via clic-droit **Properties\Security**.

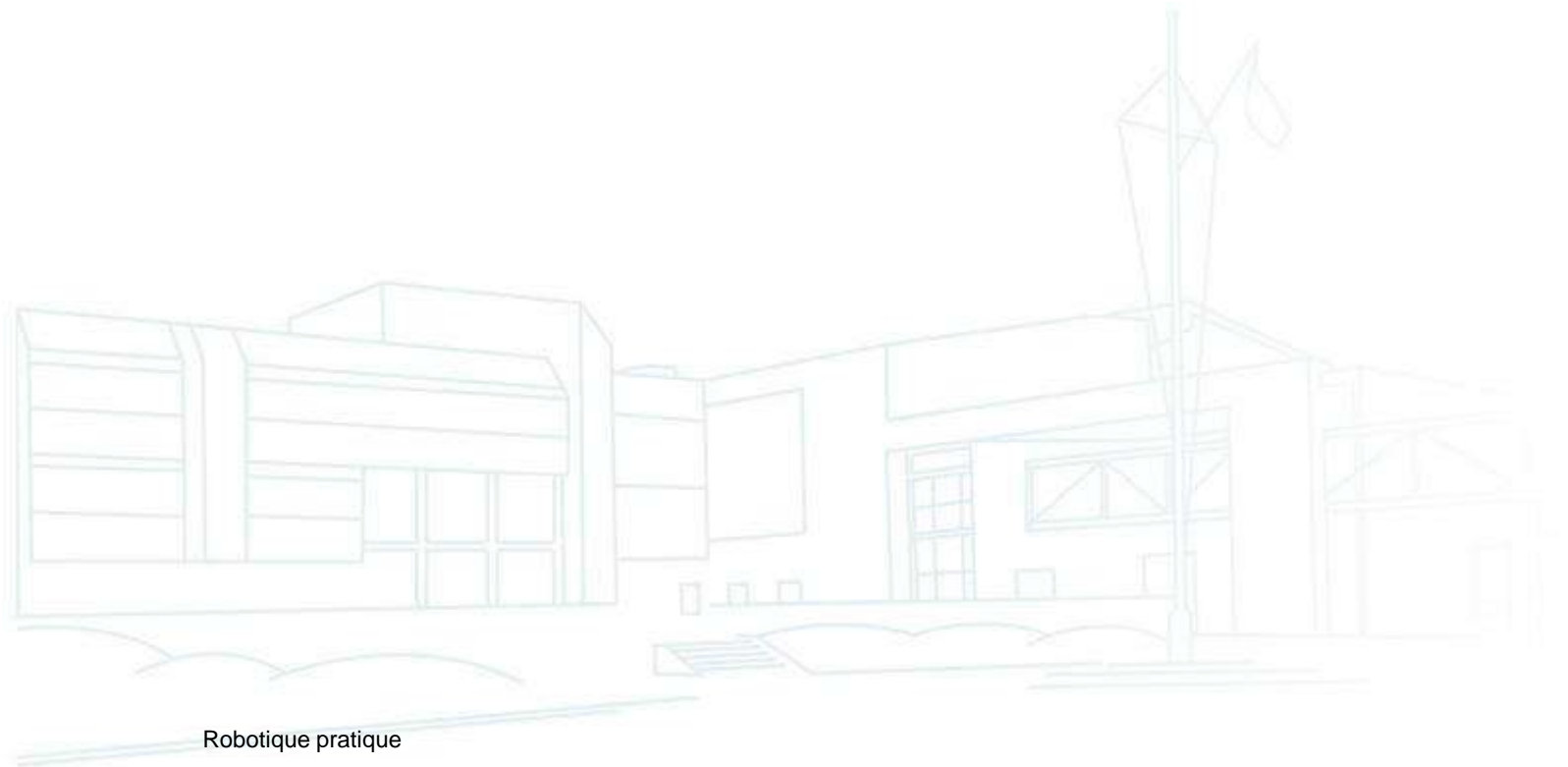


# Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :

- Utilisation de bibliothèques spécifiques en C/C++

Voir : [http://www.ensta-bretagne.fr/lebars/tutorials/Complements\\_C-C++.pdf](http://www.ensta-bretagne.fr/lebars/tutorials/Complements_C-C++.pdf)

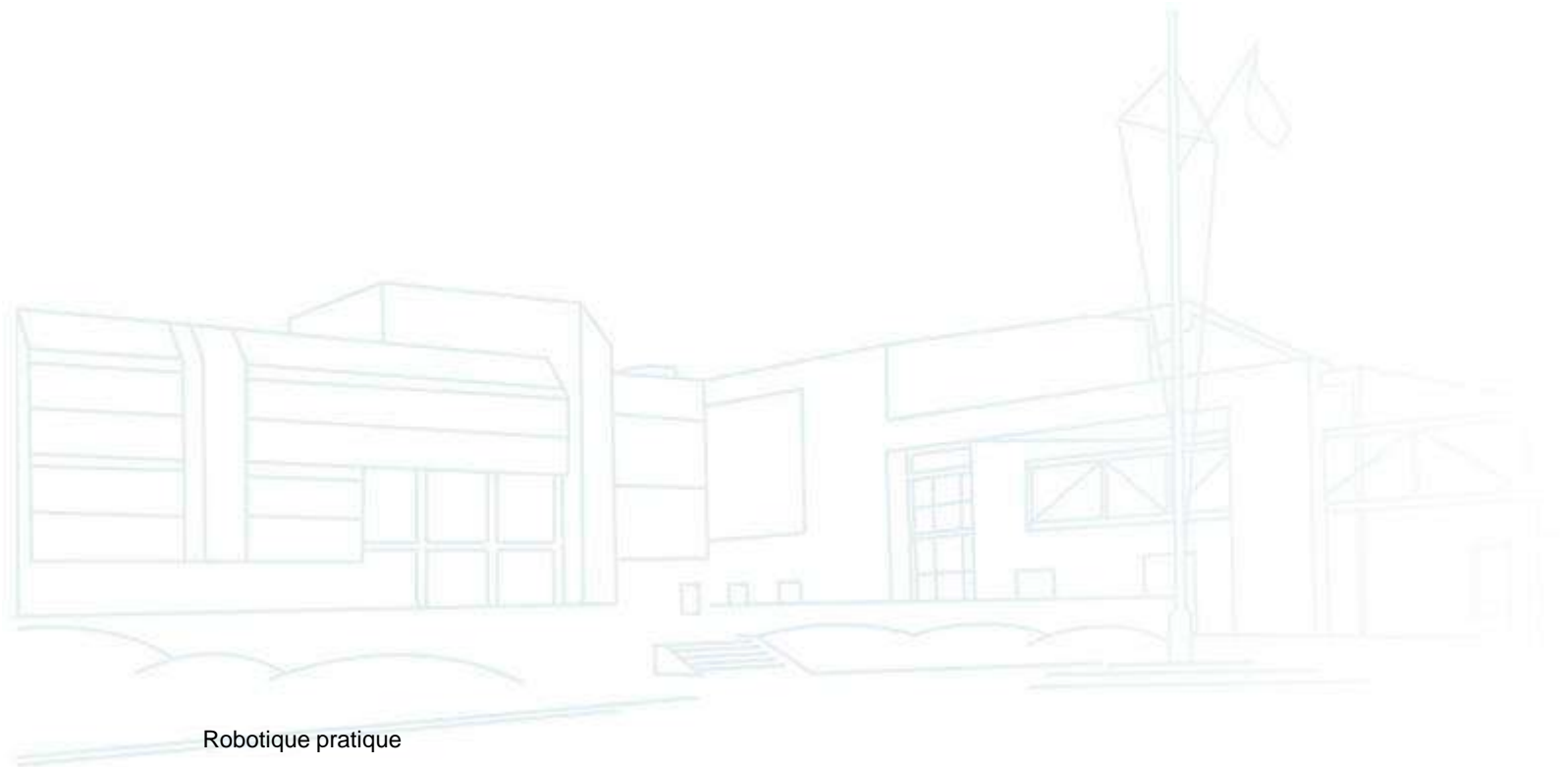




# Algorithmes pour l'autonomie



- But :
  - Rendre un robot autonome i.e. capable d'effectuer des missions de détection, localisation, cartographie, mesures avec un minimum d'intervention humaine



## ■ Simulation :

### • Robot **en cours de construction** :

Réaliser un **simulateur logiciel** pour commencer à mettre au point les algorithmes de régulation, etc.

Réaliser un **modèle réduit ou simplifié (simulateur matériel)** du vrai robot...

### • Robot **téléopéré** ou véhicule conduit par un humain **existant** :

**Ajouter des capteurs et enregistrer les commandes** des actionneurs pour se donner une idée de son comportement dynamique

On pourra alors se servir de ces données pour mettre au point un simulateur

## ■ Simulation :

- Intérêts de faire d'abord des simulations :

Moins de risque de casse

Débuggage en général plus facile

etc.

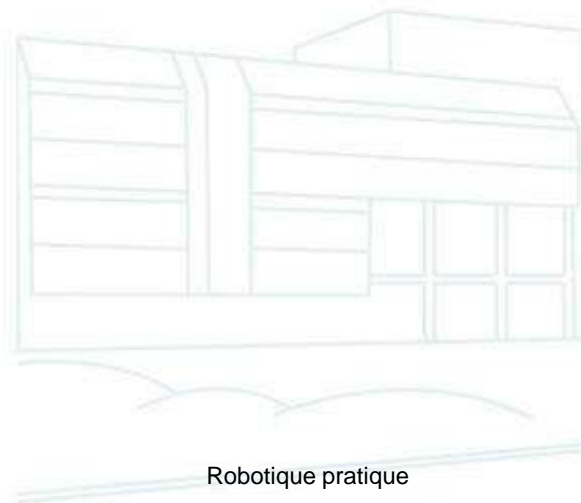
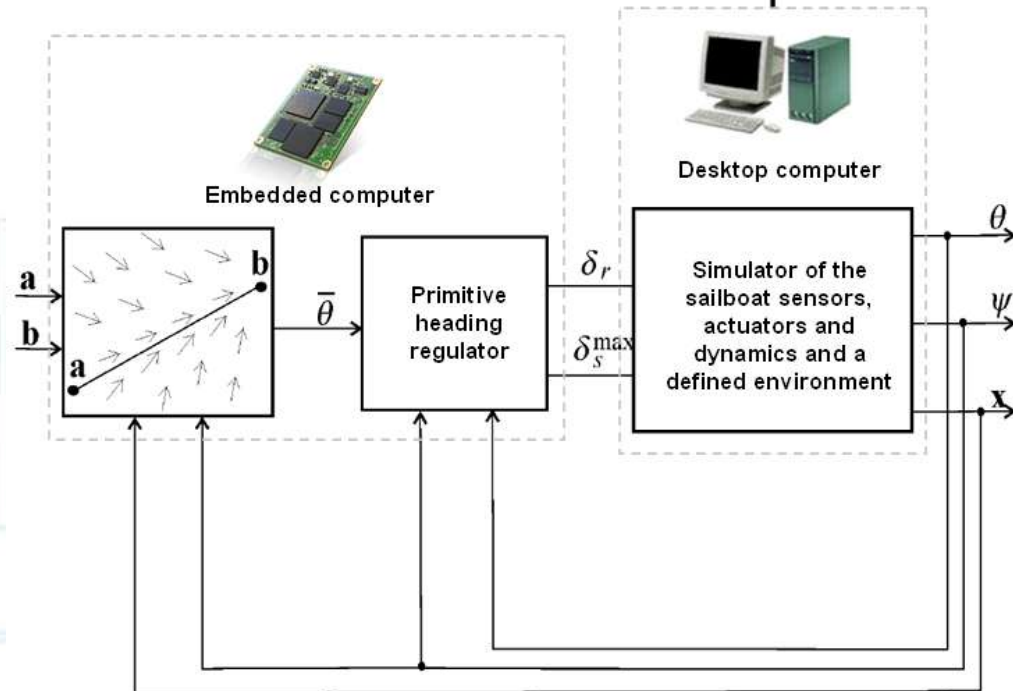
- Inconvénients :

Parfois difficile d'obtenir des modèles décrivant correctement le robot  
=> faire des expériences en téléopéré pour prendre des données si possible

Modèle trop complexe => éventuellement découper en plusieurs simulateurs pour tester différents aspects particuliers

Risque de développer des algorithmes qui ne fonctionnent pas en pratique => prévoir des simulations HIL (Hardware In the Loop)

- Exemple de simulation  
HIL (Hardware In the  
Loop) : avec VAIMOS

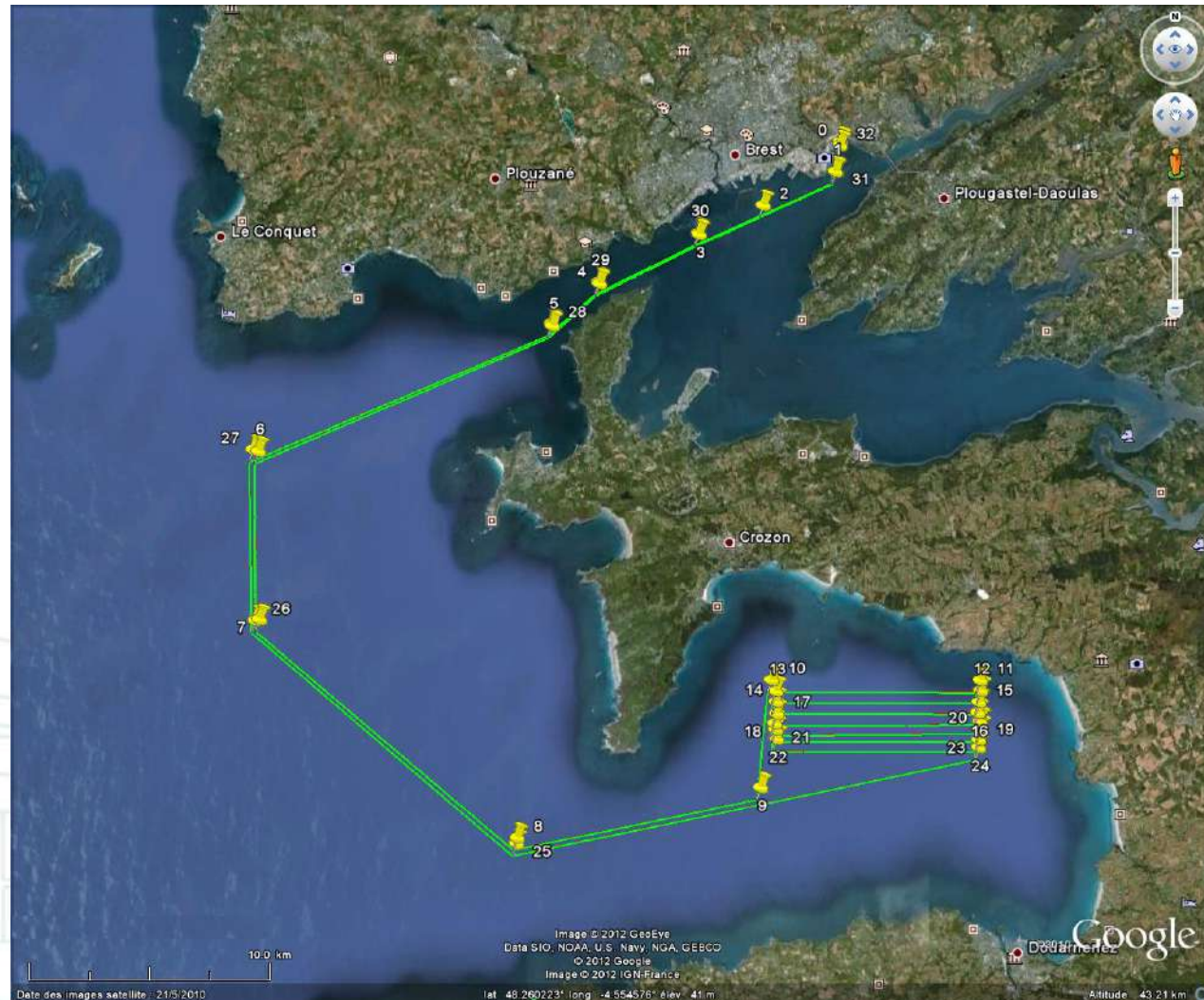


- Exemple de simulation HIL (Hardware In the Loop) : avec VAIMOS





- Exemple de simulation HIL (Hardware In the Loop) : avec VAIMOS



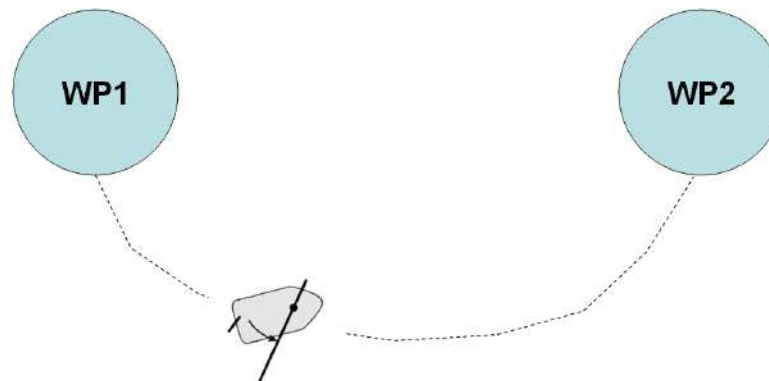


# Algorithmes pour l'autonomie

- Simulation de robots en MATLAB :
  - Voir [http://www.ensta-bretagne.fr/lebars/simu\\_robots\\_matlab.ppt](http://www.ensta-bretagne.fr/lebars/simu_robots_matlab.ppt)
- Régulation simple de robot pour du suivi de waypoints :
  - Voir [http://www.ensta-bretagne.fr/lebars/buggy\\_control\\_simple.ppt](http://www.ensta-bretagne.fr/lebars/buggy_control_simple.ppt)



- Du suivi de waypoints au suivi de ligne :
  - Suivi de waypoints de base :
    - Le robot cherche à suivre un cap dans la direction du waypoint
    - Le waypoint est validé quand on est dans un rayon prédéfini
    - Problème : rien n'empêche qu'on dérive entre waypoints (e.g. sur l'eau à cause des courants...)

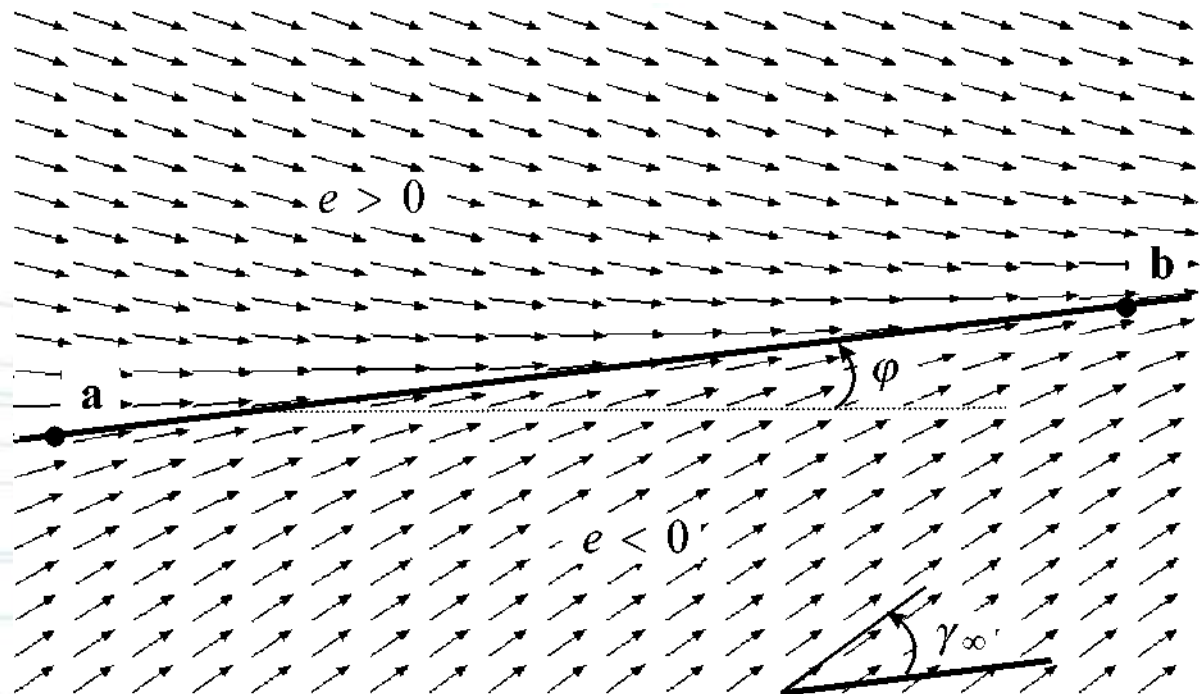


- => Suivi de ligne (utilise la distance à la ligne)

## ■ Suivi de ligne :

- Le cap à suivre va être la ligne entre les 2 waypoints courants, avec un angle d'attraction à la ligne en fonction de notre écart à la ligne

$$e = \det \left( \frac{\mathbf{b} - \mathbf{a}}{\|\mathbf{b} - \mathbf{a}\|}, \mathbf{m} - \mathbf{a} \right)$$
$$\varphi = \text{atan2}(\mathbf{b} - \mathbf{a})$$
$$\bar{\theta} = \varphi - \frac{2 \cdot \gamma_{\infty}}{\pi} \cdot \text{atan} \left( \frac{e}{r} \right)$$



## ■ Suivi de ligne :

- Un module de regulation en cap
  - Contrôle du gouvernail

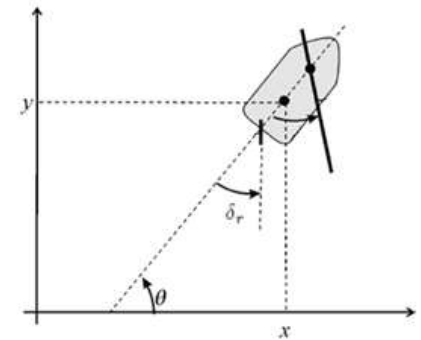
$$\delta_r = \begin{cases} \delta_r^{\max} \cdot \sin(\theta - \bar{\theta}) & \text{if } \cos(\theta - \bar{\theta}) \geq 0 \\ \delta_r^{\max} \cdot \text{sign}(\sin(\theta - \bar{\theta})) & \text{otherwise} \end{cases}$$

- Un superviseur envoie des caps à suivre au module de regulation en cap en fonction de la ligne courante

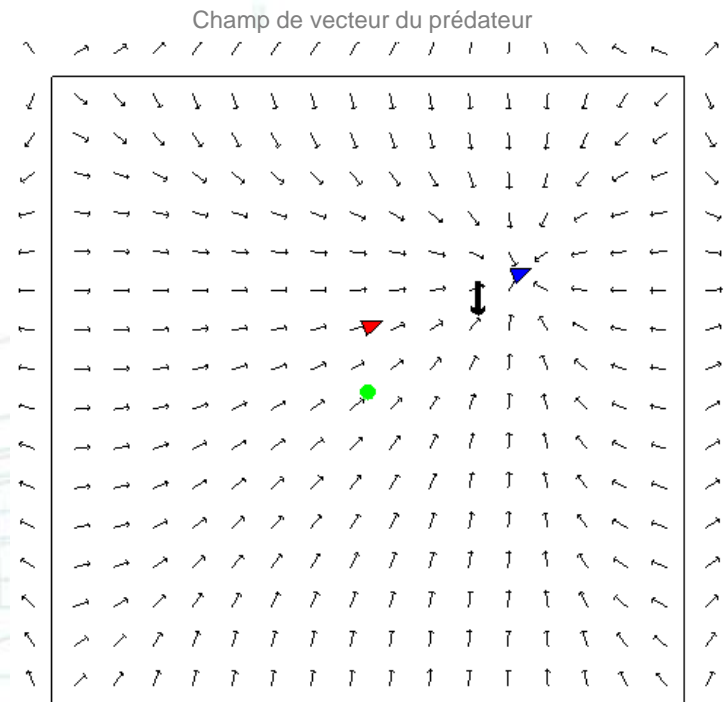
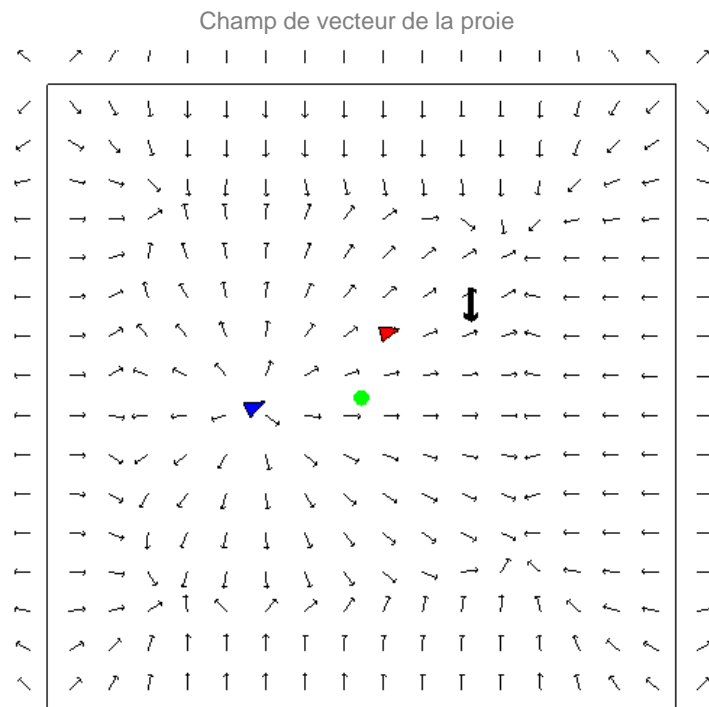
$$\begin{aligned} e &= \det\left(\frac{\mathbf{b}-\mathbf{a}}{\|\mathbf{b}-\mathbf{a}\|}, \mathbf{m}-\mathbf{a}\right) \\ \varphi &= \text{atan2}(\mathbf{b}-\mathbf{a}) \\ \bar{\theta} &= \varphi - \frac{2 \cdot \gamma_{\infty}}{\pi} \cdot \text{atan}\left(\frac{e}{r}\right) \end{aligned}$$

- Un module de gestion de la navigation envoie des lignes au superviseur et decide de leur validation
  - Condition de validation

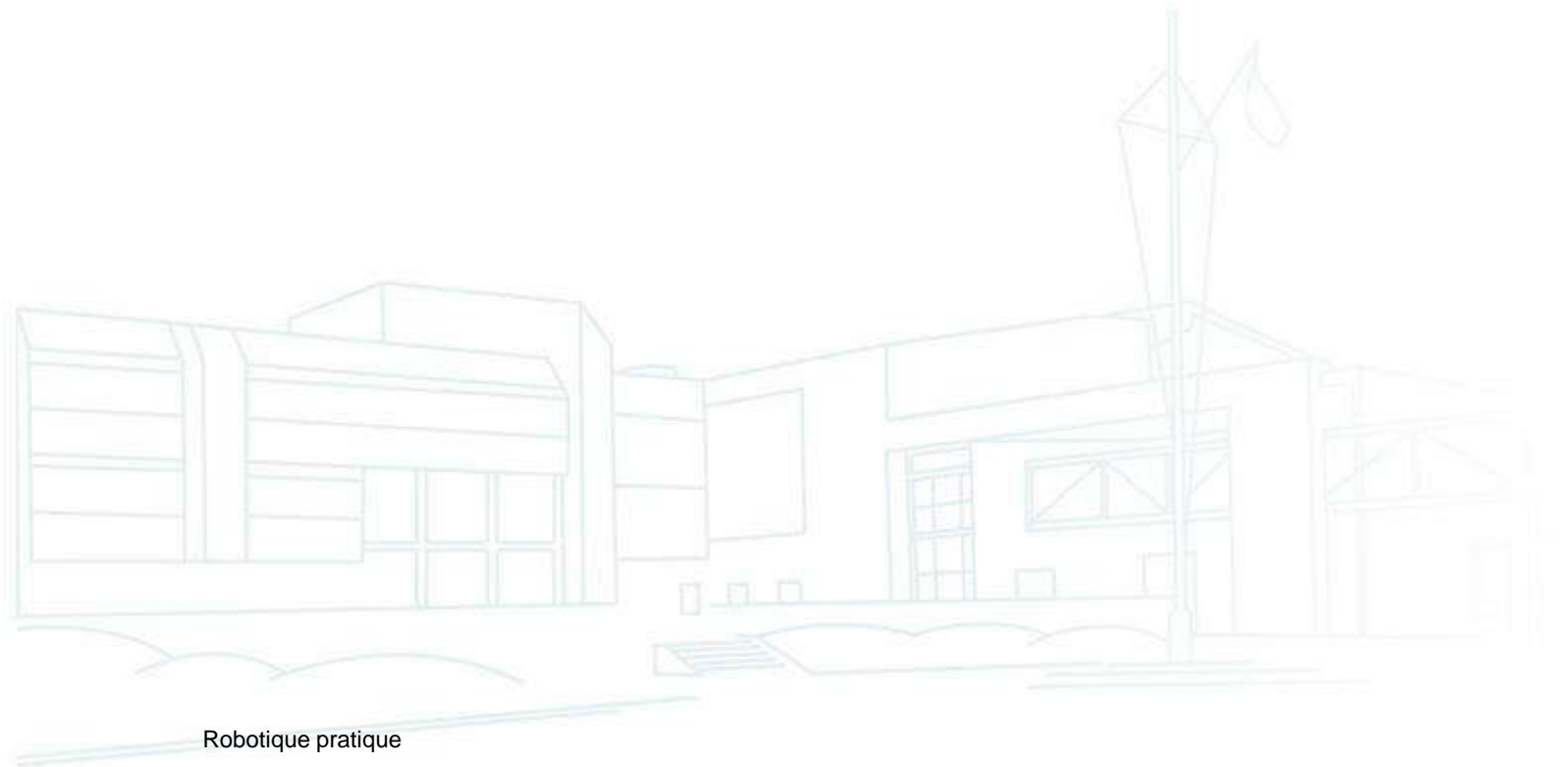
$$\langle \mathbf{b}_j - \mathbf{a}_j, \mathbf{m} - \mathbf{b}_j \rangle \geq 0$$



- Suivi de champs de potentiels :
  - Permet la combinaison de points ou lignes attractifs, répulsifs...
  - Exemple : proie et prédateur dans un carré dont il ne faut pas sortir



- Autres régulations :
  - Voir <http://robmooc.ensta-bretagne.fr/>





## ■ Localisation :

- **Connaître sa position** à tout moment est indispensable dans la plupart des cas
- Cependant, des **capteurs** ne sont **pas toujours disponibles** pour retourner directement une position précise quelles que soient les conditions...

Exemple : Le GPS marche bien quand on roule sur une route dégagée mais dans un long tunnel il ne marche pas du tout

- Il faut donc trouver des méthodes théoriques et numériques pour évaluer ces valeurs : on appelle **observateur d'état** ces méthodes

## ■ Localisation :

- En Automatique 1A, vous avez peut-être vu les **observateurs de Luenberger pour les systèmes linéaires invariants**, on peut en faire d'autres selon les spécificités du système étudié :

**Dead-reckoning** (navigation à l'estime) : en utilisant les équations d'état pour prédire la position à  $t+dt$  à partir de celle à  $t$

Calculs par relations géométriques : relations de distance et/ou d'angles à des amers (objets de position plus ou moins connue)

**Filtre de Kalman** : voir [http://www.ensta-bretagne.fr/lebars/Obs\\_et\\_filt\\_de\\_Kalman.ppt](http://www.ensta-bretagne.fr/lebars/Obs_et_filt_de_Kalman.ppt)

**Méthodes ensemblistes** : voir <https://www.ensta-bretagne.fr/jaulin/iamooc.html>

- Localisation :
  - Note : souvent en robotique, on va confondre **estimation d'état**, **observateur d'état** avec **localisation**, **estimation de position** car bien souvent c'est la position qui nous intéresse parmi les variables d'état



## ■ Localisation :

- Conversions de repères courantes :

Repères courants :

NED (North-East-Down)

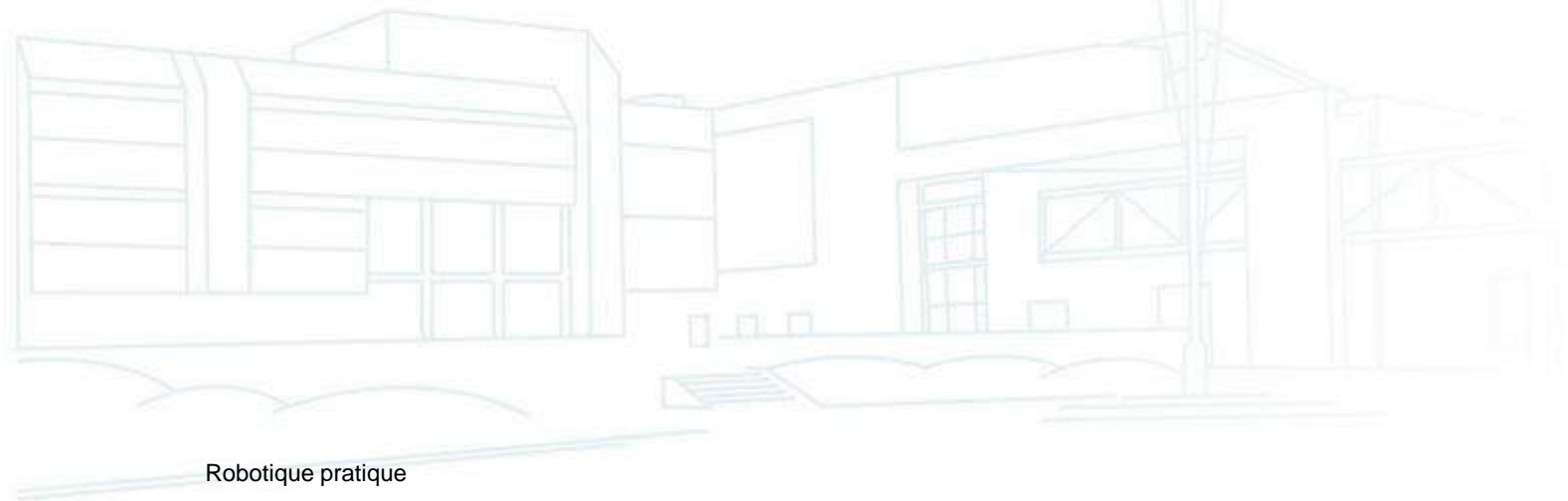
NWU (North-West-Up)

ENU (East-North-Up)

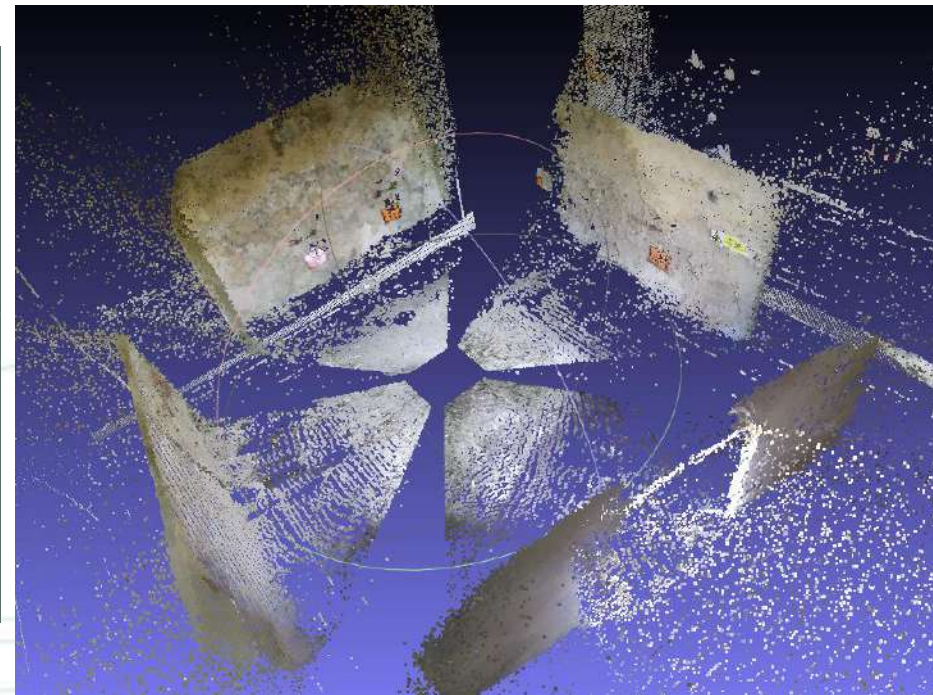
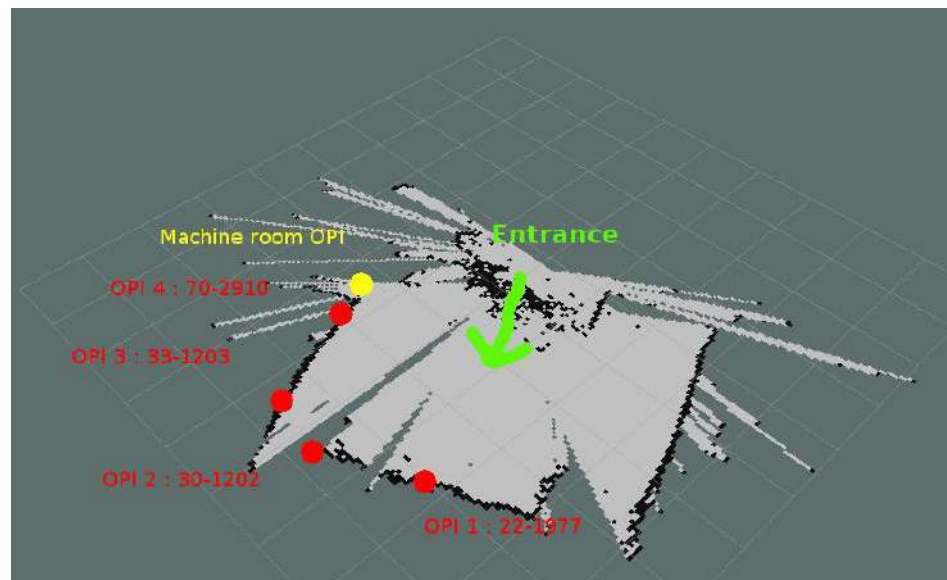
WGS84 (orientation des axes comme ENU)

Faire aussi la différence entre les concepts différents d'altitude (e.g. souvent pour un sous-marin la distance w.r.t. fond), profondeur (distance w.r.t. surface), qui peuvent être différents selon les capteurs et le principe physique qu'ils utilisent

- Localisation :
  - Conversions de repères courantes :
    - NED->ENU : 90-yaw, -pitch, roll
    - WGS84 vers repère ENU local : voir [http://www.ensta-bretagne.fr/lebars/these/thesis\\_2011-11-23.pdf](http://www.ensta-bretagne.fr/lebars/these/thesis_2011-11-23.pdf) (p28) ou « A simple controller for line following of sailboats » sur <http://www.ensta-bretagne.fr/jaulin/> pour gérer le fait que la Terre ne peut être considérée comme plate sur de longues distances



- Cartographie :
  - **Dessin de l'environnement** du robot  
e.g. regroupement de données lidar, sonar, caméra, Kinect, etc.





## ■ Cartographie :

- Exemple de capteur intéressant pour la cartographie : Kinect 2 Xbox One

Regroupement de plusieurs capteurs :

- Une caméra Full HD dans le visible
- Une caméra dans l'infrarouge (avec éclairage actif)
- De nombreux micros



## ■ Cartographie :

- Exemple de capteur intéressant pour la cartographie : Kinect 2 Xbox One

Elle retourne :

- Des images **couleurs**
- Des images **infrarouges**
- Des images en niveaux de gris, celui-ci étant plus ou moins foncé en fonction de la distance des objets face au capteur (**profondeur**)

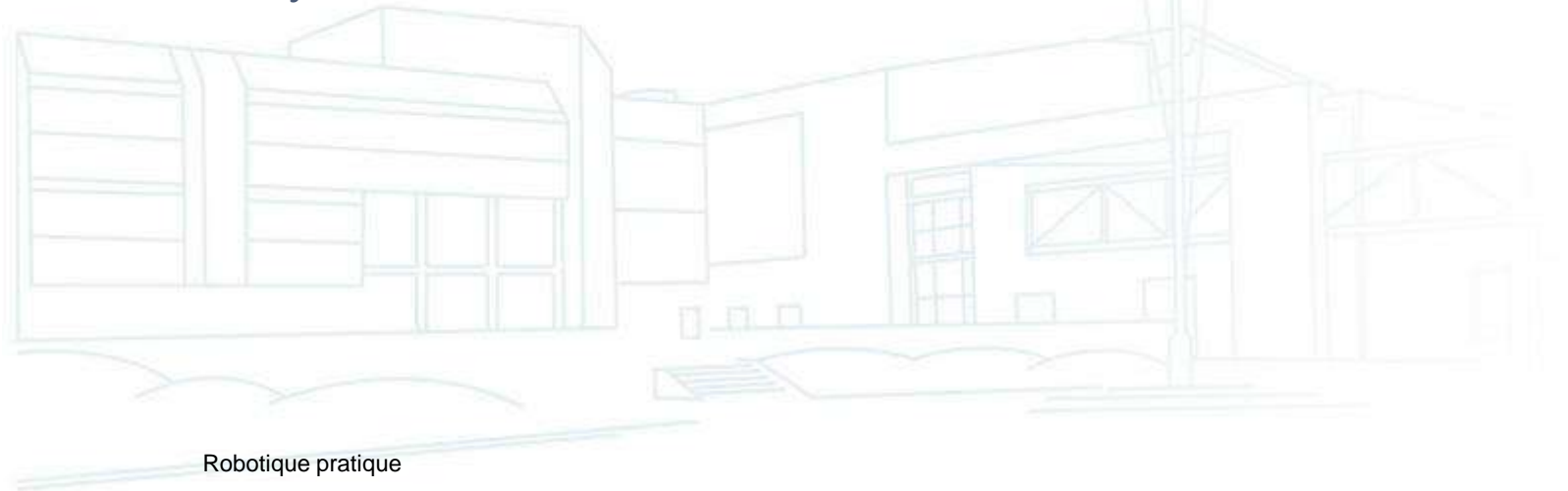


## ■ Cartographie :

- Exemple de capteur intéressant pour la cartographie : Kinect 2 Xbox One

Comme ces images viennent de capteurs séparés, plusieurs problèmes sont à considérer :

- Point de vue légèrement différent -> zones visibles sur un capteur mais pas sur l'autre
- Résolution et ouverture angulaire (FOV) différentes
- Symétries, etc.



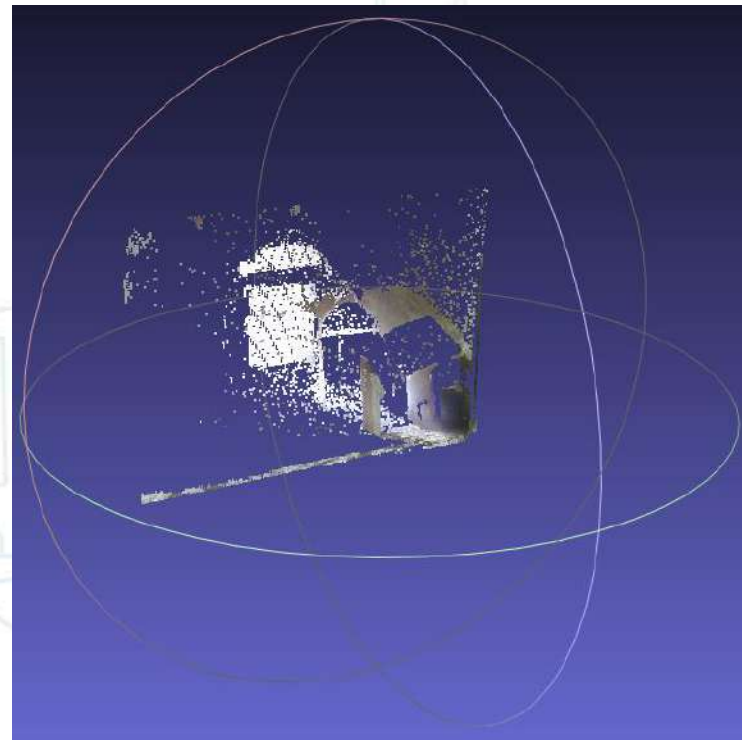
## ■ Cartographie :

- Exemple de capteur intéressant pour la cartographie : Kinect 2 Xbox One

Ensuite, connaissant l'ouverture angulaire de la caméra et la portée en profondeur, on peut combiner les informations de l'image couleur et l'image en profondeur pour dessiner un nuage de points colorés dans l'espace...



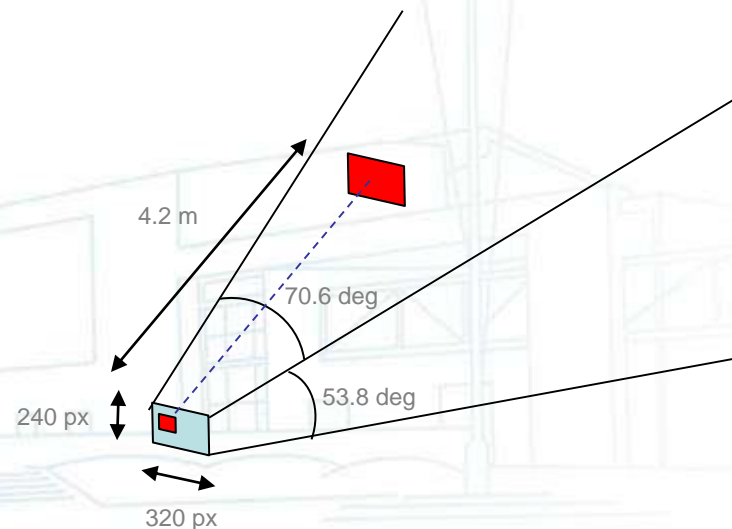
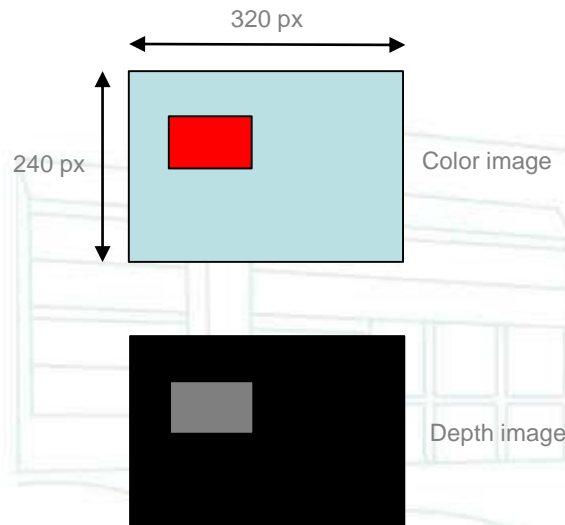
Robotique pratique



## ■ Cartographie :

- Exemple de capteur intéressant pour la cartographie : Kinect 2 Xbox One

Exemple : représentation dans l'espace d'un **groupe de pixels rouges** de l'image couleur, dont le **niveau de gris** correspondant dans l'image en profondeur se traduit par une **distance de 4.2 m**, si images e.g. de **résolution 320 x 240** avec un **FOV 70.6 x 53.8**

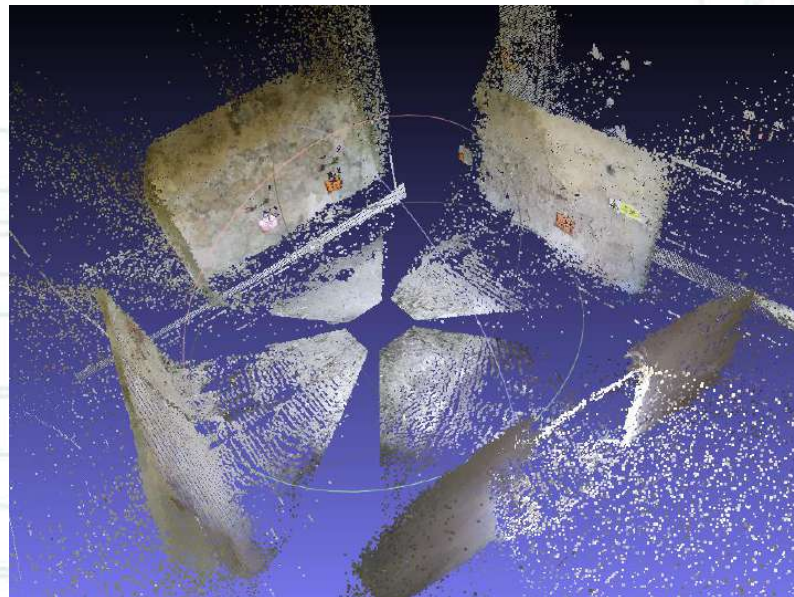




# Algorithmes pour l'autonomie

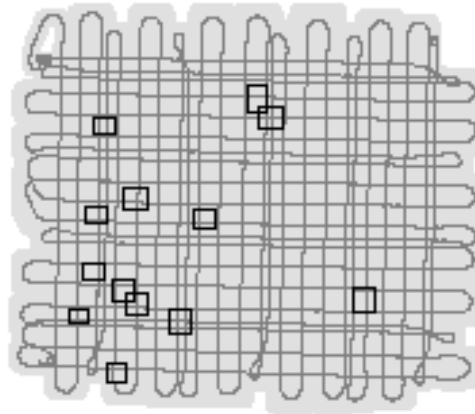
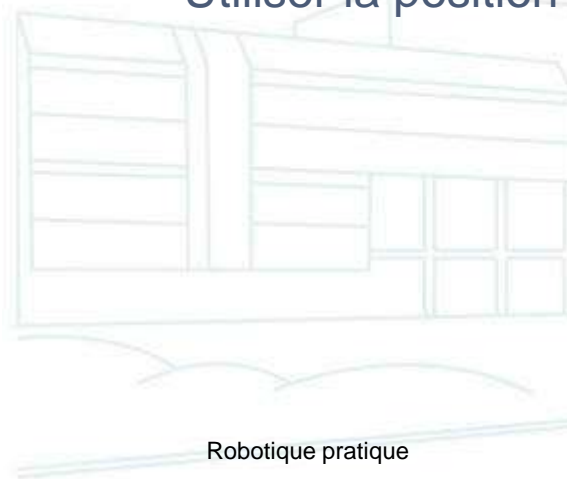


Robot ayant pris des données Kinect  
en tournant 4 fois de 90 deg





- SLAM (Simultaneous Localization And Mapping) :
  - Bien souvent on connaît ou le robot mesure :
    - Sa position initiale et finale
    - Son modèle de déplacement (équations d'état)
    - Ses données de navigation (orientation, vitesse, etc.)
    - Ses données de télédétection (images caméra, sonar, etc.)
  - Idée du SLAM :
    - Localiser des amers (points de repère) à partir de sa propre position
    - Utiliser la position de ces amers pour se localiser



- SLAM (Simultaneous Localization And Mapping) :
  - Association des données : savoir si un point qu'on voit est un nouvel amers ou un amers connu est un problème à part entière
  - Données aberrantes (outliers) : déterminer si une donnée de capteur est valide ou ne l'est pas est aussi un problème à considérer

