



Robotique pratique

Fabrice LE BARS

■ Notations

- Certaines slides ou points dans des slides seront marqués avec le symbole



pour indiquer des informations a priori **non indispensables** à la compréhension des parties importantes du cours et/ou **pas à apprendre par cœur** à ce stade, mais qu'il faudra peut-être **savoir retrouver** rapidement quand le besoin se présentera e.g. au cours d'un projet dans l'année

Par analogie avec le concept de pointeur en C/C++ : mémoriser le pointeur (i.e. l'emplacement de la slide) quand il y a le symbole, mémoriser les données (i.e. ce qui est dans la slide) quand il n'y est pas!

■ Sommaire

• Concevoir un robot

Structure et supports mécaniques

Actionneurs

Capteurs

Ordinateur embarqué

Autopilote

Communication

Energie

Connecteurs et câblage

Manipulations, interventions, tests, résolution de problèmes, etc.

Exemples d'architectures pour différents types de robots

• Communiquer, contrôler à distance un système informatique autonome

Principe de fonctionnement d'un ordinateur simplifié

Montage d'un PC

Installation d'un PC

• Algorithmes pour l'autonomie



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** ou **uncrewed**

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



Autres acronymes et expressions : MAV : Micro Aerial Vehicle, RPAS : Remotely Piloted Aircraft System, MASS : Maritime Autonomous Surface Ships, « aéronef sans pilote à bord » , AMR : Autonomous Mobile Robot (typiquement transport de charge en usine)

- **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
 - Possibilité de monter/démonter facilement?
 - 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 et supports mécaniques :
 - 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 cours de CAO avec le logiciel CATIA sur <https://moodle.ensta-bretagne.fr/mod/resource/view.php?id=32600> et atelier CNC sur https://www.ensta-bretagne.fr/lebars/atelier_CNC.pdf



■ Structure et supports mécaniques :

• Imprimante 3D

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

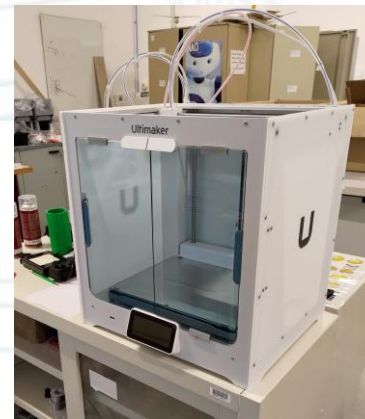
Eviter les simples plaques trouées, plutôt découper une plaque PVC et percer à la main

Eviter les grandes pièces, la température non uniforme risque de provoquer des déformations aux extrémités, essayer de décomposer en assemblage

Percer les trous à la main, ceux faits par l'imprimante risquent de ne pas être précis

Attention aux différences dans le modèle CAO après conversions e.g. en GCODE!

Les impressions prennent en général plusieurs heures voire jours et des changements de rouleaux de matière peuvent être nécessaires!



- Structure et supports mécaniques :
 - Perceuse-taraudeuse : pour faire des trous rapidement

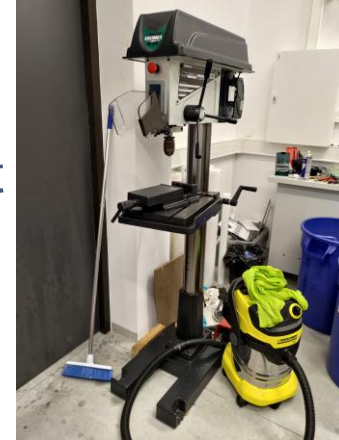
Attention aux vêtements, cheveux qui peuvent se prendre dedans, bien attacher les pièces, utiliser limiteur de couple, gants anti-coupure...

Tips :

Aligner foret dans trou existant puis monter plus gros foret quand on souhaite agrandir un trou existant. Si on doit agrandir de très peu, éventuellement utiliser une lime aiguille e.g. <https://www.amazon.fr/gp/product/B07RZ6DXVQ/>

Pour perçage transversal d'un axe, limer pour rendre plan l'endroit où on veut percer, pour pas que le foret glisse

- Filetage-taraudage manuel : e.g. <https://fr.rs-online.com/web/p/kits-de-tarauds-et-filières/6681129>
- Note : en général sens horaire pour visser...



Concevoir un robot

- Structure et supports mécaniques :
 - Fraiseuse : pour découper de la matière et faire des trous
Nécessite une formation d'1/2 journée pour prise en main, sous conditions
Fixation des pièces pas toujours facile...



Concevoir un robot

■ Structure et supports mécaniques :

- Tour : pour l'usinage de pièces circulaires en plastique, alu, etc.

Pour des raisons de sécurité, nécessite d'être accompagné par un personnel compétent



- Thermoformeuse : pour le moulage de coque de protection, carrosserie ou similaire en plastique. Le moule peut être fait avec une fraiseuse par exemple



■ Structure et supports mécaniques :

- Pour les assemblages rapides :

Serre-câbles/colliers plastique ou métal (aussi appelés Rilsan, Colson, Ty-Rap, Serflex), pince coupante nécessaire pour les retirer

Scratch/velcro : peuvent être en 2 produits séparés : produit avec côté autocollant et côté boucles (e.g. RS 549937) et produit avec côté autocollant et côté crochets (e.g. RS 549921); peuvent être en 1 produit avec côté boucle et côté crochets (pas d'autocollant, e.g. VEL-OW64140); peuvent être en 1 produit avec côté autocollant et côté champignons (fait à la fois office de côté boucles et crochets, e.g. 3M SJ3550)

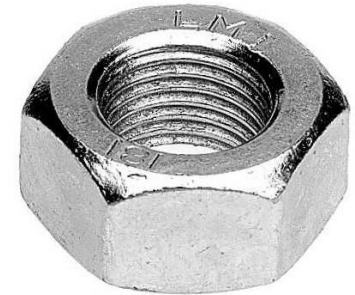
Scotch, double face, colles (attention selon le matériau), etc.

Attention aux scratch/velcro et scotch dont la partie autocollante peut être très adhésive au point qu'on risque d'arracher la matière sur laquelle ils sont collés si on veut les enlever...



- Structure et supports mécaniques :
 - Assemblages :

Écrous standards



Écrous frein/Nylstop (à bague en Nylon bleue ou noire)



Attention à ne pas confondre les écrous standards avec certains écrous indesserrables qui ont un triangle dessiné dessus, le pas peut aussi être inversé, etc.



Voir aussi https://www.ensta-bretagne.fr/lebars/atelier_CNC.pdf

■ Structure et supports mécaniques :

• Outils classiques :

Clés/wrenches

Clé plate/à fourche/Double ended open spanner



Clé mixte/Combination spanner



Clé mixte à cliquet/Ratchet combination spanner



Clé à molette/Adjustable spanner



Clé à pipe/douille/Socket wrench



Clé Allen/Hex wrench



Pinces

Pince multiprise/Adjustable pliers

etc.



- Structure et supports mécaniques :

- Matériaux :

- L'acier inox A4 est amagnétique et adapté au milieu marin, attention cependant aux réactions possibles avec d'autres métaux...



- 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 sur une 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/brosses »** (+ 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 (Electronic Speed Controller) 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 :

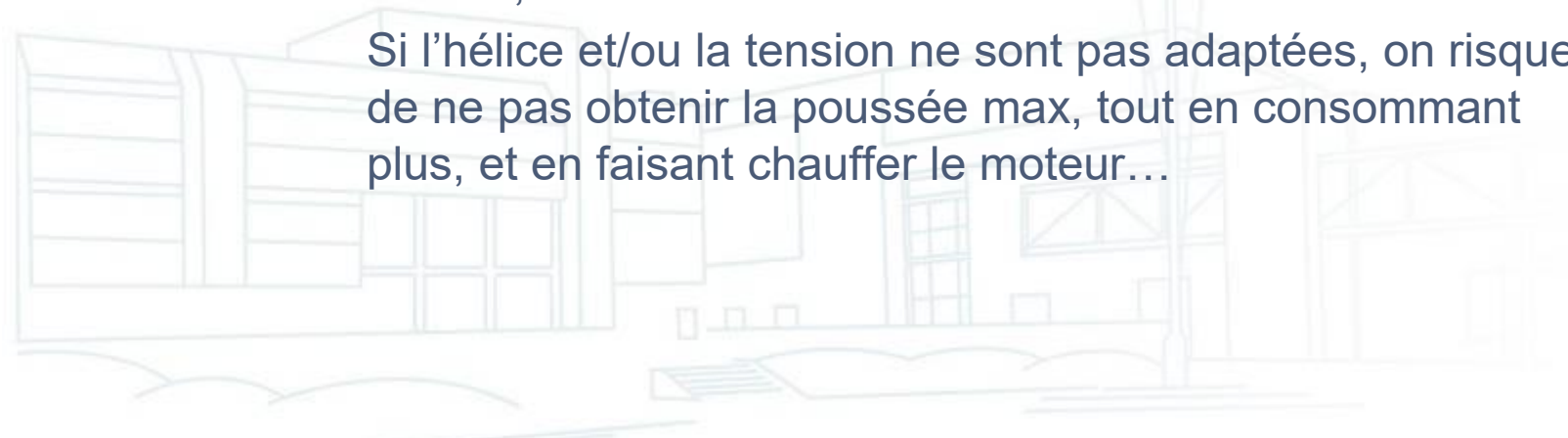
Souvent utilisés en modélisme aérien

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...



Concevoir un robot

■ Actionneurs :

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

Moteurs brushless :



Tendances empiriques observées pour le choix d'hélice :

Motor name	Motor Kv	Motor m (in g)	Motor P (in W)	Propeller	Nb blades	Thrust (in Kgf)	Voltage (in V)	Current (in A)	P (in W)	100*Thrust/P	Comments
MN4014 T-motor	330	149	750	16x5	bi	2.7	6S	25	555	0.486486486	
MN4014 T-motor	330	149	750	MASTER AIRSCREW 11x7	tri	1.1	2S	11	275	0.4	
MK3644/24	490	111	380	MASTER AIRSCREW 11x7	tri	2.1	24.4	21	512.4	0.409836066	
MK3644/24	490	111	380	MASTER AIRSCREW 11x7	tri	1	4S	10	148	0.675675676	Approx
MK3644/24	490	111	380	APC Thin Electric 13x4	bi	2.2	6S	20	444	0.495495495	Approx
Turnigy G15	710	170	575	MASTER AIRSCREW 11x7	tri	2.2	4S -0.9	32	444.8	0.494604317	
Dualsky XM3548EA-5	740	173	671	MASTER AIRSCREW 11x7	tri	2.2	4S -1.2	33	448.8	0.490196078	
Dualsky XM3548EA-5	740	173	671	APC Thin Electric 13x4	bi	2.2	4S -1.2	27	367.2	0.59912854	
Dualsky XM3548EA-5	740	173	671	APC Thin Electric 13x6.5	bi	2.7	4S -0.7	41	578.1	0.467047224	
EMAX GT2826-05	860	175	1000	MASTER AIRSCREW 11x7	tri	2.4	4S -1.4	45	603	0.39800995	
EMAX GT2826-05	860	175	1000	APC Thin Electric 13x4	bi	2.7	4S -1.4	37	495.8	0.544574425	
EMAX GT2826-05	860	175	1000	16x5	bi	4	4S	60	888	0.45045045	
EMAX GT2826-05	860	175	1000	APC Thin Electric 13x6.5	bi	3.1	4S -1.4	54	723.6	0.428413488	

Updated version: https://www.ensta-bretagne.fr/lebars/tests_motors_propellers.csv

■ 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 brushless :

Peut griller partiellement et ne pas pouvoir démarrer à certaines positions...

Les ESC utilisés pour les contrôler les font souvent vibrer de manière à produire des bips indiquant leur état : en général il y a une sonnerie polyphonique de démarrage, et si les ESC reçoivent des commandes valides une sonnerie spécifique suit, sinon les moteurs bippent périodiquement...



■ Actionneurs :

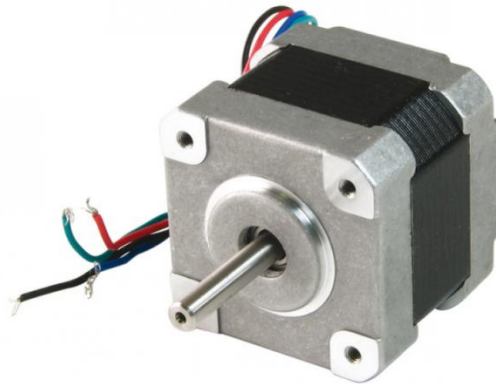
- Moteurs électriques : brushed vs brushless, pas-à-pas ou non?

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 :

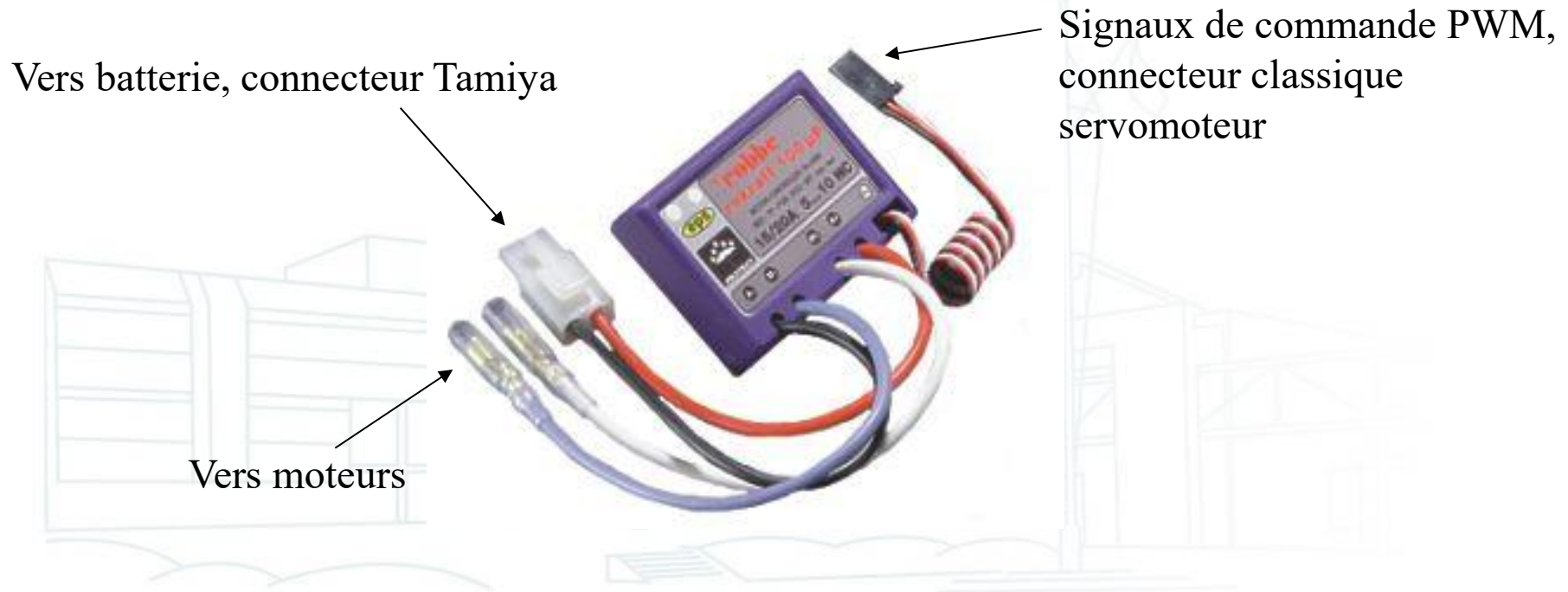
- Carte de puissance/variateur/ESC

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

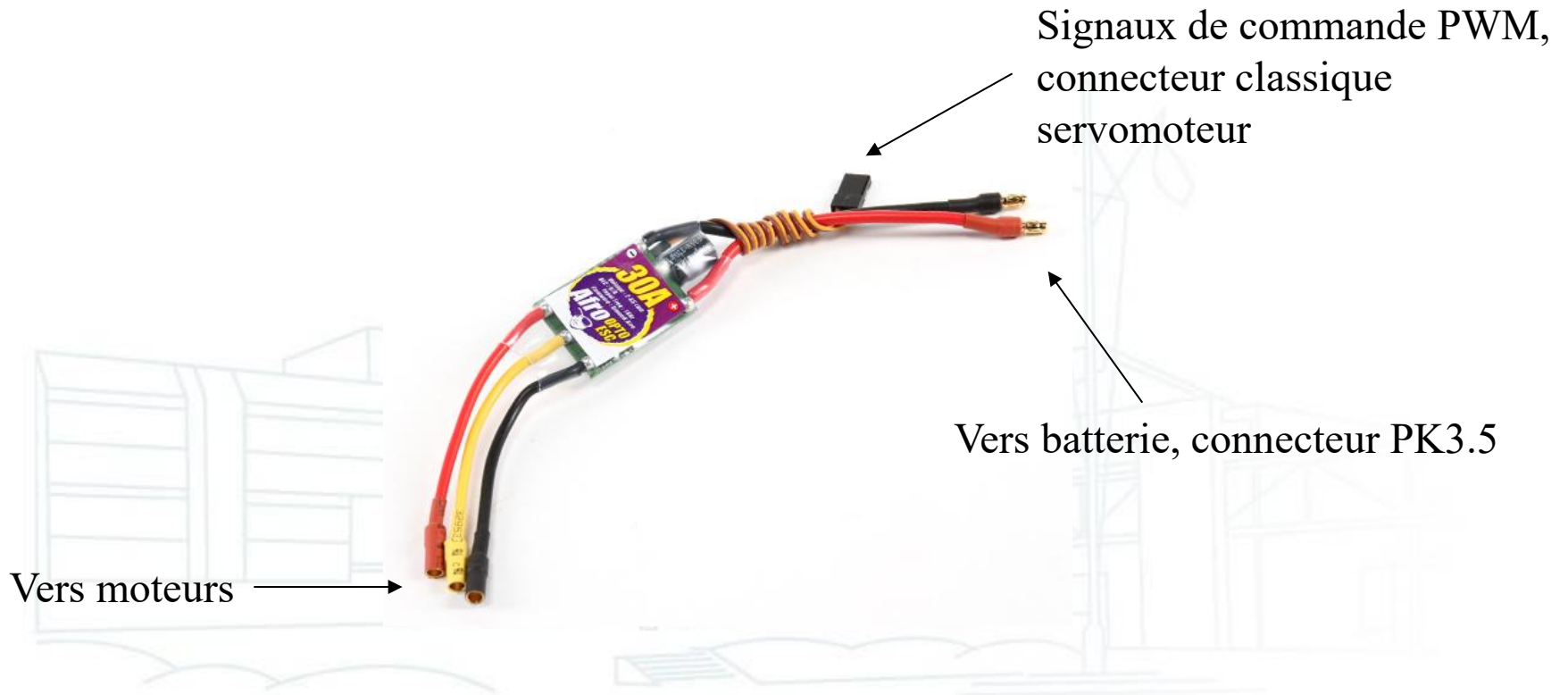
ESC pour moteurs brushless : en modélisme, souvent utilisés pour les avions, quadrirotors, hélicoptères donc il n'y a pas toujours de marche arrière...



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



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

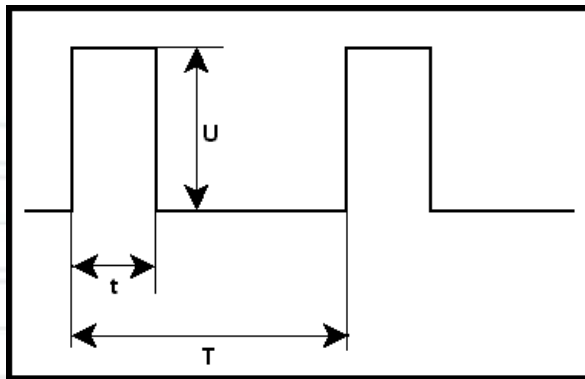


- Actionneurs :
 - Carte de puissance/variateur/ESC

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)

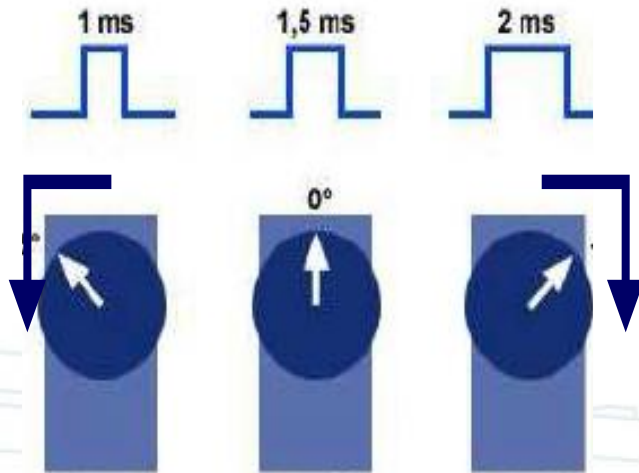
T : période (20 ms)

Note : certains ESC (e.g. pour quadrirotors) peuvent aussi fonctionner avec des PWM plus haute fréquence pour être plus réactifs...

- Actionneurs :
 - Carte de puissance/variateur/ESC

Fonctionnement :

Correspondance largeur d'impulsion / vitesse de rotation



État du moteur	Largeur d'impulsion
Moteur à l'arrêt/neutre	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

Notes :

Certains ESC (e.g. pour drones aériens) peuvent n'avoir qu'une marche avant s'étalant sur la plage entière des 1 à 2 ms (arrêt/neutre à 1 ms).

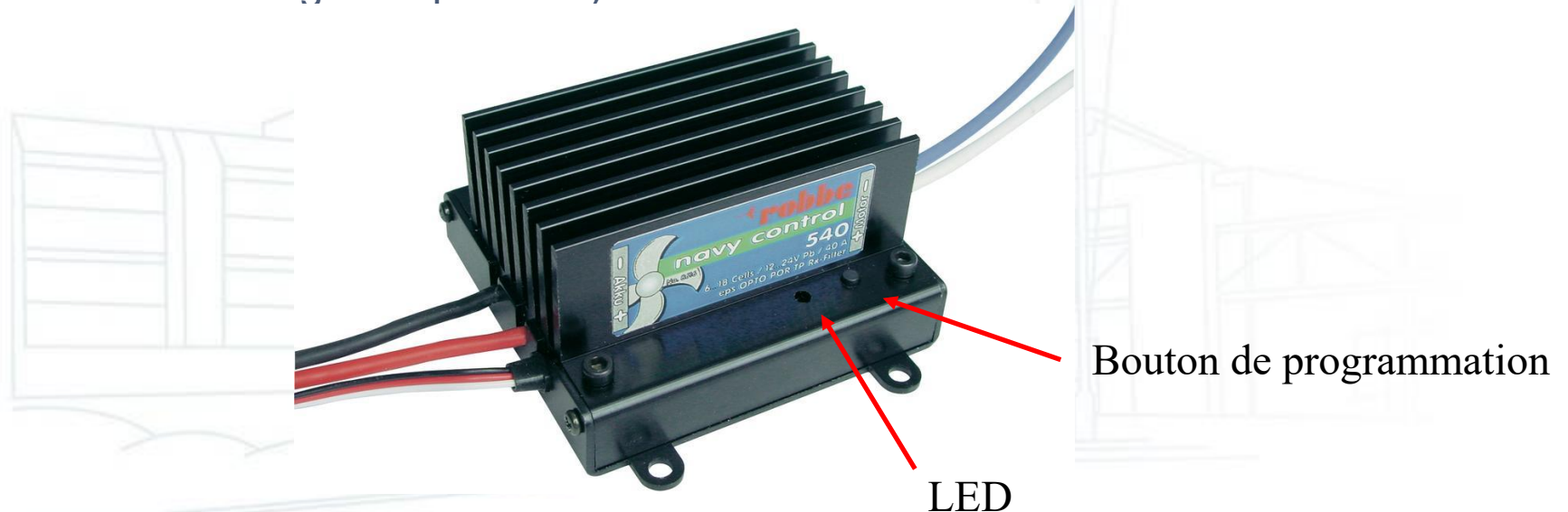
Beaucoup d'ESC refusent de démarrer si le signal de commande ne correspond pas à l'arrêt/neutre lors de la mise sous tension, il faut alors en général leur envoyer les 2 signaux extrêmes puis revenir à l'arrêt/neutre pour les débloquer

ARRET

- Actionneurs :
 - Carte de puissance/variableur/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 (généralisés à partir de la batterie moteurs via un **régulateur/DC-DC convertisseur 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.

Concevoir un robot

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

FAQ BEC et alimentations :

<https://copilot.microsoft.com/shares/wsnB1tQAww5yCyipwfsDp>



■ Actionneurs :

- Carte de puissance/variateur/ESC

Paramètres importants :

Le **courant** supporté par l'ESC doit être $>$ à ce que le **moteur** risque de 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 : Arduino UNO

Se branche sur le port USB du PC et doit contenir un programme communiquant avec les programmes exécutés sur le PC

Peut générer des signaux PWM (**attention : au démarrage de la carte les signaux peuvent être invalides et causer des problèmes si des actionneurs sont sous tension...**), I2C

Peut générer et lire des signaux numériques

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



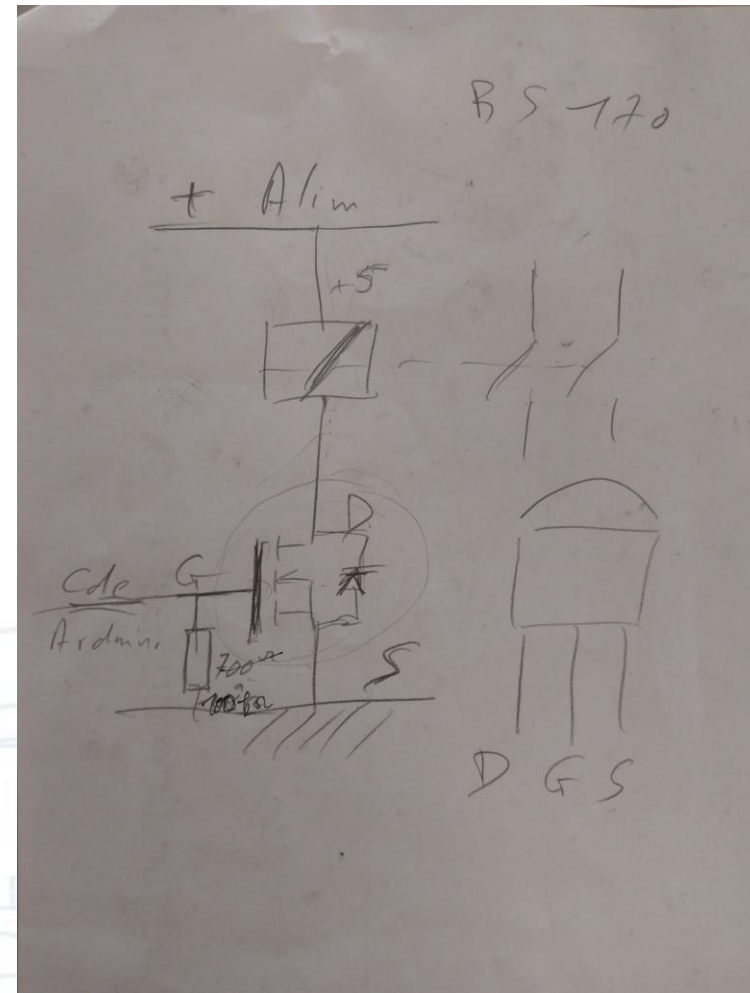
...

- Actionneurs :
 - Carte d'interface

Exemple : Arduino UNO



Sorties numériques pour contrôler des relais : **transistor BS170** nécessaire pour commander **relais AQZ102** (ou la plupart des relais) car les sorties numériques envoient souvent trop peu de courant pour pouvoir le contrôler, et ceci peut être le cas pour d'autres cartes que les Arduino...



■ Actionneurs :

- Carte d'interface

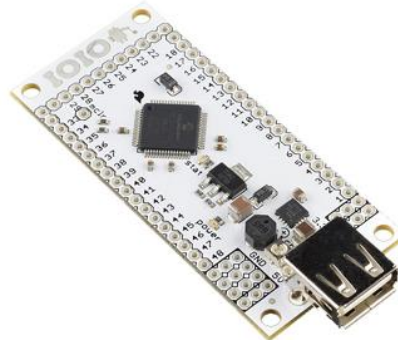
Exemple : carte IOIO pour smartphone/tablette Android, PC
Windows/Linux

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

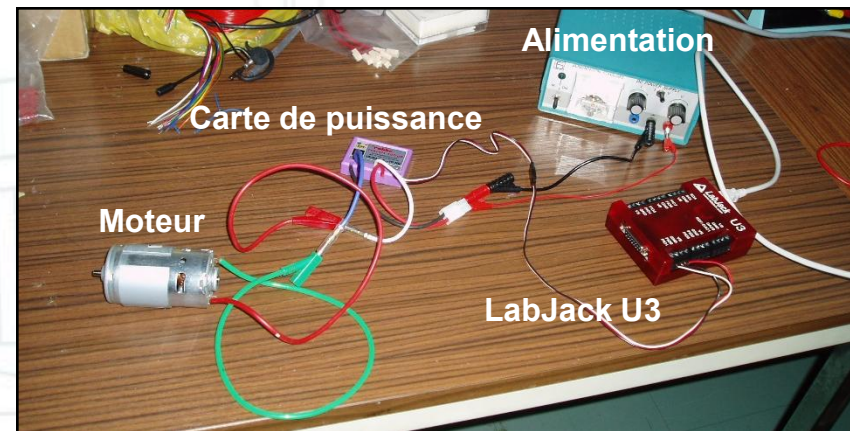
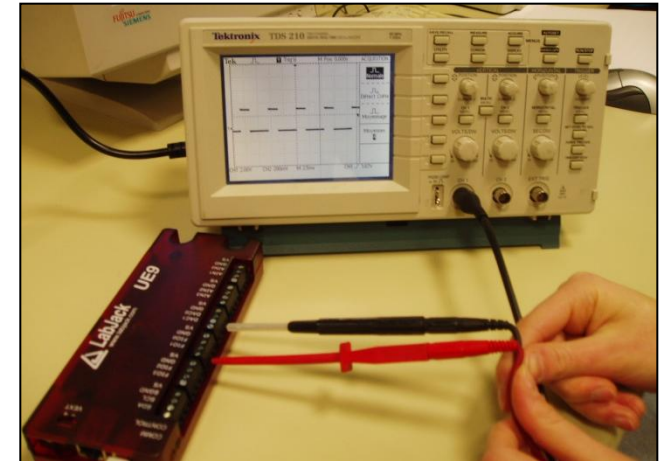
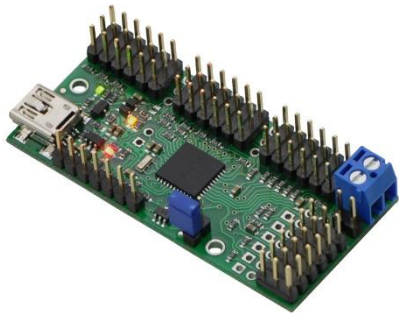
Peut générer et lire des signaux numériques

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



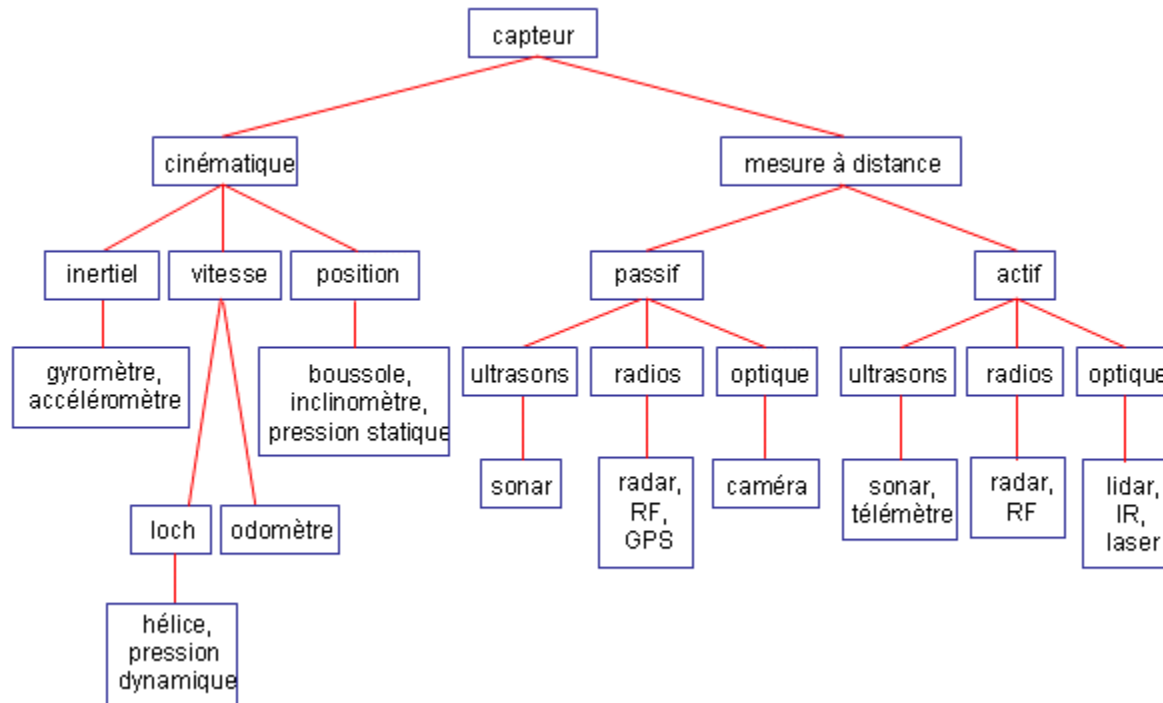
■ Capteurs :

- **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		+		+	

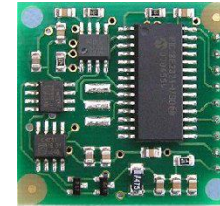
Concevoir un robot

■ Capteurs :



■ Capteurs :

- **Boussole** (compass) : angle w.r.t. 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

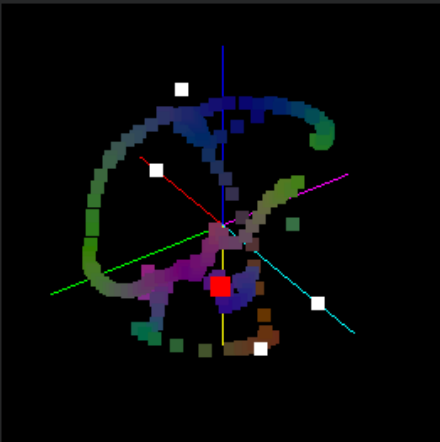
- Capteurs :
 - Calibration magnétique
Bien faire la calibration magnétique en tournant autour de tous les axes, à l'endroit et à l'envers!

Progress

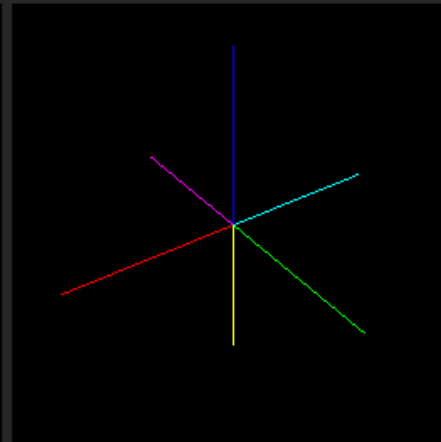
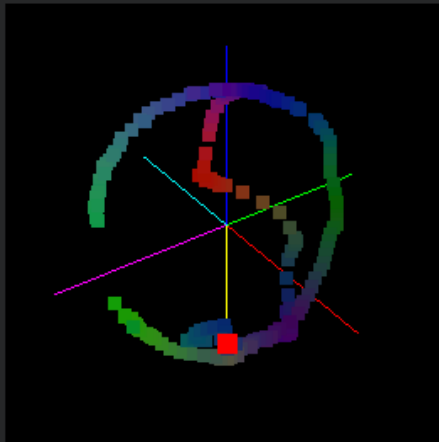
Got + 679 samples
Compass 1 error: 0.23
Compass 2 error: 0.27
Compass 3 error: 99
more data needed Aim For Yellow-Green

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

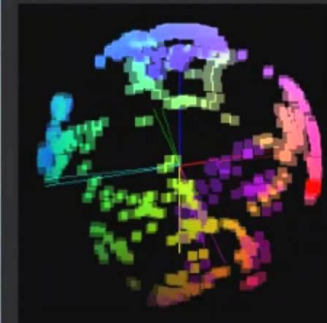


Done

COMPASS CALIBRATION
Compass calibration is important for accurate heading. Point every axis north, and rotate full 360 around the left-right axis.

Live Calibration

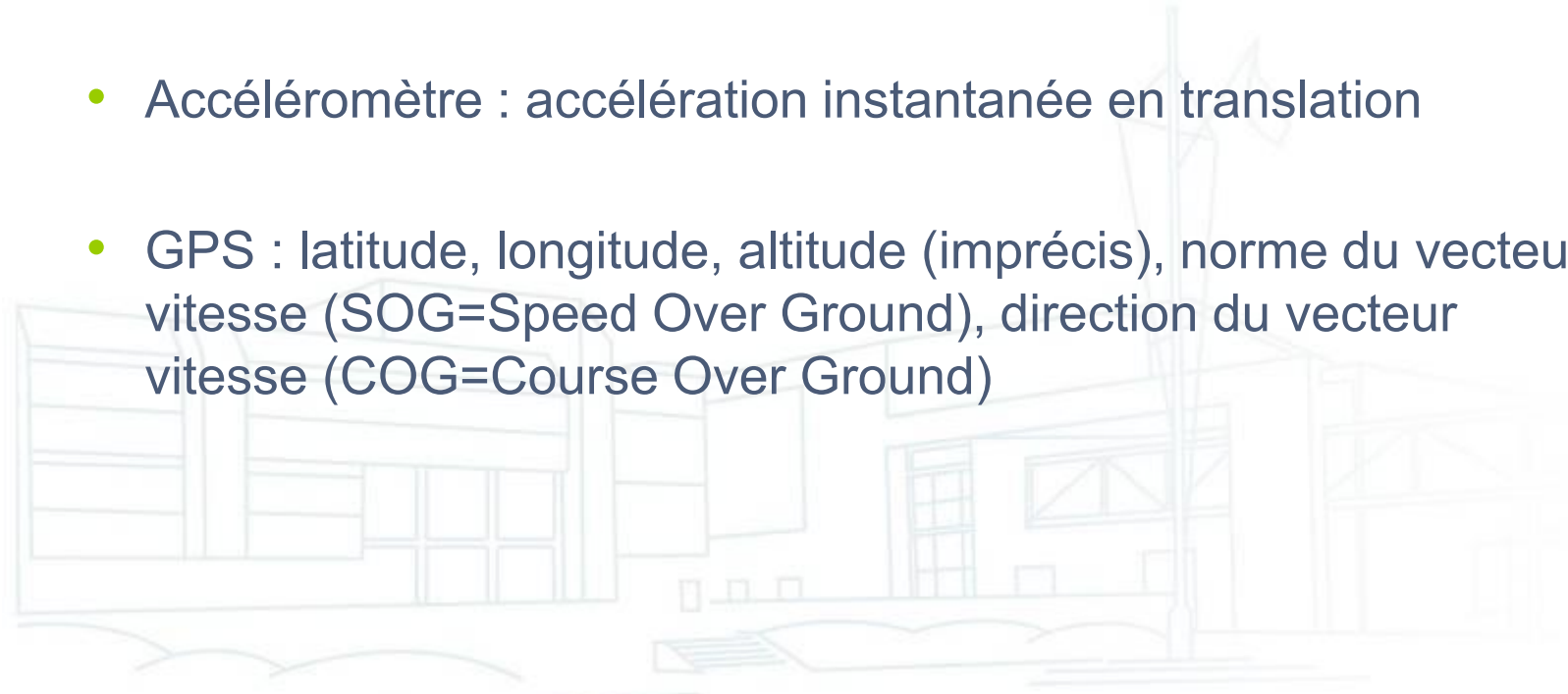
Got 472 Samples



Rotate with each data point



- Capteurs :
 - Inclinomètre : angle par rapport à l'horizontale
 - Gyromètre : vitesse instantanée de rotation
 - Accéléromètre : accélération instantanée en translation
 - GPS : latitude, longitude, altitude (imprécis), norme du vecteur vitesse (SOG=Speed Over Ground), direction du vecteur vitesse (COG=Course Over Ground)



■ Capteurs :

- GPS

Multi constellation : en fait on ne devrait plus dire « GPS » mais « **GNSS** » (**Global Navigation Satellite System**) car la plupart des modules récents supportent plusieurs types de satellites : GPS (USA), GLONASS (russe), Galileo (européen), BeiDou (chinois), etc.

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, ou directement un GPS double antenne

Mode RTK : « base » (désigne le module GPS de référence, souvent fixe) proche (<10 km, lié à zone atmosphérique) qui envoie des corrections (avec le protocole RTCM, ou à travers un serveur NTRIP) au « rover » (désigne le module GPS mobile) pour améliorer estimations de position relative (ou absolue si base de position bien connue), peut être utilisé pour estimer cap si base mobile avec le rover à plus d'1 m sur le même robot



■ 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

- On appelle souvent ces regroupements de capteurs « centrale inertielle »



■ Capteurs :

- Consulter la documentation pour voir les différentes calibrations possibles/nécessaires, et vérifier leurs résultats
- 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/Share/RI_magnetometer.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 (ou autres périphériques) :

- Types de communication :

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

Sorties numériques :

Point à point :

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, BlueTooth, ZigBee, LoRa, some acoustic modems

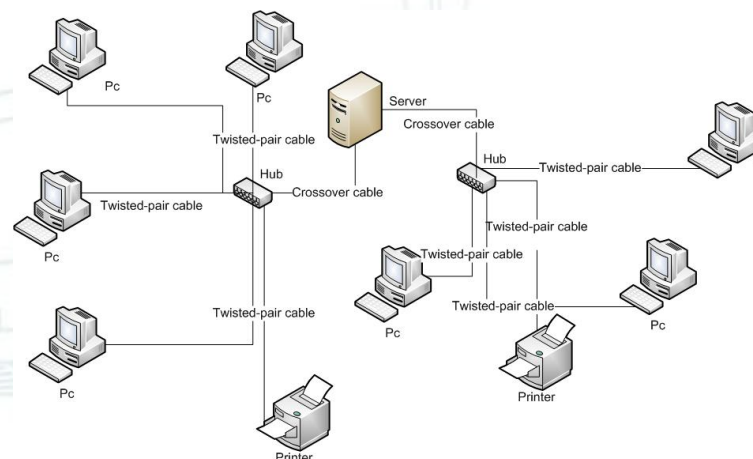
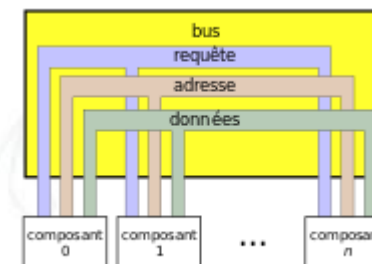
■ Capteurs (ou autres périphériques) :

- 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

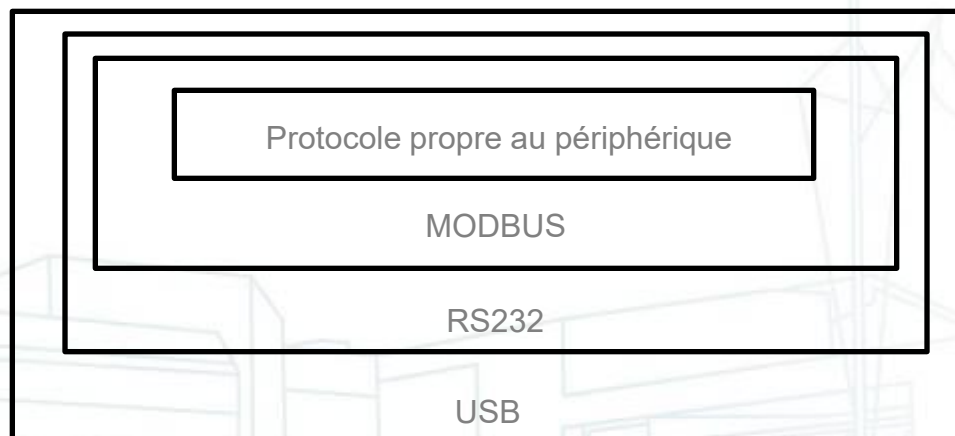


■ Capteurs (ou autres périphériques) :

- Types de communication :

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

Exemple : capteur de pression Keller



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

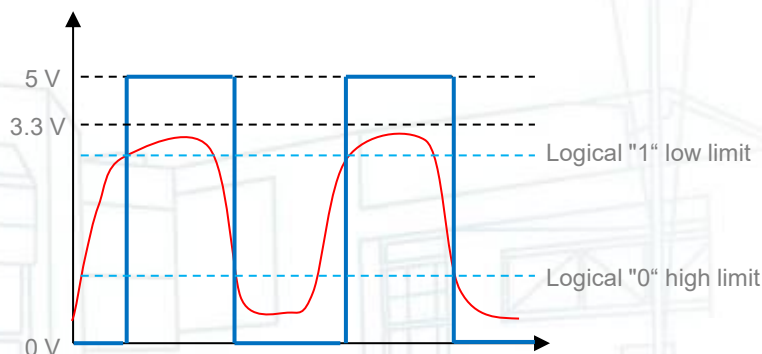
■ Capteurs (ou autres périphériques) :

- Types de communication :

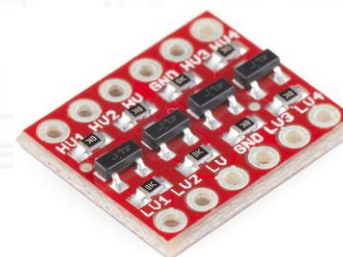
Attention : la tension max d'alimentation du capteur peut être différente de la tension max sur ses pins de communication!

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

Certains peuvent aussi être « +3.3 V - +5 V-tolerant »



Voir <https://learn.sparkfun.com/tutorials/bi-directional-logic-level-converter-hookup-guide> si besoin de conversion 3.3-5 V pour des signaux numériques



■ Capteurs (ou autres périphériques) :

- Types de communication :

Liaison série RS232/UART

A l'origine, port DB9 directement sur PC

Maintenant, via convertisseurs USB

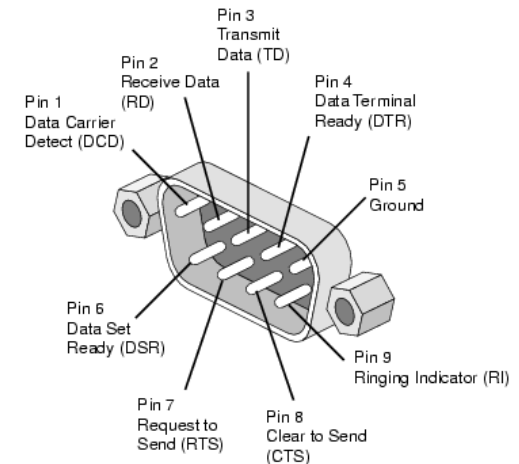
Niveaux de tensions standards PC : -12..+12 V

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>

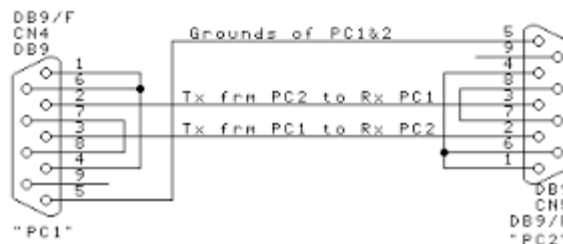
■ Capteurs (ou autres périphériques) :

- 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 (ou autres périphériques) :

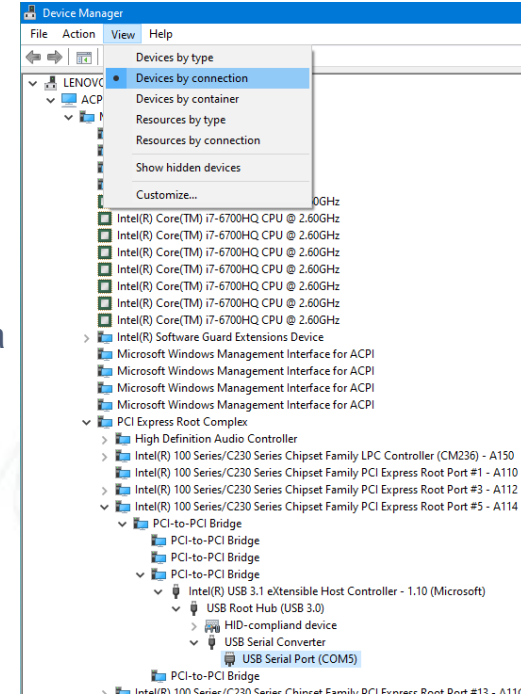
- Types de communication :

Liaison série RS232/UART

Port DB9 directement sur PC ou pins spécifiques : consulter la documentation du PC pour connaître la correspondance entre port physique et nom attribué (e.g. **COM1** sous Windows, **/dev/ttyS0** sous Linux, etc.)

Convertisseurs USB-RS232 : en les déconnectant puis reconnectant, on peut déterminer le nom attribué dans le **Gestionnaire de périphériques** sous Windows, idem avec la commande **sudo dmesg** sous Linux

Il se peut que des drivers doivent être installés



```
user@THINKPAD-T420: ~/Desktop
[ 17.259770] Bluetooth: RFCOMM TTY layer initialized
[ 17.259789] Bluetooth: RFCOMM socket layer initialized
[ 17.259805] Bluetooth: RFCOMM ver 1.11
[ 17.588060] nouveau 0000:01:00.0: [drm] Cannot find any crtc or sizes
[ 17.792061] nouveau 0000:01:00.0: [drm] Cannot find any crtc or sizes
[ 20.006746] rtkill: input handler dsabled
[ 3912.759853] usb 2-1.1: new full-speed USB device number 5 using ehci-pci
[ 3912.875324] usb 2-1.1: New USB device found, idVendor=0403, idProduct=6001, bcdDevice= 6.00
[ 3912.875347] usb 2-1.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 3912.875354] usb 2-1.1: Product: TTL232R-3V3
[ 3912.875359] usb 2-1.1: Manufacturer: FTDI
[ 3912.875363] usb 2-1.1: SerialNumber: FT64F7D
[ 3912.897296] usbcore: registered new interface driver usbserial_generic
[ 3912.897315] usbserial: USB Serial support registered for generic
[ 3912.902491] usbcore: registered new interface driver ftdi_sio
[ 3912.902511] usbserial: USB Serial support registered for FTDI USB Serial Device
[ 3912.902624] ftdi_sio 2-1.1.0: FTDI USB Serial Device converter detected
[ 3912.902676] usb 2-1.1: Detected FT232R
[ 3912.903936] usb 2-1.1: FTDI USB Serial Device converter now attached to ttyUSB0
user@THINKPAD-T420: ~/Desktop$
```

Le chemin complet e.g. **/dev/ttyUSB0** doit souvent être spécifié dans les applications



■ Capteurs (ou autres périphériques) :

- 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 <https://www.ensta-bretagne.fr/lebars/Share/hypertrm.zip>), **Realterm**, ou **minicom**, **gtkterm** (Linux)

Attention à la configuration du caractère de fin de ligne si nécessaire...

Voir aussi **Virtual Serial Port Emulator**, **com0com**, **socat** pour certaines manipulations avec des ports série

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

Concevoir un robot

■ Capteurs (ou autres périphériques) :

- Types de communication :

Liaison série RS232/UART

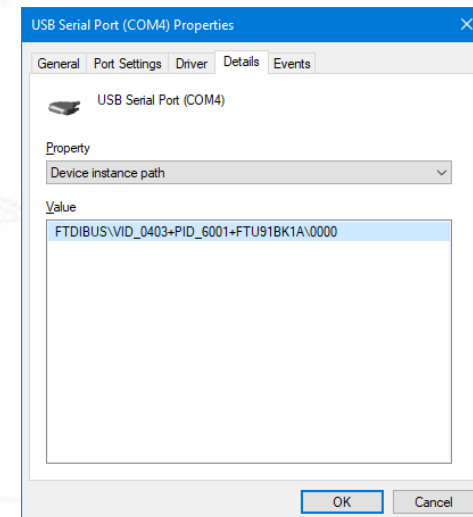
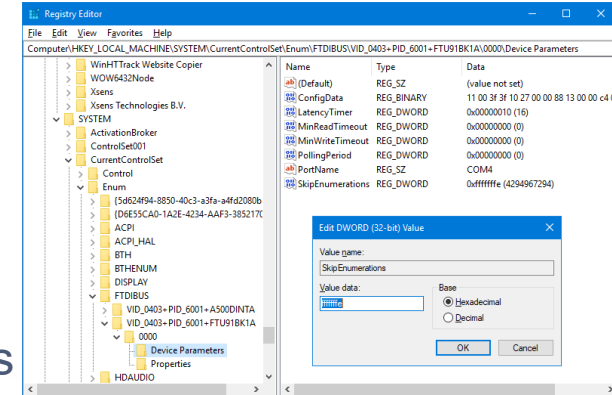


Attention : certains périphériques envoyant des données en continu peuvent être parfois pris pour une souris au démarrage de **Windows!** Voir

<http://support.microsoft.com/kb/283063> pour résoudre ce problème, e.g. modifier la valeur de registre **DWORD SkipEnumerations** à

FFFFFFFFE dans

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\FTDIBUS\VID_0403+PID_6001+FT0CC2ZSA\0000\Device Parameters (remplacer FTDIBUS\VID_0403+PID_6001+FT0CC2ZSA\0000 par les identifiants de votre périphérique, qu'on peut obtenir dans le **Gestionnaire de périphériques**)



- Capteurs (ou autres périphériques) :

- Types de communication :

Liaison série RS232/UART



Attention : certains périphériques peuvent être parfois perturbés par **brlty** (gestionnaire de périphériques braille) notamment sous Ubuntu 22.04

Après débranchement-rebranchement puis consultation des derniers messages de la commande **sudo dmesg** pour voir si brlty est évoqué, on peut par exemple le désinstaller avec **sudo apt remove brlty**



- Capteurs (ou autres périphériques) :

- 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 (ou autres périphériques) :

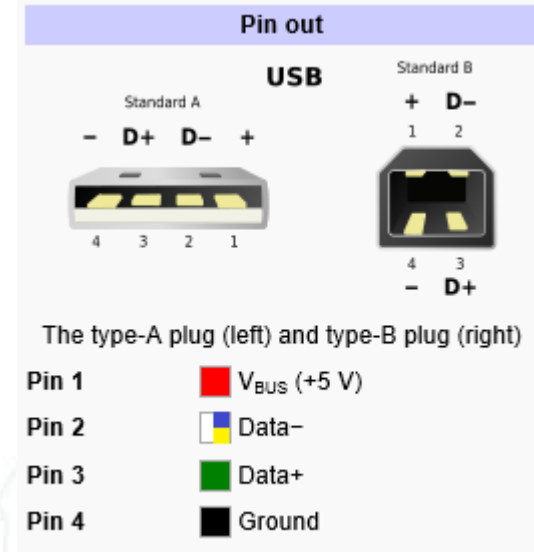
- Types de communication :

Bus USB 2.0

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

Grand nombre de **connecteurs différents**

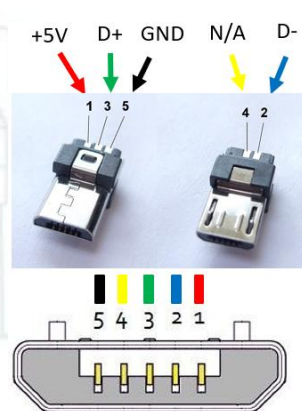
Utiliser des **câbles de qualité** (blindé, etc.), éviter de couper les câbles et de passer par des connecteurs non-standards, signal très sensible en pratique...



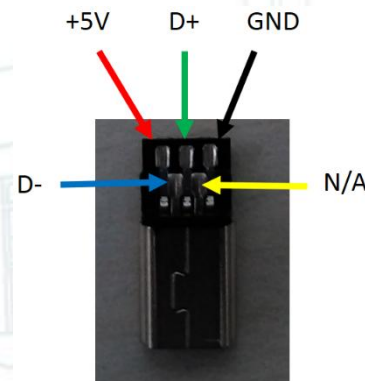
USB Plug (Male):



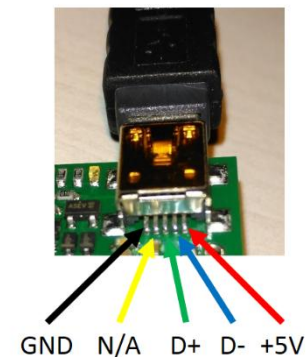
USB Port (Female):



Adafruit USB DIY Connector Shell Type Micro-B Plug #1390



Adafruit Slim MiniB DIY #1825



Mini USB port on Pololu Micro Maestro 6

- Capteurs (ou autres périphériques) :

- 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 (ou plus et même en d'autres tensions que 5 V pour certains ports USB supportant e.g. QuickCharge, USB Power Delivery)

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

- Capteurs (ou autres périphériques) :
 - Types de communication :
 - Bus USB 2.0

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

Attention : certains PC ont parfois 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)! Par exemple, on observe souvent que plusieurs webcams sur un même hub USB ne fonctionnent pas...

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

■ Capteurs (ou autres périphériques) :

- 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 (ou autres périphériques) :

- Types de communication :

Bus I2C/SMBus

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

Sous Linux, un driver, des commandes, des API (Application Programming Interface) permettent d'utiliser ce bus si présent sur l'ordinateur



- Capteurs (ou autres périphériques) :

- Types de communication :

Numéros/noms 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!

■ Capteurs (ou autres périphériques) :

- 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 ou multiprocess**

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/paquet/message, qui ne commence pas forcément par le début des données récupérées!

Certains capteurs renvoient des données tout le temps, d'autres uniquement sur demande. Protocoles courants : **NMEA**, **MODBUS**, **MAVLink**, mais en pratique le plus souvent le protocole est spécifique au constructeur...

Bien vérifier comment interpréter un octet (**binaire vs texte**, voir e.g. encodage **ASCII**), un paquet (**endianness**), interprétation du **decimal separator**...

Voir aussi https://www.ensta-bretagne.fr/lebars/initiation_reseaux.pdf

■ Capteurs (ou autres périphériques) :

- Programmation :

Commande des actionneurs à partir de données brutes de capteurs :
Le **bruit** (erreurs aléatoires de mesure) risque de provoquer des micro-mouvements à haute fréquence dans les actionneurs, ce qui pourrait les **user**

Filtres (e.g. https://en.wikipedia.org/wiki/Moving_average), **seuils**, **diminuer la fréquence de rafraîchissement** peuvent limiter ces effets, avec pour inconvénients potentiels de diminuer la précision de l'actionneur, rajouter des retards, etc.

```
if (abs(new_value-last_applied_value) < threshold) {  
    // Leave the actuator at last_applied_value...  
}  
else {  
    set_actuator(new_value);  
    last_applied_value = new_value;  
}
```

- Capteurs (ou autres périphériques) :

- Programmation :

Actionneurs capables de signaler une erreur mais dont on sait qu'on ne pourra pas corriger l'erreur s'il y en a une signalée (e.g. pas de possibilité d'intervenir sur le robot en cours de mission pour remplacer l'actionneur défectueux ou se reconfigurer pour faire sans) :

Il vaut mieux peut-être ignorer les signalements d'erreur pour maximiser les chances de réussir la mission en cas de faux positif ou erreur sans conséquences réelles (erreurs signalées alors qu'en réalité l'actionneur réalise bien l'action voulue), pour minimiser le risque de bug dans la gestion des erreurs, etc.



■ 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

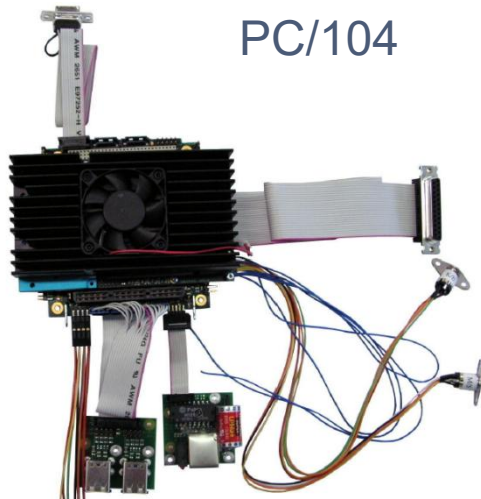
- Ordinateur embarqué :
 - Exemples : PC i.e. ordinateur avec processeur compatible Intel x86 avec BIOS/UEFI, etc.

HTPC (Home Theater PC)

Netbook/convertible

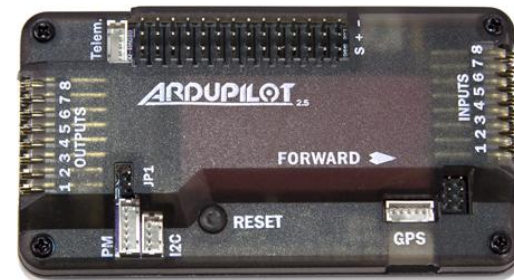
Mini ITX

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

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



Risques classiques dans le cas de l'alimentation d'une Arduino UNO : voir <https://copilot.microsoft.com/shares/FxqUM4AhwpQWnUpPPAD64>

Concevoir un robot

- 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!

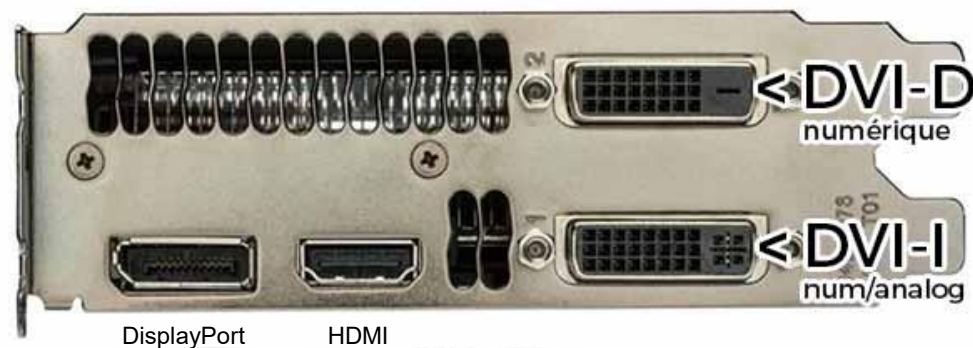
e.g. Raspberry Pi Zero (5\$) + hub USB (5\$) + adaptateur micro USB (1\$) + adaptateur mini HDMI (5\$) + clé Wi-Fi (5\$) + clavier/souris (5\$)...



Concevoir un robot

■ Ordinateur embarqué :

- Différents ports et câbles pour écran : VGA, HDMI, DVI, DisplayPort, Thunderbolt à ne pas confondre avec certains ports USB (cependant certains ports USB Type C peuvent aussi gérer plusieurs usages)...



Standard HDMI Port	Mini HDMI Port	Micro HDMI Port
		
Width: 14mm Height: 4.5mm Used in: TV, Computer	Width: 10.5mm Height: 2.5mm Used in: PC, Camera	Width: 6mm Height: 2.3mm Used in: PC, Mobile Phone

■ Autopilote :

- Un **autopilote** (appelé aussi souvent **flight controller**, **flight management unit**, etc.) contient en général une ou plusieurs **centrales inertielles** (pour la **redondance** en cas de problème sur l'une), de nombreuses **entrées et sorties PWM** (souvent entrées PPM ou SBUS (sérialisation de PWM) pour lire ce qui vient d'un récepteur, sorties PWM pour commander des servomoteurs ou parfois PWM haute fréquence pour les ESC de moteurs), **numériques** (ports série, I2C, etc.) et parfois aussi analogiques, ainsi qu'un **processeur** et **carte micro SD** (pour enregistrer logs)



■ Autopilote :



- Il existe plusieurs « familles » d'autopilotes, celle que l'on utilise à l'école est compatible avec les **firmware ArduPilot** (qui se déclinent en **ArduPlane**, **ArduCopter**, **ArduRover**, **ArduSub**, etc., voir <http://ardupilot.org>) et PX4. Il existe plusieurs versions matérielles officielles de cette famille d'autopilotes, e.g. APM2.5, Pixhawk, Pixhawk 4 Mini, etc. Et comme ces versions matérielles sont open source, il existe de nombreuses variantes pour chaque version matérielle, e.g. :



APM2.5, parfois aussi vendu sous le nom ArduPilot car sans doute le premier à le supporter (et par abus de langage, on appelle souvent ArduPilot les autopilotes compatibles avec les firmware ArduPilot...) = **HKPilot Mega** = **ArduFlyer**



Pixhawk (FMUv2) = HKPilot32 = Dropix v2

Pixhawk 2 (FMUv3)

Pixracer (FMUv4)

Pixhawk 3 Pro (FMUv4-PRO)

Pixhawk 4 Mini (FMUv5)



- Autopilote :
 - Pratiquement tous les multicopters (quadrirotor, hexarotor, octorotor, etc.) sont vendus avec un autopilote, la plupart étant propres à la marque ou au modèle. En effet, contrôler manuellement la vitesse des 4 (ou plus) moteurs n'est pas évident, d'où le besoin d'aides à la stabilisation au moins en roulis et tangage

DJI A3 de DJI Matrice 600



SwellPro S3 de SwellPro Splash Drone 3



■ Autopilote :

- Il y a des concepts et du vocabulaire particuliers qu'on retrouve souvent quand on utilise un autopilote

Arm and disarm: autoriser ou désactiver la génération de signaux de commandes (nécessairement au neutre au départ) par l'autopilote vers les ESCs des moteurs

Arming checks: vérifications faites par l'autopilote avant d'autoriser l'armement, e.g. état de la centrale inertielle, qualité du GPS, qualité de réception de la télécommande, état du bouton d'armement sur le drone, etc.

RTL/RTH: Return To Launch/Home est une fonction de retour automatique au point de départ. Il faut bien lire la documentation du drone pour comprendre ce qu'il considère comme position de départ! En général, c'est la position du dernier armement, mais si le drone ne peut pas estimer sa position il se peut qu'il ne fasse qu'atterrir sur place... Il y a parfois aussi des modes **SmartRTL** qui cherchent à revenir en suivant le même parcours qu'à l'aller, ce qui est utile en particulier pour les drones n'ayant pas de capteurs pour détecter les obstacles

Failsafe: Ce sont des actions pouvant être activées lorsqu'un problème spécifique est détecté en vol par l'autopilote (voir aussi le concept de watchdog : https://en.wikipedia.org/wiki/Watchdog_timer), le problème le plus classique étant la batterie faible ou la perte de communication avec la télécommande. Il faut donc bien lire la documentation du drone pour comprendre ce qu'il va faire, en général on peut configurer un RTL en cas de batterie faible et un atterrissage immédiat en cas de batterie très faible. Il y a souvent possibilité d'interrompre l'action d'un failsafe en forçant un changement vers un autre mode de vol puis en revenant à un mode de contrôle manuel

Geo fencing: mettre en place une barrière virtuelle que le drone n'a pas le droit de franchir

Voir aussi: <https://ardupilot.org/planner/docs/common-glossary.html>

- Autopilote :
 - Exemple de construction de drone avec autopilote compatible **ArduCopter** contrôlé via **Raspberry Pi** avec le protocole **MAVLink** : https://www.ensta-bretagne.fr/lebars/tutorials/TD_ARDU_PARROT.pdf



■ Communication :

- Commande des actionneurs

Récepteur de modélisme côté robot, télécommande côté opérateur

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

Typiquement 4-8 PWM

Fréquences classiques : actuellement 2.4 GHz, 868 MHz, 433 MHz, anciennement 27, 35, 40, 41 MHz

Communication en général uniquement dans le sens télécommande vers récepteur



■ Communication :

- Commande des actionneurs

Récepteur de modélisme côté robot, télécommande côté opérateur

Télécommandes configurables (e.g. FrSky Taranis QX7 ACCESS, Turnigy 9XR Pro)

Bien vérifier la **batterie** (tension, forme, type, connecteur, etc.) qui peut être mis dedans

En fonction de la batterie, il faut régler dans les paramètres de la télécommande l'**intervalle de tension possible** et le **seuil d'alarme**, si la télécommande a une fonction de charge, s'assurer qu'elle est compatible avec le type de batterie installé avant de l'utiliser

Certaines télécommandes modernes (e.g. FrSky Taranis QX7 ACCESS) peuvent parfois être branchées en **USB** sur un PC et être reconnues comme un **joystick, tout en chargeant la batterie**

Elles ont un **firmware** qui peut parfois être mis à jour via carte micro SD en navigant dans les menus correspondants

Concevoir un robot

■ Communication :

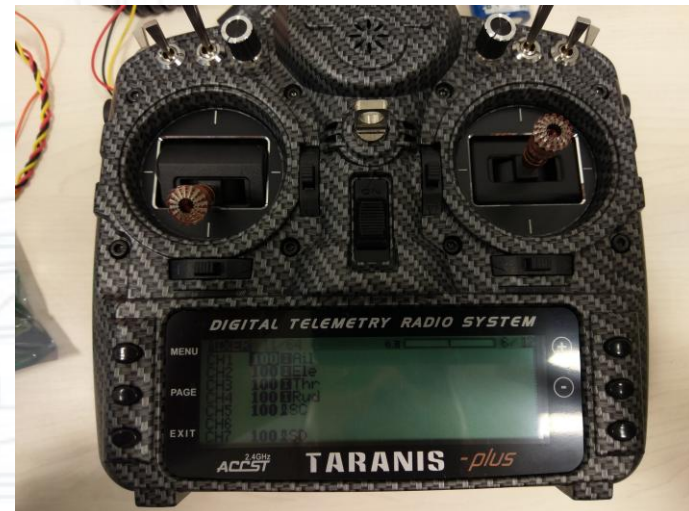
- Commande des actionneurs

Récepteur de modélisme côté robot,
télécommande côté opérateur

Télécommandes configurables (e.g. FrSky
Taranis X9D, Turnigy 9XR Pro)

La navigation dans ses **menus de configuration** est en général basée sur des **appuis courts, longs ou simultanés de boutons**, éventuellement dès la mise sous tension. Il y a en général une section liée aux **paramètres globaux** de la télécommande (e.g. alarme batterie), une section pour créer des « profils » (souvent appelés « **MODELS** ») pour certains paramètres particuliers (e.g. **mixage des canaux**)

Les sticks sont souvent appelés Throttle, Rudder, Ailerons, Elevons dans les menus. Les autres interrupteurs et potentiomètres ont souvent leurs noms inscrits à proximité



Concevoir un robot

■ Communication :

- Commande des actionneurs

Récepteur de modélisme côté robot,
télécommande côté opérateur

Télécommandes configurables (e.g. FrSky Taranis X9D, Turnigy 9XR Pro)

Elles peuvent parfois avoir un **émetteur interne** et/ou une **baie JR pour mettre un émetteur externe** à acheter séparément. Ils ont souvent des paramètres devant être réglés dans les menus (e.g. liés au protocole de communication entre la télécommande et l'émetteur s'il est externe (e.g. PPM), paramètres de configuration de l'émetteur interne (e.g. protocole de communication sans fil avec le récepteur)). Le firmware de l'un ou l'autre peut/doit parfois être mis à jour via une carte micro SD dans la télécommande. Les émetteurs externes peuvent parfois avoir leurs propre écran et boutons de configuration



Télécommande Turnigy 9XR Pro avec un émetteur externe Immersion RC EzUHF 433 MHz dans sa baie JR



Emetteur externe TBS Crossfire 868 MHz dans la baie JR d'une télécommande. Il est muni d'un écran et d'un bouton-joystick servant à sa configuration ainsi qu'à celle des récepteurs connectés

■ Communication :

- Commande des actionneurs

Récepteur de modélisme côté robot,
télécommande côté opérateur

Télécommandes configurables (e.g. FrSky Taranis X9D, Turnigy 9XR Pro)

Un **récepteur compatible** doit être choisi: il faut s'assurer que sa fréquence et son **protocole de communication sans fil** (e.g. FrsKy ACCESS, ACCST D16, ACCST D8, attention aux différences entre versions USA (FCC) et EU, anciennement FM, AM, SSS, SPCM, PPM avec nécessité de choisir des quartz pour choisir un canal de fréquence...) est compatible avec l'émetteur et sa **configuration**, que les **sorties du récepteur** correspondent bien à ce qu'on veut contrôler sur le robot (e.g. **PWM** pour des servos ou ESCs standards, **PPM** ou **SBUS** pour communiquer avec un autopilote (il se peut que des **jumpers** soient à installer, que le **firmware** doive être changé (ce qui peut parfois être fait via la télécommande, des **scripts LUA** sont aussi parfois utilisés pour changer des paramètres), interrupteurs à bouger, etc.), parfois il y a aussi de la télémétrie qui peut être transmise vers la télécommande voire un PC connecté à la télécommande (e.g. via Bluetooth pour les TBS Crossfire))



Mise à jour d'un récepteur FrSky X8R avec une télécommande FrSky Taranis QX7 ACCESS contenant une carte SD avec le bon firmware

Voir aussi: <https://oscarliang.com/rc-protocols/>

■ Communication :

• Commande des actionneurs

Récepteur de modélisme côté robot, télécommande côté opérateur

Télécommandes configurables (e.g. FrSky Taranis X9D, Turnigy 9XR Pro)

Une procédure de « **bind/appairage** » doit en général être suivie pour associer un récepteur avec un émetteur (en général appui d'un bouton spécifique à la mise sous tension, jumper, sélection de la fonction parfois notée « Bnd » dans un menu)

A noter qu'il faut parfois **éloigner la télécommande du récepteur** pour qu'ils puissent communiquer correctement, redémarrer la télécommande et le récepteur en cas d'échec, éviter que d'autres télécommandes soient allumées à proximité...

Un récepteur peut avoir ses propres paramètres de **failsafe** en cas de perte de communication avec l'émetteur, e.g. ne plus envoyer de PWM ou envoyer des PWM prédéfinis, souvent réglés en appuyant un bouton ou via un paramètre de configuration



Exemples de quartz FM 41 MHz, un pour l'émetteur (TX) et l'autre pour le récepteur (RX). Une procédure de bind remplace maintenant l'utilisation des quartz pour associer un émetteur à un récepteur

■ 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, 868 MHz, 433 MHz
 - Communication en général bidirectionnelle



- 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
 - Communication en général uniquement dans le sens caméra vers écran/lunettes



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 :

Attention aux fréquences et puissances d'émission autorisées, qui peuvent être différentes selon qu'on soit en UE ou USA (e.g. l'équivalent de la bande classique 868 MHz est 915 MHz aux USA)

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 pour savoir laquelle durera plus longtemps

- Courant max de charge/décharge

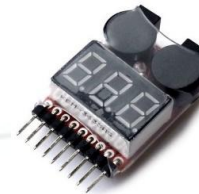
e.g. 30C/5C : si e.g. C=2500 mAh, le courant max de décharge serait 30*2500 mAh et charge de 5*2500 mAh



■ 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 (nominal (pour stockage long), 4.2 V si bien chargé) pour Li-Po, 1.2 V (nominal, 1.5 V si bien chargé) pour NiMH

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



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

- Voir https://www.ensta-bretagne.fr/lebars/robots_batteries.pdf

Concevoir un robot

■ 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 : précision parfois limitée (e.g. 0.2 V d'erreur)...
- En particulier, les **Li-Po** constituées de plusieurs éléments ont souvent un **connecteur d'équilibrage** qui permet au chargeur de réajuster chaque élément indépendamment
- BMS (Battery Management System) : circuit de protection parfois intégré dans des packs de batteries
- **Ecrasement, surcharge, court-circuit = danger!**
<https://bluerobotics.com/learn/battery-info/>
<https://youtu.be/-DcpANRFrI4&t=12>

- Energie :
 - **Bien faire attention à ne pas créer un court-circuit dans la batterie notamment au moment du branchement de câbles adaptateurs sur les chargeurs, en particulier brancher d'abord les câbles au chargeur et en dernier la batterie pour minimiser les risques que les fiches bananes se touchent!**
 - **Si les adaptateurs sont avec de multiples connecteurs, vérifier qu'ils ne risquent pas d'être court-circuités facilement!**



Concevoir un robot



Concevoir un robot

■ Energie :



- Panneau solaire en moyenne sur 1 an (comprenant jours, nuits, temps couverts, ensoleillés, etc.)

Panneau solaire propre et bien orienté à Kiel, Allemagne : **20 W/m²** (d'après https://www.tf.uni-kiel.de/matwis/amat/iss/kap_8/advanced/a8_3_1.html)

Panneau solaire en mer (sur bouée flottante, pas forcément bien orienté, sale, avec de l'eau dessus...) à Brest, France : **1 W/m²** (info issue de discussions orales)



■ Energie :

- **Brownout** : problème de chute transitoire de l'alimentation. Ceci peut se produire au niveau de batteries ou régulateurs qui ont été mal dimensionnés par rapport à ce que demandent les composants dessus et peut avoir des conséquences très aléatoires...
- Les symboles « - », « **0 V** », « **GND** », « **ground** », « **masse** » sont souvent reliés ensemble et viennent en général du fil **noir** (ou plus rarement marron) de la batterie (et à l'opposé il y a le symbole « + » venant souvent du fil **rouge** de la batterie). Ce n'est potentiellement pas le cas s'il y a des signaux alternatifs ou négatifs avec des tensions < 0 V. Il peut y avoir aussi e.g. « **+5 V** », « **+12 V** » séparés qui peuvent venir de **régulateurs/DC-DC converters** de tension
- Si la batterie d'un multimètre est déchargée, ses mesures peuvent être fortement faussées!
- **Les mauvaises alimentations sont des problèmes récurrents pour les robots autonomes**
- **Risques classiques** dans le cas de l'alimentation d'une Arduino UNO : <https://copilot.microsoft.com/shares/FxqUM4AhwpQWnUpPPAD64>
- Quelques exemples de consommation d'éléments de robot : https://www.ensta-bretagne.fr/lebars/tests_devices_power.csv

■ Connecteurs et câblage :

- Connecteurs et/ou câbles identiques : selon le contexte, les signaux passant dessus peuvent ne pas être les mêmes

e.g. sur des PCs de bureau ou portable pour le grand public, un connecteur RJ45 permet en général de se connecter à un réseau Ethernet, cependant sur une carte électronique industrielle, ce type de connecteur peut être utilisé pour d'autres choses...

- Câble droit vs croisé ou autre : même si un câble a les mêmes connecteurs à ses 2 extrémités, il est possible que les pins d'un côté ne soient pas reliés aux pins correspondantes de l'autre côté...



- Connecteurs et câblage :
 - **Tester des branchements au hasard se termine en général par du matériel endommagé (e.g. des inversions de polarité ou mauvaises tensions arrivent très facilement...)!**
 - **Bien lire les documentations quand on manipule des cartes électroniques, sinon risque de les endommager ainsi que les autres choses qui sont ou seront branchées dessus!**



■ Connecteurs et câblage :

- Le choix d'un type de connecteur et câble peut dépendre de plusieurs critères :

Taille, forme, poids, étanchéité

Courants, tensions

Etre facilement débranchable un grand nombre de fois, ou au contraire être difficilement débranchable, à noter que certains connecteurs (e.g. étanches) peuvent n'être spécifiés que pour e.g. 10 débranchements-rebranchements

Qualité du contact en cas de vibrations ou chocs

Fréquences des signaux

Blindage électromagnétique pour limiter les perturbations

Atténuation des signaux selon la longueur du câble

- **Couper un câble pour le raccourcir ou rallonger modifie ses propriétés!**

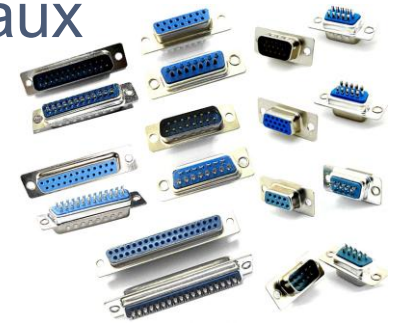
Concevoir un robot

- Connecteurs industriels classiques pour signaux numériques :

DB9, DB25



RJ45



A noter que les connecteurs USB mini sont particulièrement peu fiables en cas de vibrations ou chocs...

Concevoir un robot



- Connecteurs industriels classiques pour signaux numériques :

Molex PicoBlade (e.g. RS 125-0742, 279-9229)



JST-GH (e.g. chez Drotek)



JST-XH (e.g. RS 820-1551, 820-1618, 820-1541)



Hirose DF13 (e.g. RS 188-6317, 143-009, 189-6748)



Concevoir un robot



- Connecteurs industriels classiques pour signaux radio :

SMA/RP-SMA (e.g. Conrad 734400-62 (ou 733656-62), 733629-62)



Hirose U.FL-LP-088 (e.g. RS 794-2841), I-PEX MHF4 (e.g. amazon.fr B075X4PQYX)



Type N

BNC

RCA





- Connecteurs industriels classiques pour alimentation de PC ou cartes électroniques :

Barrel connector 5.5 x 2.5, 5.5 x 2.1 (diamètre externe x diamètre interne)



■ Connecteurs classiques de modélisme :

- Puissance :

T/Deans



XT60



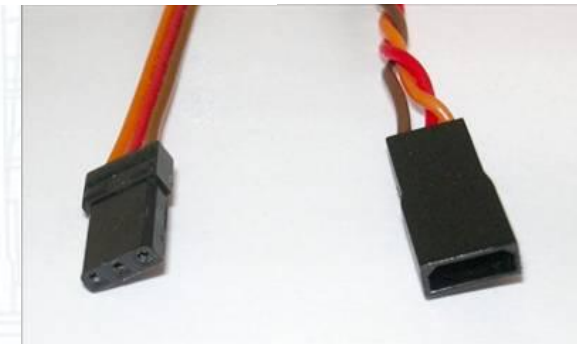
PK3.5 (conseillé pour les moteurs brushless)



- 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)





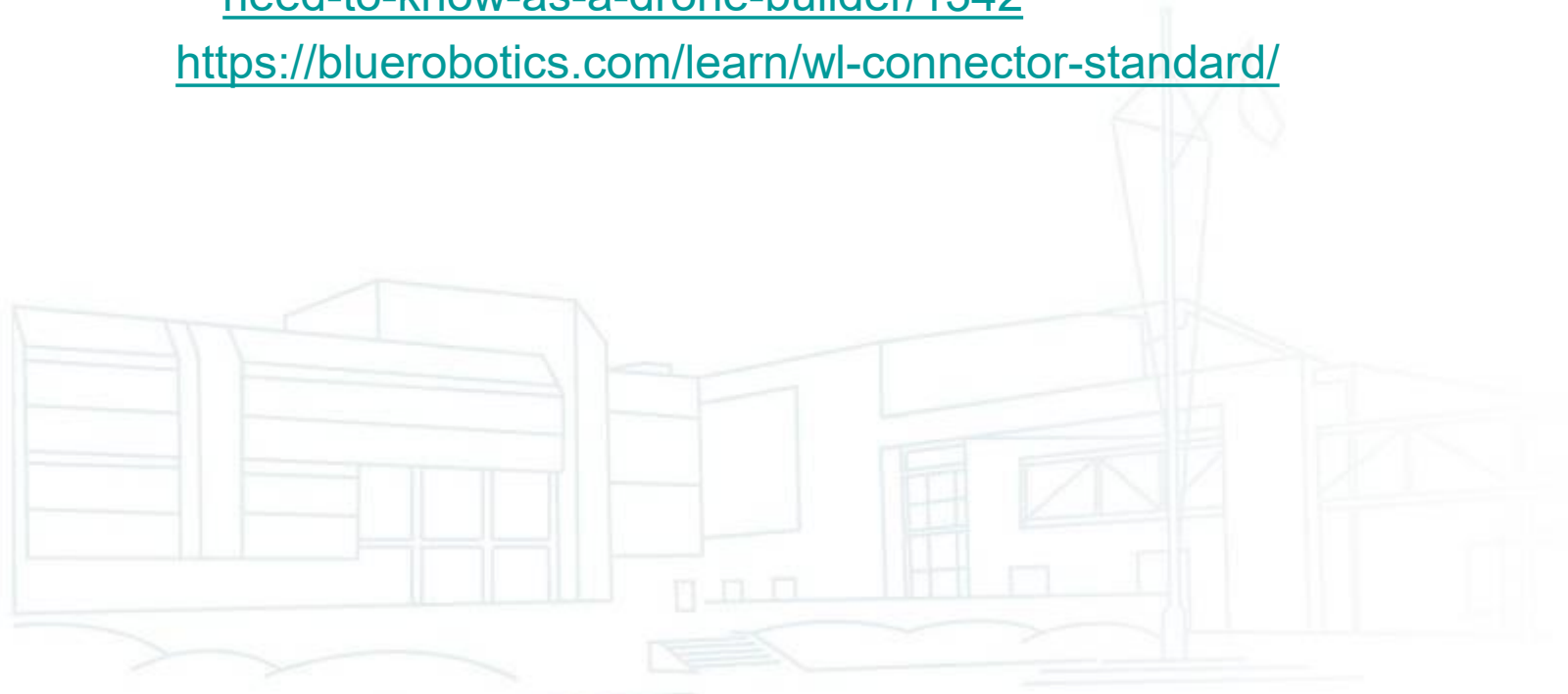
■ Connecteurs classiques de modélisme :

- Conventions usuelles, connecteurs étanches, etc. :

https://www.ensta-bretagne.fr/lebars/conventions_connecteurs.pdf

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

<https://bluerobotics.com/learn/wl-connector-standard/>



■ Passe-câbles :

- Si on n'a pas besoin de connecter-déconnecter un périphérique externe mais qu'on veut le relier à un boîtier étanche, on peut utiliser un ou des passe-câbles/presse-étoupes/penetrators :



Attention au diamètre et la déformation du câble notamment quand on le plie, 1 câble par passe-câble pour être étanche, voir la norme d'étanchéité IPXX (e.g. IP68) ainsi que pression ou profondeur max, attention au matériau constituant le câble si une résine (qui doit coller dessus) est nécessaire pour garantir l'étanchéité (e.g. pour anciens penetrators Blue Robotics), etc.

Si un gros connecteur doit être passé à travers (sans être démonté du câble), voir e.g. <https://www.icotek.com/fr/produits/presse-etoupes>

■ Câblage :

- Attention au **diamètre des câbles** (voir valeur AWG souvent inscrite sur les 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 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



■ Câblage :

- Certains signaux doivent aussi être sur des **paires torsadées** pour filtrer certaines perturbations : e.g. Ethernet, RS232, PWM (câbles servos > 25 cm), on peut aussi **blinder** les câbles i.e. les envelopper dans de l'aluminium qu'on relie à la masse à chaque extrémité, utiliser une ferrite...
- Pour les **dépôts 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 (mais pas toujours facile/pratique)...



Data rates and the maximum distances recommended in RS-232

Data rate (bps)	Distance (m)
2400	60
4800	30
9600	15
19200	7.6
38400	3.7
56000	2.6



<https://www.tldp.org/HOWTO/Remote-Serial-Console-HOWTO/serial-distance.html>

■ Soudure :

• Gaine thermorétractable

Pour **éviter les court-circuits**, on utilise des **gaines thermorétractables** par-dessus les soudures : pour cela on doit choisir une gaine thermorétractable suffisamment grande pour qu'on puisse la passer à l'avance sur le câble à souder, puis on soude en faisant attention à ce que la chaleur (qui se propage à travers les câbles) ne la rétracte pas, puis on la positionne au-dessus de la soudure et on la chauffe avec un pistolet à air chaud pour la rétracter...

Attention au diamètre de la gaine : le choisir suffisamment plus grand que le diamètre du câble car la soudure est souvent plus grande et la chaleur conduite à travers le câble fait souvent un peu rétrécir la gaine, ne pas le choisir trop grand non plus car selon la qualité de la gaine, elle peut ne pas se rétracter suffisamment même en chauffant beaucoup (typiquement le **rapport de rétractation est de 2**)

■ Soudure :

- Pour souder un connecteur, **brancher le connecteur opposé** permet souvent de limiter les déformations dues à la chaleur au moment de la soudure...
- Pour les soudures devant supporter des **gros courants**, ne pas hésiter à prendre le temps de **mettre beaucoup d'étain** pour que les surfaces de contacts soient suffisantes
- **Soudures sur batteries : attention à ne pas faire de court-circuit pendant la soudure!**
- Pannes de fer à souder : plusieurs formes existent en général, dépend de ce qu'on veut souder
- Tresse à dessouder : sert à enlever de l'étain, pour dessouder...
- Pompe à dessouder : aspire l'étain, pour dessouder...

- Manipulations, interventions, tests, résolution de problèmes, etc. :
 - **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!
 - Quand il y a un **problème** avec e.g. un capteur, **vérifier l'alim** avec un multimètre (pour être sûr que l'alim est bien configurée, que les fils sont branchés aux bons endroits et en bon état), vérifier avec un ampèremètre (ou directement avec l'alim si elle indique la conso) que le capteur consomme un peu de courant (pour s'assurer qu'il démarre)...



Concevoir un robot

- Manipulations, interventions, tests, résolution de problèmes, etc. :
 - Attention si l'espace de travail contient des **objets métalliques** : **risques de court-circuit!**
 - Attention aux **décharges électrostatiques** : toucher e.g. le métal d'un PC relié à la terre, voir e.g. <https://www.seagate.com/fr/fr/support/kb/what-is-electro-static-discharge-218791en/>
 - 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 check-list d'utilisation de robot, de départ en test en mer, etc., voir https://www.ensta-bretagne.fr/lebars/conseils_preparation_experimentations.pdf

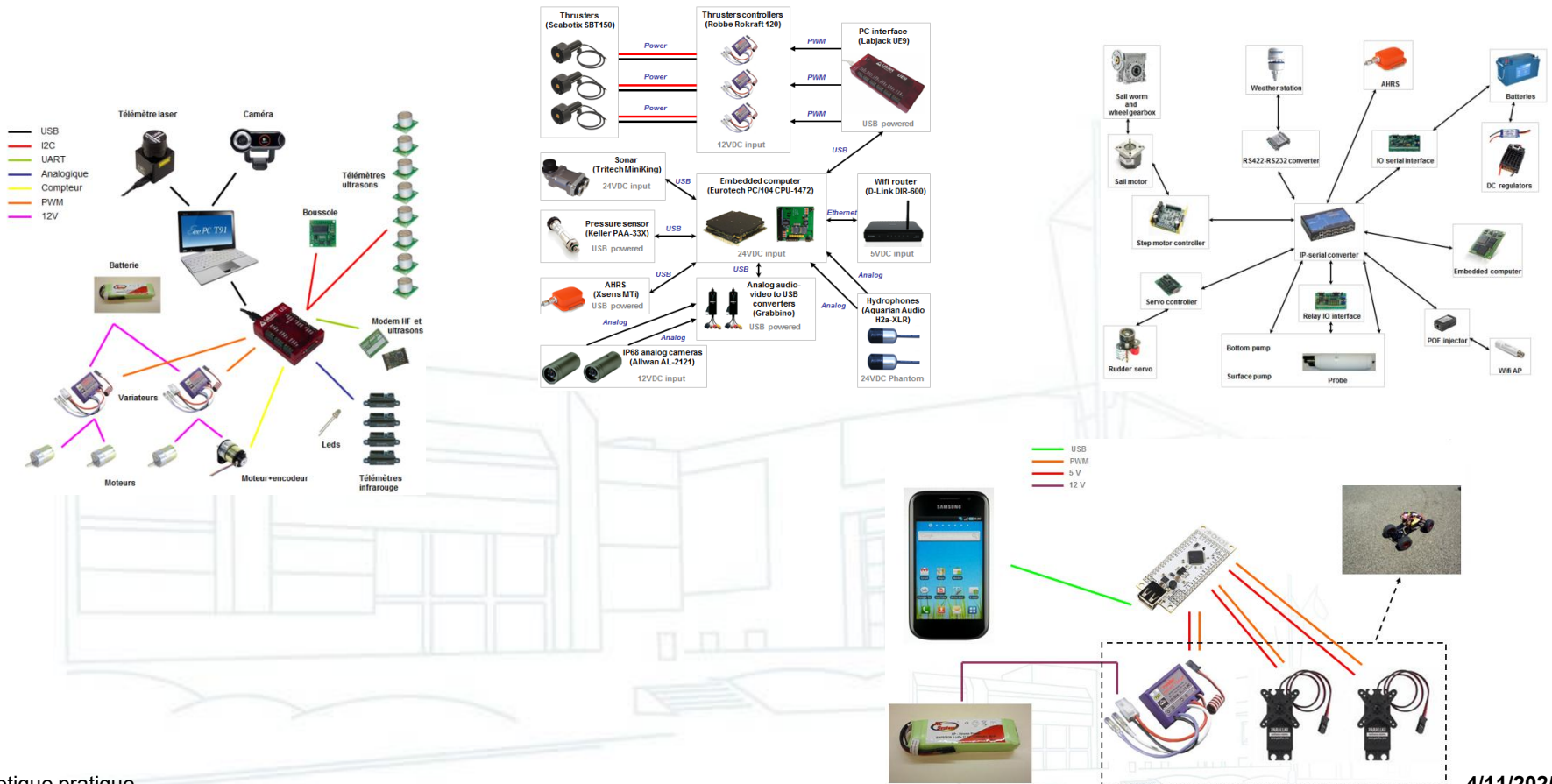
- Boucle de conception de robot :
 - En particulier pour les drones (mais pas seulement), problèmes interdépendants pour le poids, surface alaire, motorisation, hélice, batterie, etc.



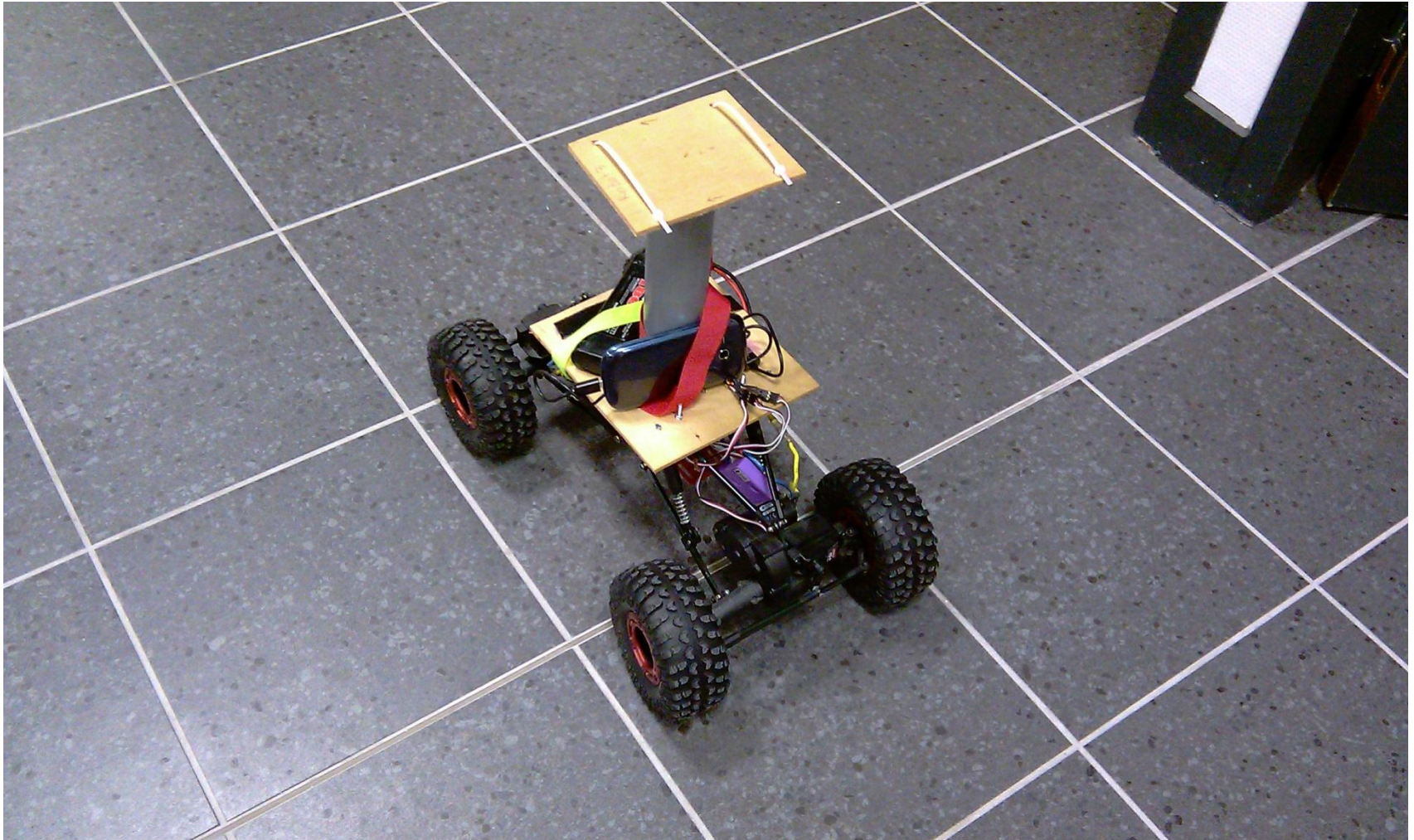
Concevoir un robot



Exemples d'architectures pour différents types de robots



AGV Buggy (2012)



AGV Buggy (2012)

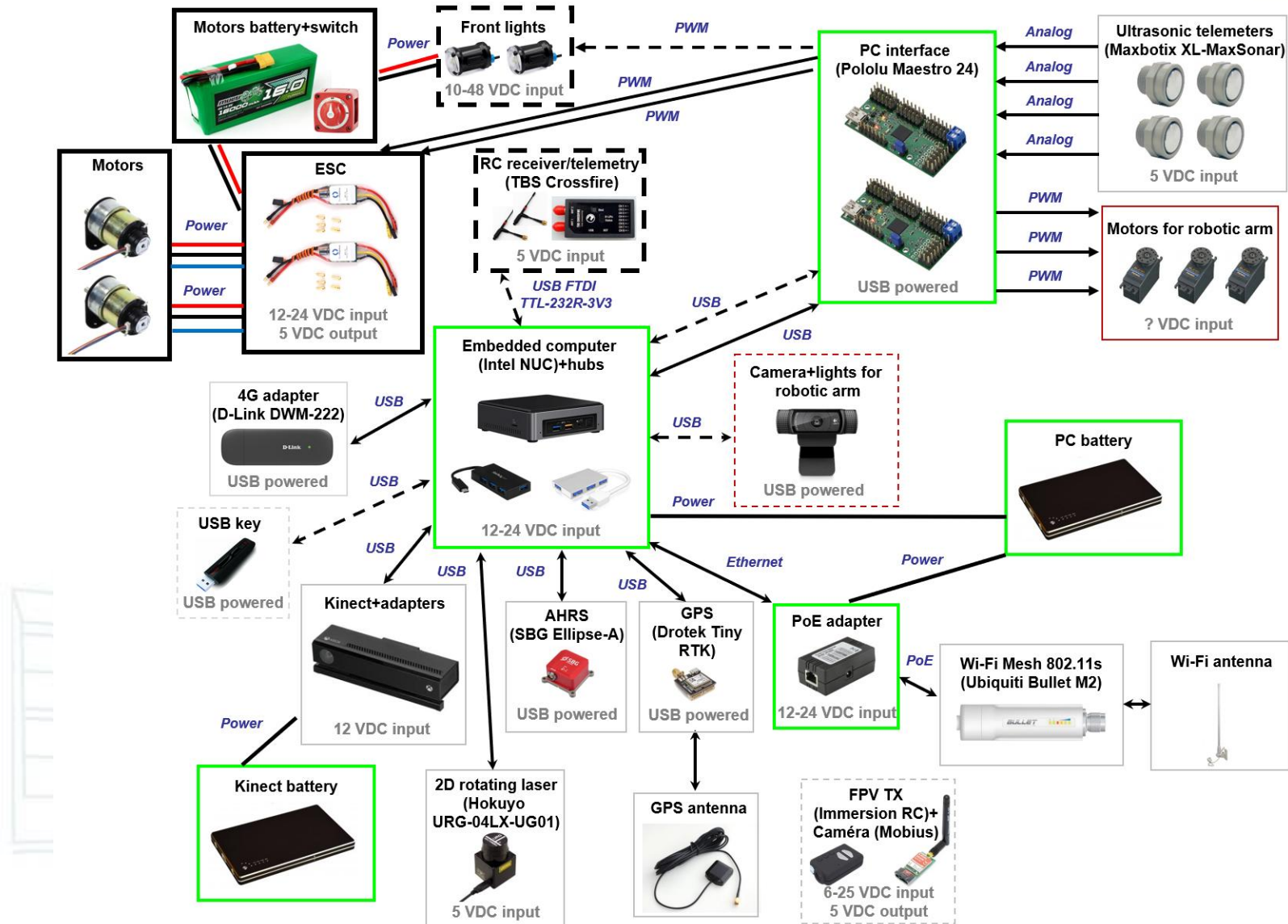
- USB
- PWM
- 5 V
- 12 V



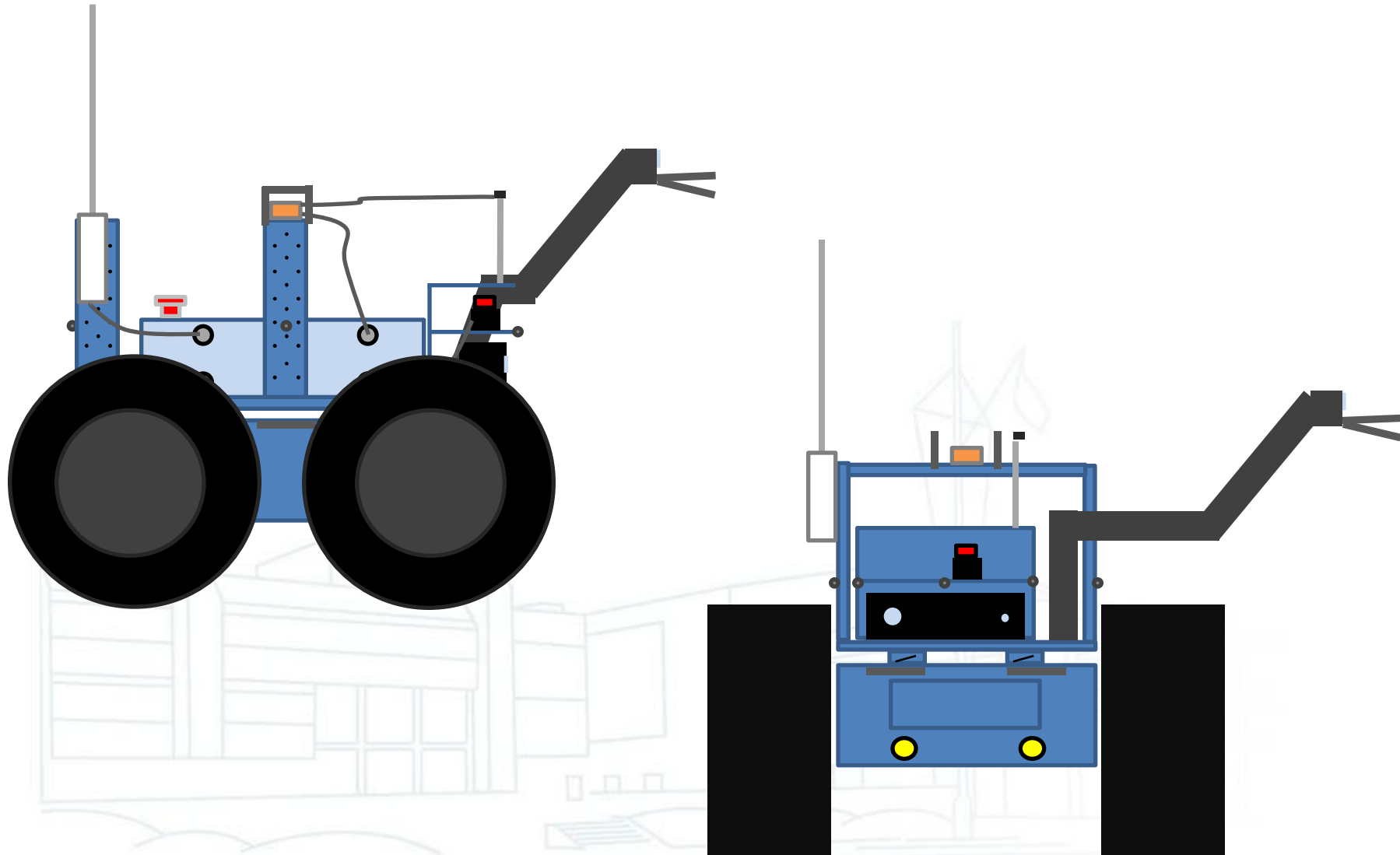
AGV 4-WHEEL (2019)



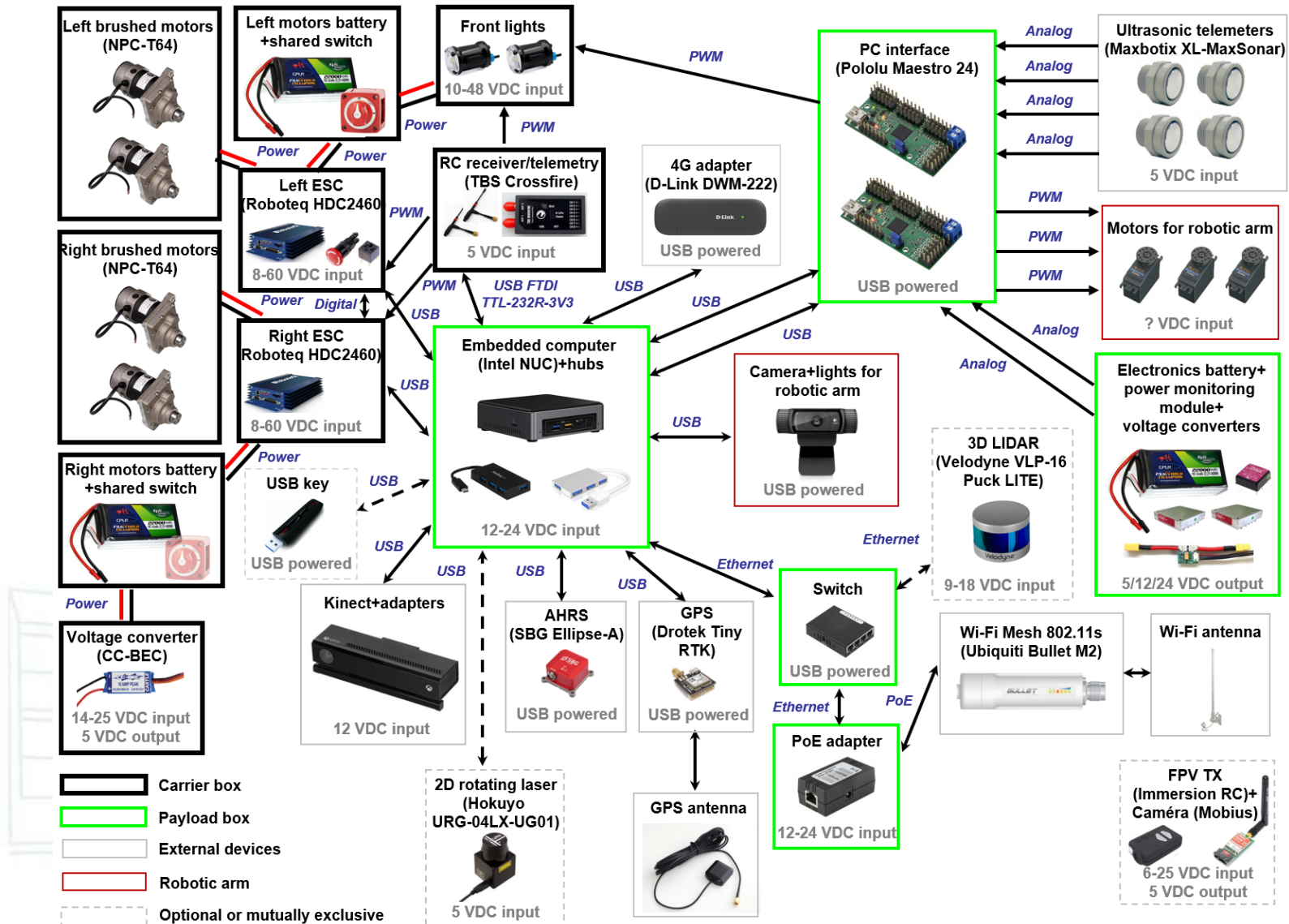
AGV 4-WHEEL (2019)



Future AGV for euRathlon/ERL



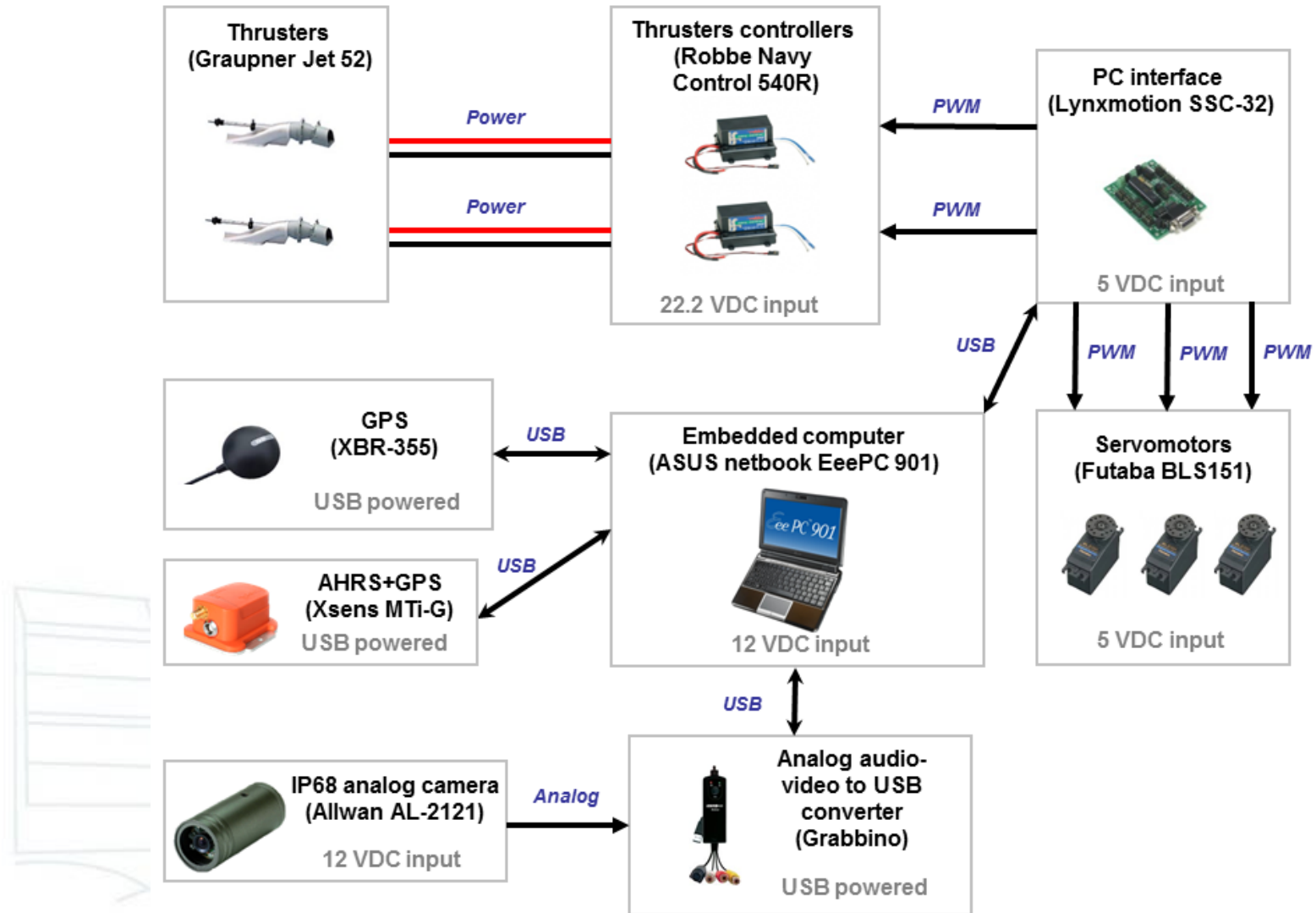
Future AGV for euRathlon/ERL



ASV Motorboat (2013)



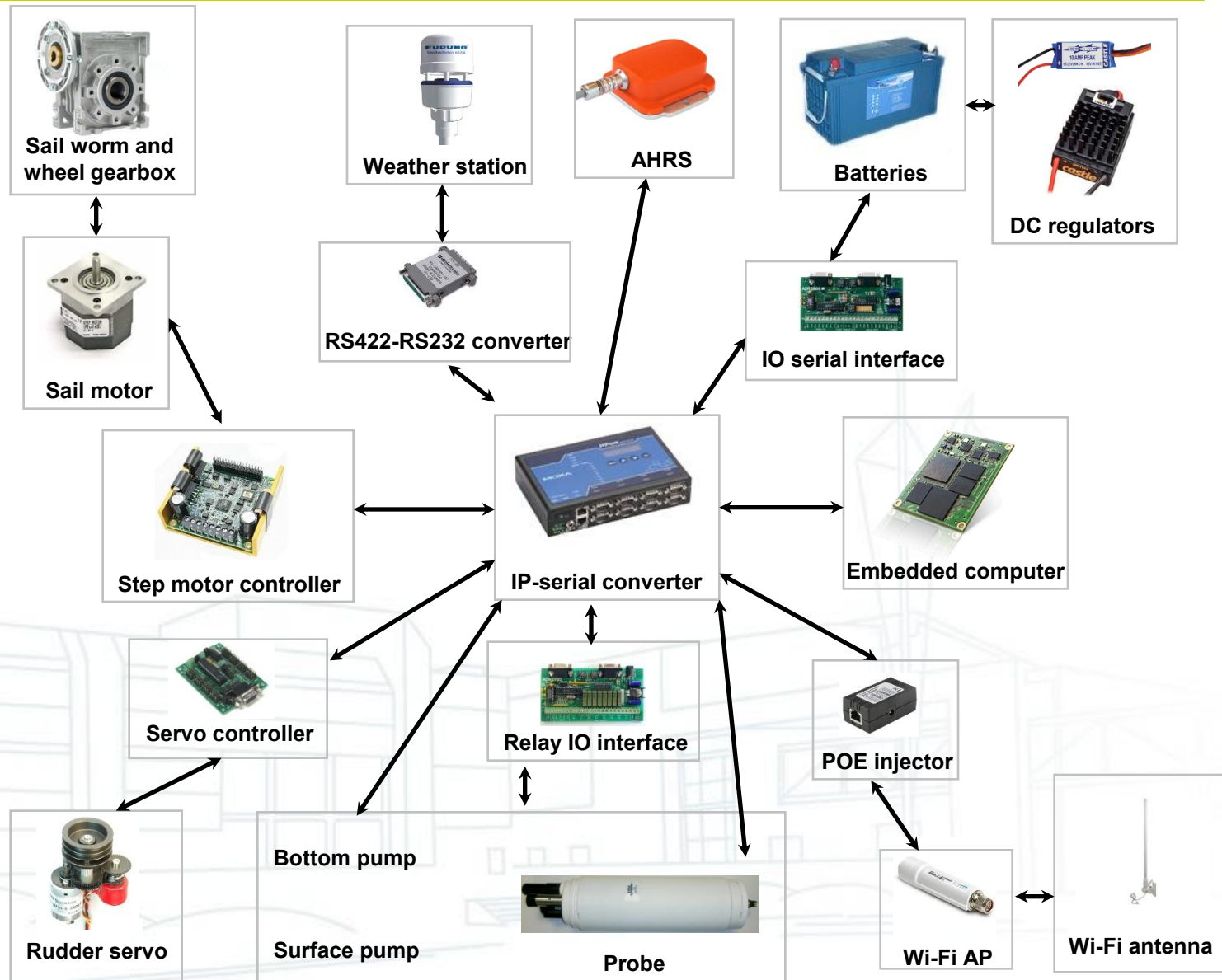
ASV Motorboat (2013)



ASV VAIMOS (2011)



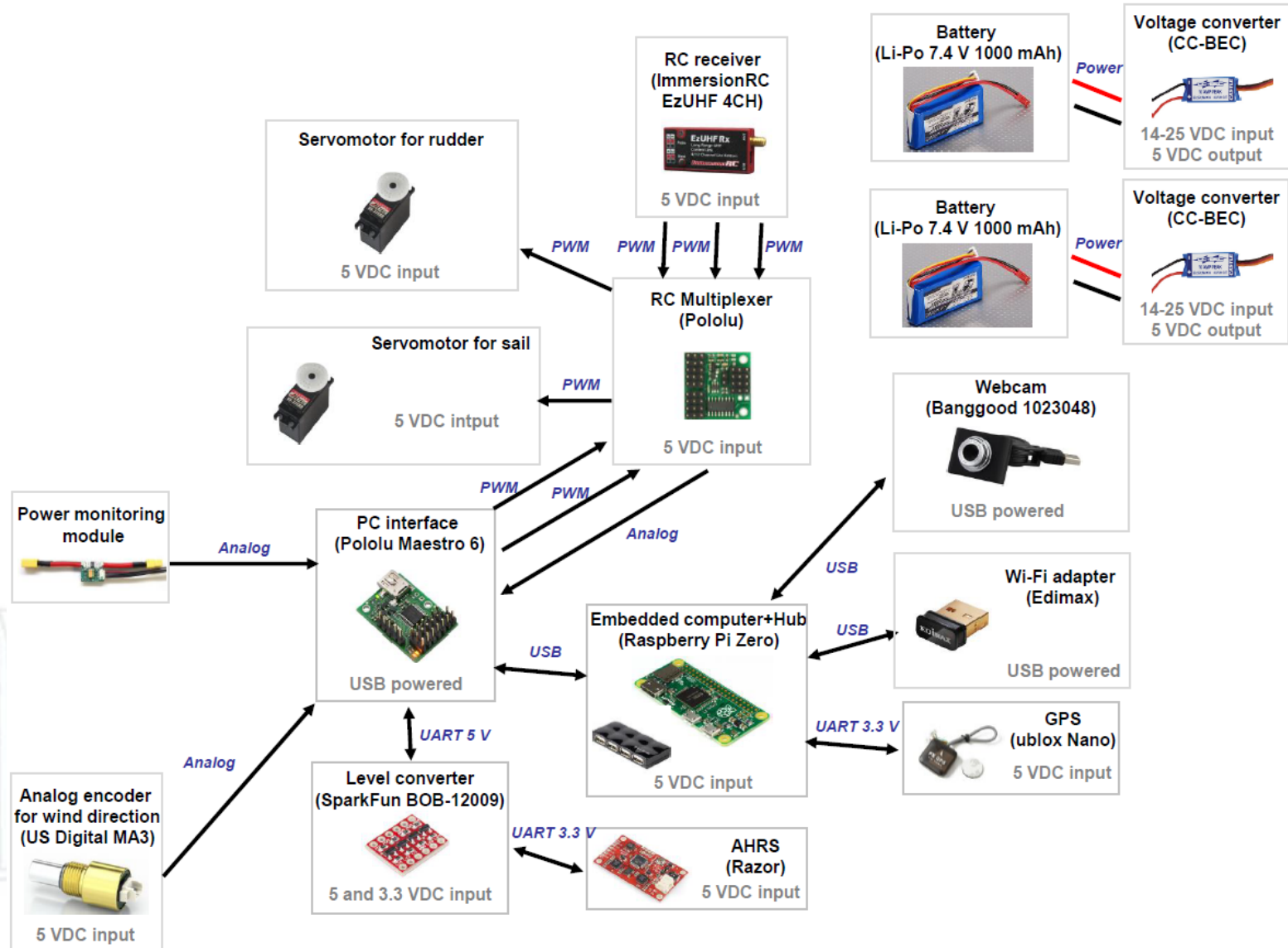
ASV VAIMOS (2011)



ASV White Sailboat (2016)



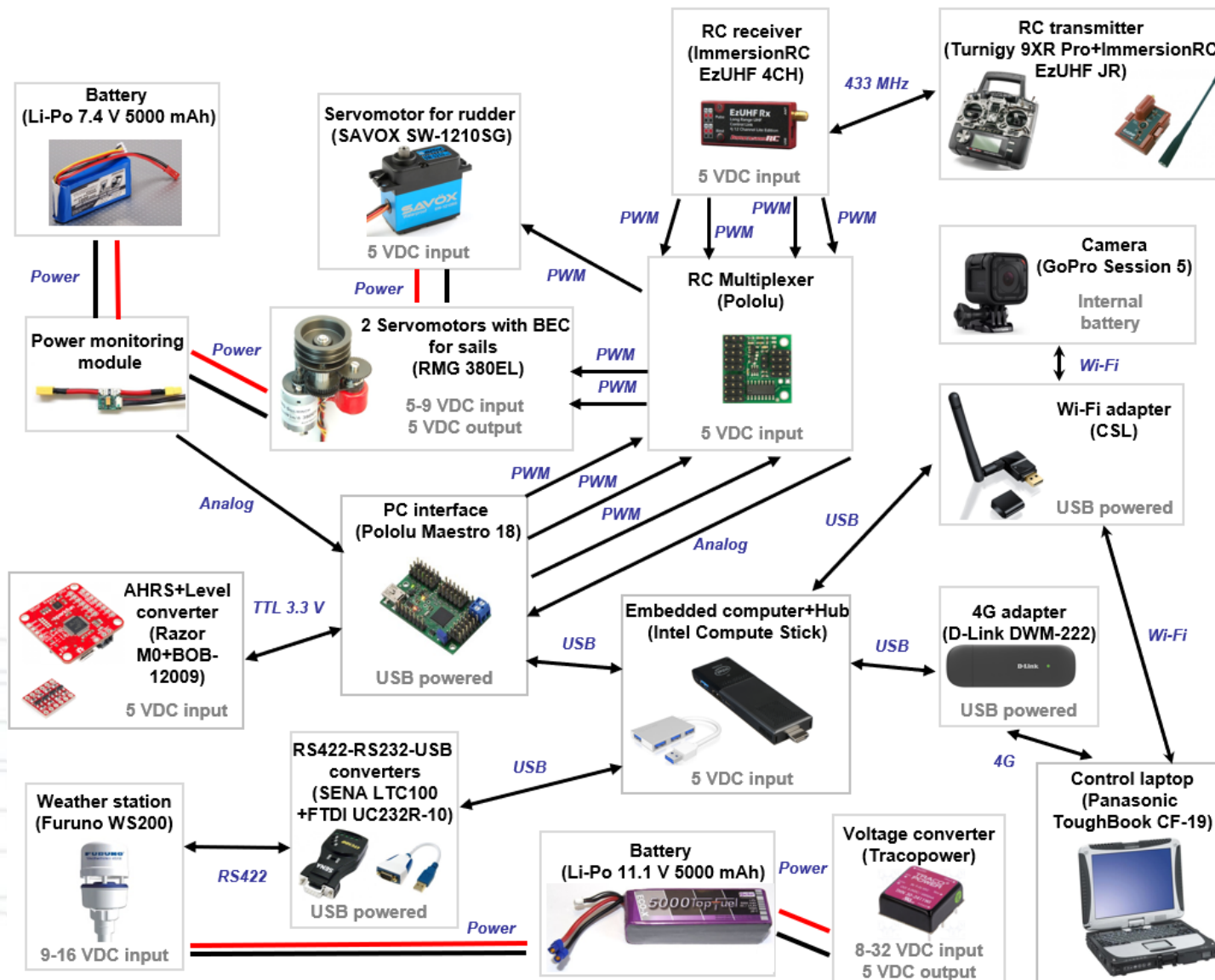
ASV White Sailboat (2016)



ASV BRAVE (2018)



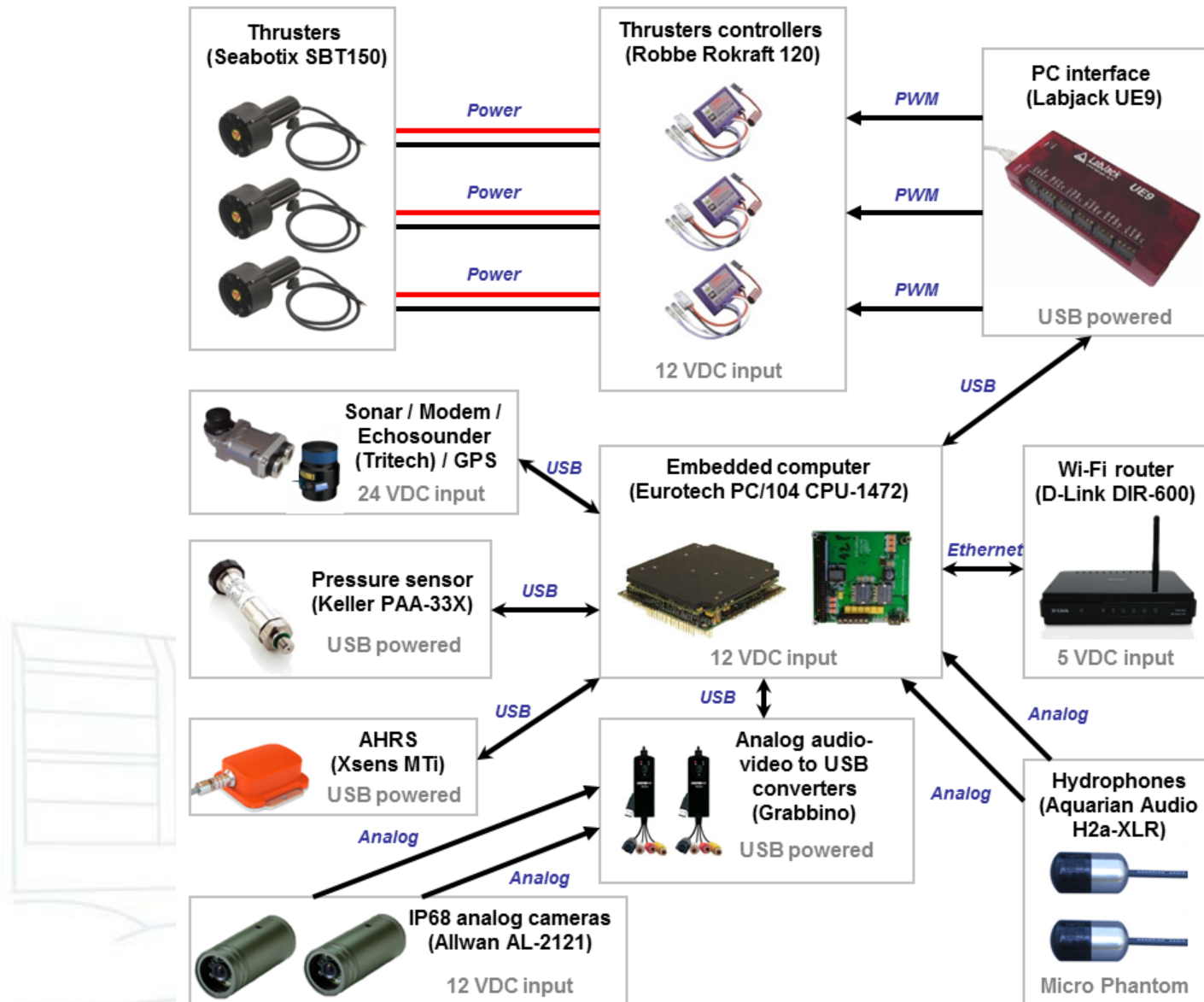
ASV BRAVE (2018)



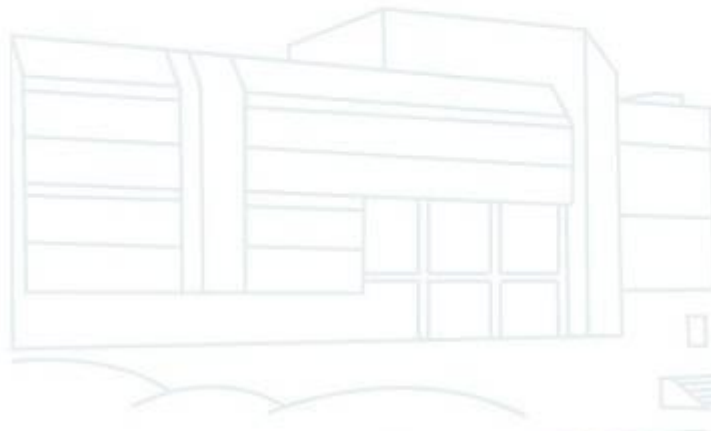
AUV SAUC'ISSE (2014)



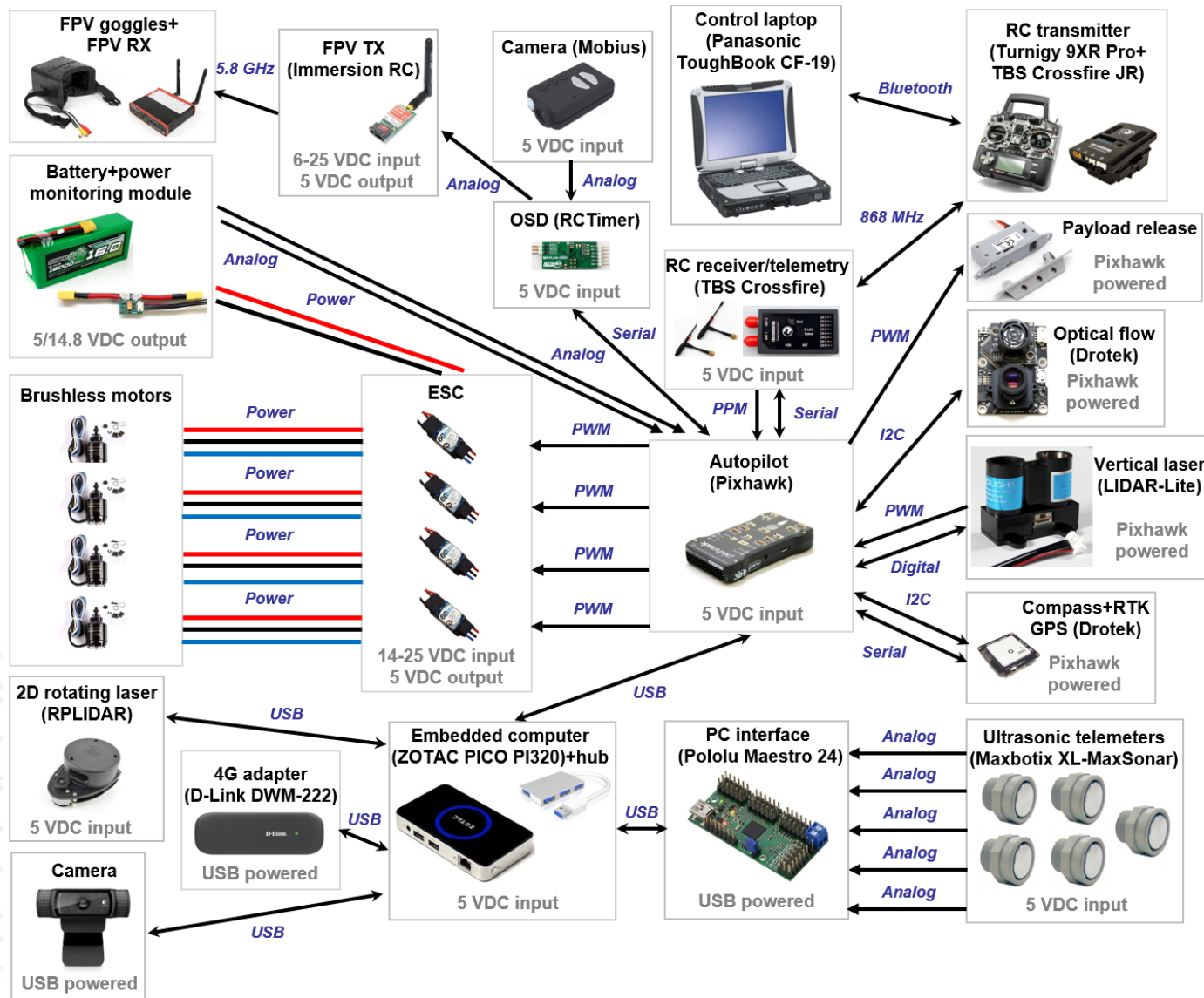
AUV SAUC'ISSE (2014)



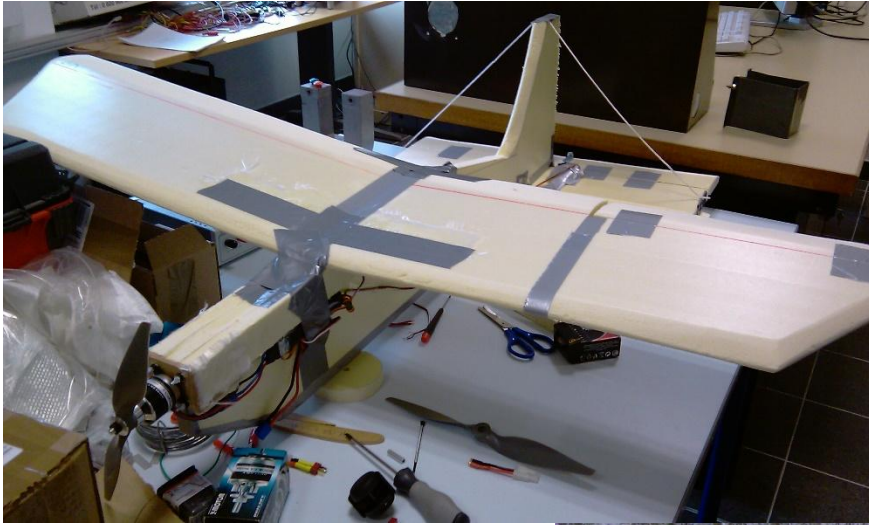
Indoor Quadrotor euRathlon/ERL



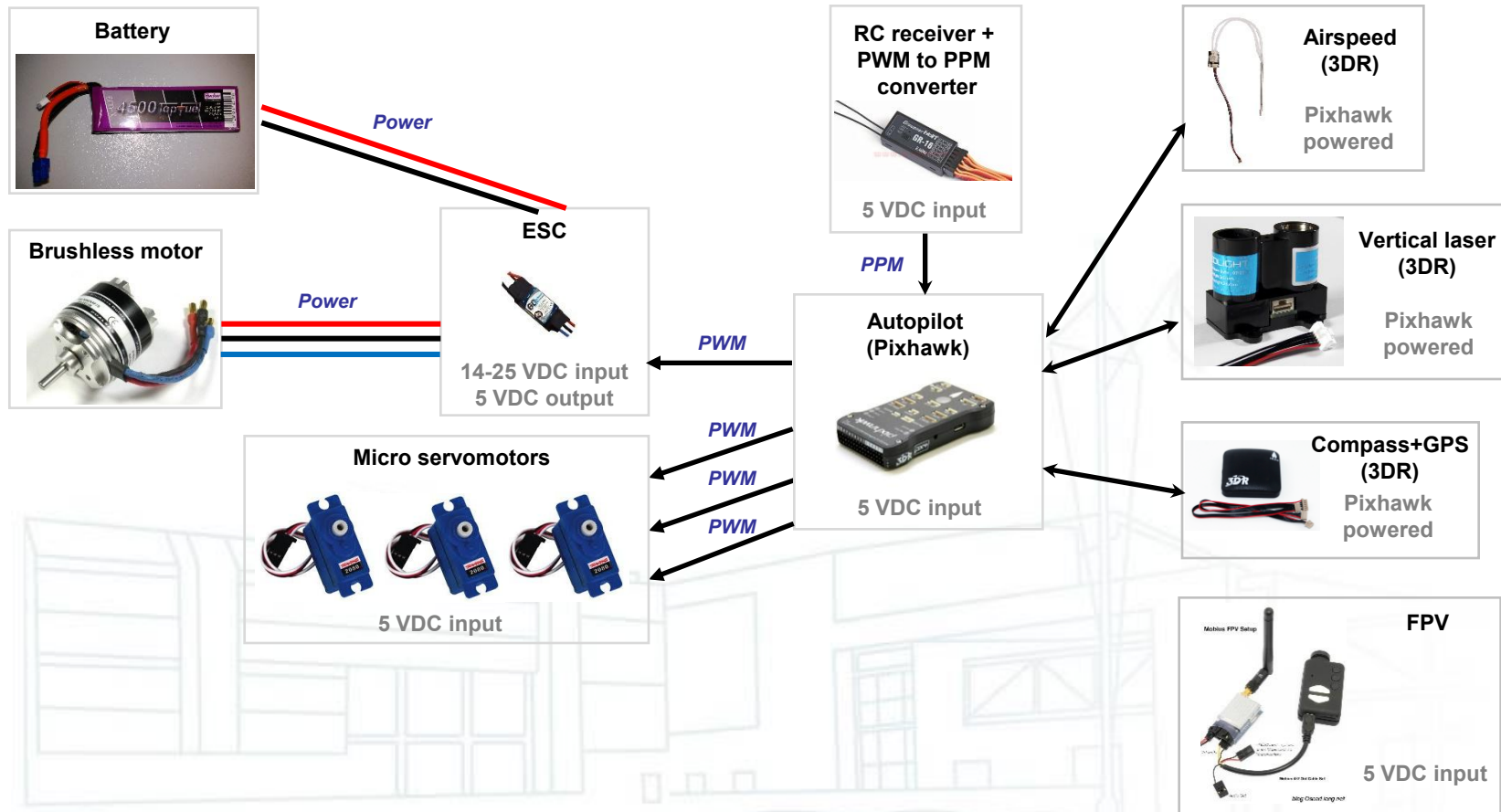
Indoor Quadrotor euRathlon/ERL



Plane

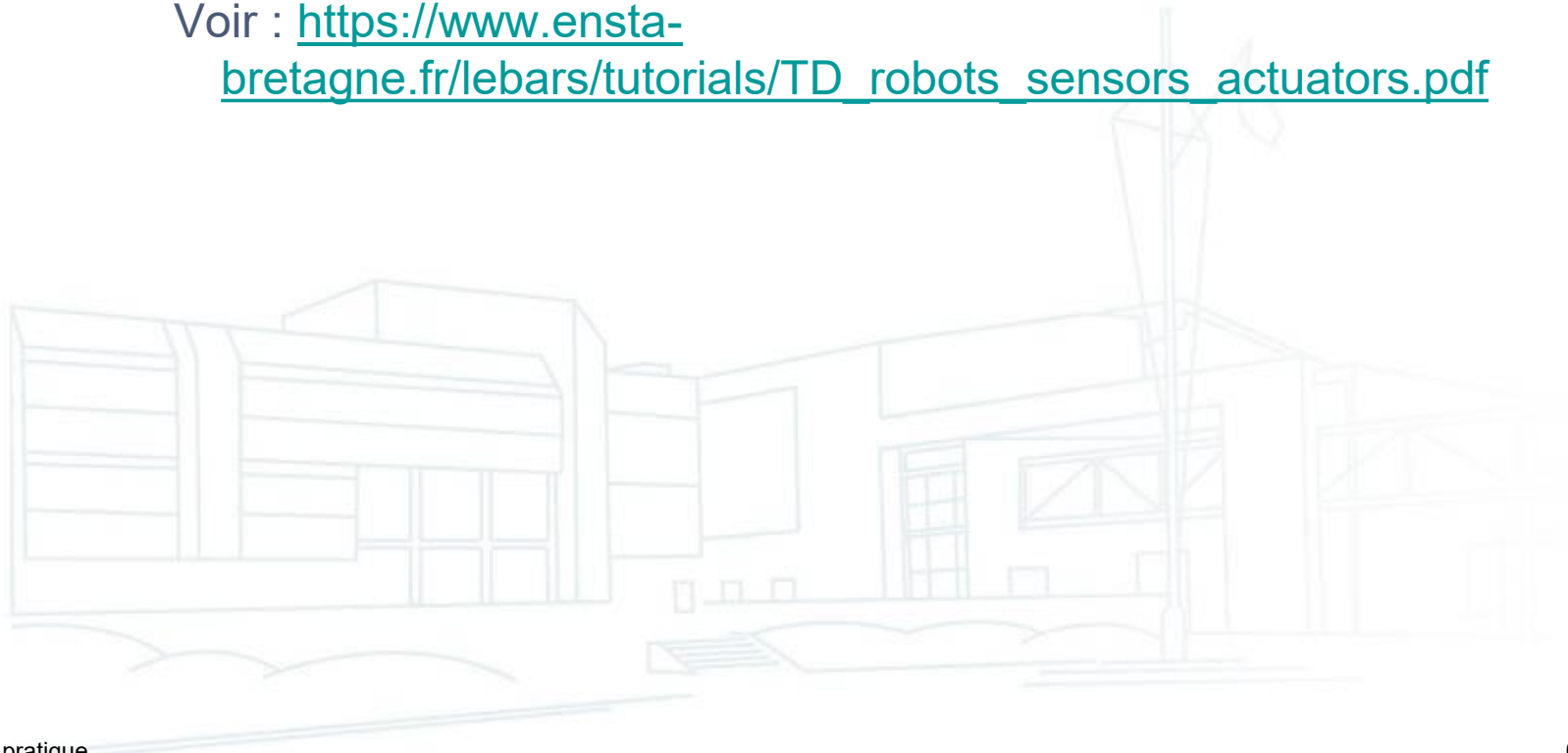


Plane



Concevoir un robot

- Choix, configuration, utilisation de centrales inertielles, GPS, moteurs, servomoteurs, batteries, télécommandes, autopilotes, etc. :
 - Voir plus d'infos dans le sujet de TD correspondant :
Voir : https://www.ensta-bretagne.fr/lebars/tutorials/TD_robots_sensors_actuators.pdf





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

- Principe de fonctionnement d'un ordinateur simplifié
 - On peut découper un ordinateur/computer simplifié en 3 parties principales :

Le processeur/processor



La mémoire/memory



Les périphériques/devices



Communiquer, contrôler à distance un système informatique autonome

■ Principe de fonctionnement d'un ordinateur simplifié

• Mémoire

Liste d'octets (1 octet/byte=8 bits=256 valeurs possibles), souvent lus 32 bits par 32 bits i.e. 4 octets par 4 octets

En pratique, elle peut être répartie dans plusieurs composants physiques ou partie de composants, et selon les situations l'ensemble est souvent vu du processeur comme une liste d'octets continue, avec parfois quelques zones mémoire séparées particulières (e.g. registres dans processeur, cache, etc.)



Communiquer, contrôler à distance un système informatique autonome

- Principe de fonctionnement d'un ordinateur simplifié
 - **Adresse mémoire**

Index dans la mémoire. L'adresse 0 pointe vers les données du 1er octet de la mémoire

Selon les situations, il se peut que les adresses soient comptées octet par octet, 32 bits par 32 bits, 64 bits par 64 bits, etc. et souvent représentées par des nombres hexadécimaux (1 chiffre hexa=4 bits, donc 1 octet est représenté par 2 chiffres hexa)



Communiquer, contrôler à distance un système informatique autonome

■ Principe de fonctionnement d'un ordinateur simplifié

• Périphériques

On pourrait imaginer que certaines adresses mémoire particulières soient scrutées par certains périphériques pour déclencher une action physiquement visible e.g.

Allumer le 1er pixel à l'**écran** si la valeur à l'adresse 0x00010000 est différente de 0

Inversement le **clavier** pourrait stocker l'état de ses différentes touches sous la forme d'une valeur allant de 0 à 255 à l'adresse 0x0000ffff

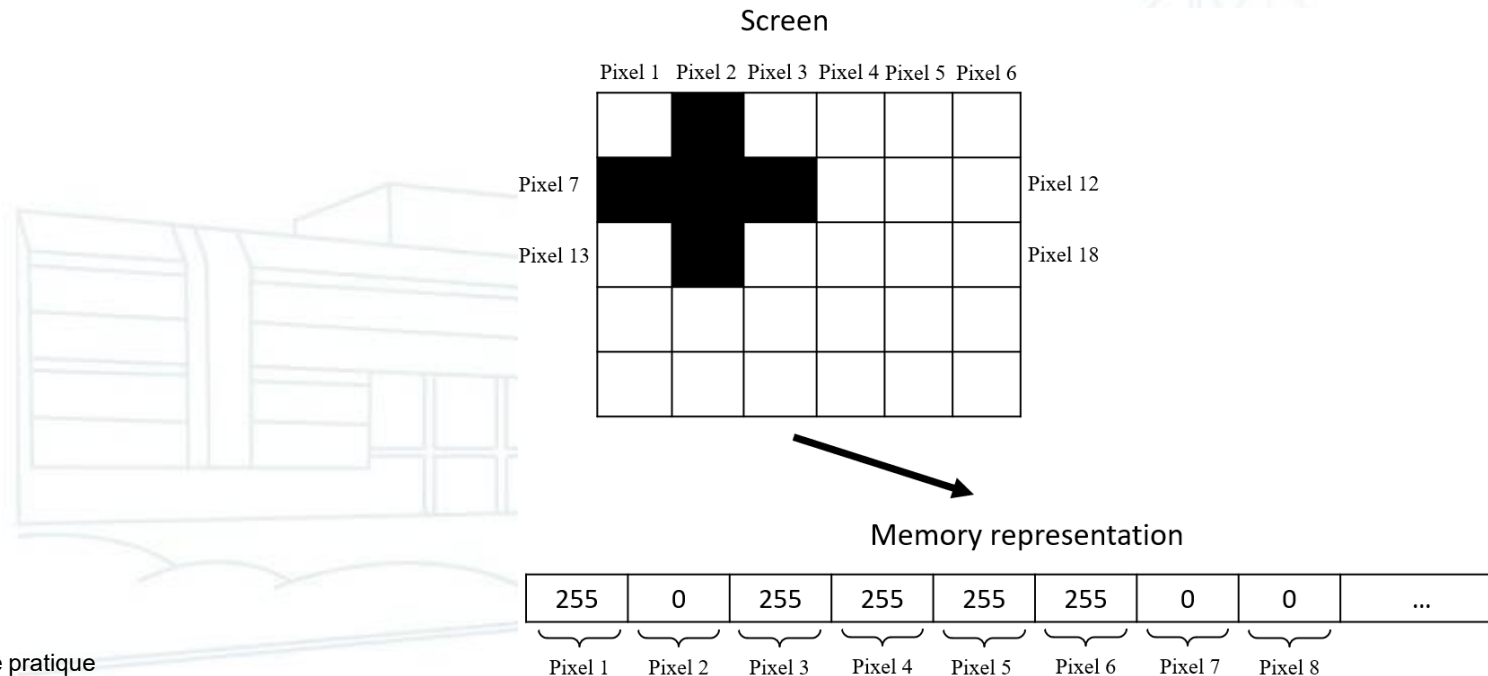


Communiquer, contrôler à distance un système informatique autonome

- Principe de fonctionnement d'un ordinateur simplifié
 - **Périphériques**

Pour l'écran, on pourrait imaginer qu'il puisse avoir 2 modes :

- **Dessin** : une zone de la mémoire continue pourrait correspondre à **1 pixel par octet**, sa valeur étant la couleur du pixel (e.g. 0 : noir, 255 : blanc, si l'écran fait 320x240 pixels la zone mémoire réservée ferait $320 \times 240 = 76800$ octets, la couleur du pixel à la ligne i et colonne j étant à l'emplacement mémoire $j + 320 * i$)



Communiquer, contrôler à distance un système informatique autonome

■ Principe de fonctionnement d'un ordinateur simplifié

• Périphériques

Pour l'écran, on pourrait imaginer qu'il puisse avoir 2 modes :

- **Caractère** : une zone de la mémoire pourrait correspondre à **1 caractère ASCII par octet**, sa valeur étant le code ASCII du caractère à afficher (l'écran s'occuperait alors en interne d'allumer les bons pixels pour dessiner le caractère voulu, si l'écran peut afficher 16x6 caractères, la zone mémoire réservée ferait 16x6=96 octets, le code du caractère à la ligne i et colonne j étant à l'emplacement mémoire $j+16*i$)



ASCII control characters			
DEC	HEX	Simbolo	ASCII
00	00h	NULL	(carácter nulo)
01	01h	SOH	(inicio encabezado)
02	02h	STX	(inicio texto)
03	03h	ETX	(fin de texto)
04	04h	EOT	(fin transmisión)
05	05h	ENQ	(enquiry)
06	06h	ACK	(acknowledgement)
07	07h	BEL	(timbre)
08	08h	BS	(retroceso)
09	09h	HT	(tab horizontal)
10	0Ah	LF	(salto de línea)
11	0Bh	VT	(tab vertical)
12	0Ch	FF	(form feed)
13	0Dh	CR	(retorno de carro)
14	0Eh	SO	(shift out)
15	0Fh	SI	(shift in)
16	10h	DLE	(data link escape)
17	11h	DC1	(device control 1)
18	12h	DC2	(device control 2)
19	13h	DC3	(device control 3)
20	14h	DC4	(device control 4)
21	15h	NAK	(negative acknowle.)
22	16h	SYN	(synchronous idle)
23	17h	ETB	(end of trans. block)
24	18h	CAN	(cancel)
25	19h	EM	(end of medium)
26	1Ah	SUB	(substitute)
27	1Bh	ESC	(escape)
28	1Ch	FS	(file separator)
29	1Dh	GS	(group separator)
30	1Eh	RS	(record separator)
31	1Fh	US	(unit separator)
127	20h	DEL	(delete)

ASCII printable characters								
DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	\$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(72	48h	H	104	68h	h
41	29h)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	
61	3Dh	=	93	5Dh]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_			

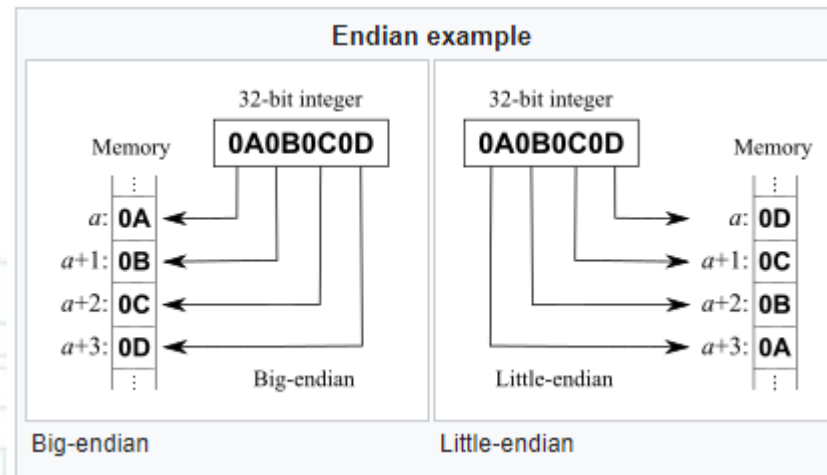
Extended ASCII characters											
DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	à	192	C0h	Ł	224	E0h	Ó
129	81h	ü	161	A1h	á	193	C1h	ł	225	E1h	ó
130	82h	é	162	A2h	â	194	C2h	Ł	226	E2h	Ô
131	83h	ê	163	A3h	ã	195	C3h	ł	227	E3h	ô
132	84h	ë	164	A4h	ä	196	C4h	Ł	228	E4h	Ï
133	85h	ä	165	A5h	å	197	C5h	ł	229	E5h	ï
134	86h	å	166	A6h	æ	198	C6h	Ł	230	E6h	µ
135	87h	ç	167	A7h	ß	199	C7h	ł	231	E7h	µ
136	88h	è	168	A8h	¸	200	C8h	Ł	232	E8h	þ
137	89h	é	169	A9h	¸	201	C9h	ł	233	E9h	þ
138	8Ah	è	170	AAh	¸	202	CAh	Ł	234	EAh	Û
139	8Bh	ï	171	ABh	¼	203	CBh	ł	235	Ebh	Û
140	8Ch	ï	172	ACH	½	204	Ch	Ł	236	ECh	Ý
141	8Dh	ï	173	ADh	¾	205	CDh	ł	237	EDh	ý
142	8Eh	ı	174	Aeh	ı	206	Ceh	Ł	238	Eeh	ÿ
143	8Fh	ı	175	Afh	ı	207	Cfh	ł	239	Efh	ı
144	90h	É	176	B0h	ı	208	D0h	Ł	240	F0h	ı
145	91h	é	177	B1h	ı	209	D1h	ł	241	F1h	ı
146	92h	Æ	178	B2h	ı	210	D2h	Ł	242	F2h	ı
147	93h	ø	179	B3h	ı	211	D3h	ł	243	F3h	ı
148	94h	ø	180	B4h	ı	212	D4h	Ł	244	F4h	ı
149	95h	ö	181	B5h	ı	213	D5h	ł	245	F5h	ı
150	96h	ö	182	B6h	ı	214	D6h	Ł	246	F6h	ı
151	97h	ù	183	B7h	ı	215	D7h	ł	247	F7h	ı
152	98h	ÿ	184	B8h	ı	216	D8h	Ł	248	F8h	ı
153	99h	Û	185	B9h	ı	217	D9h	ł	249	F9h	ı
154	9Ah	Ü	186	BAh	ı	218	DAh	Ł	250	FAh	ı
155	9Bh	ü	187	Bbh	ı	219	DBh	ł	251	FBh	ı
156	9Ch	ø	188	BCh	ı	220	DCh	Ł	252	FCh	ı
157	9Dh	ø	189	BDh	ı	221	DDh	ł	253	FDh	ı
158	9Eh	x	190	BEh	ı	222	DEh	Ł	254	FEh	ı
159	9Fh	f	191	Bfh	ı	223	DFh	ł	255	FFh	ı

Communiquer, contrôler à distance un système informatique autonome

■ Principe de fonctionnement d'un ordinateur simplifié

• Endianness

Manière (ordre...) dont les ordinateurs organisent les octets pour constituer des nombres



Les PCs travaillent principalement en Little-endian en interne

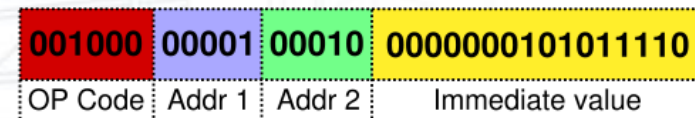
Communiquer, contrôler à distance un système informatique autonome

- Principe de fonctionnement d'un ordinateur simplifié
 - **Jeu d'instructions/instruction set architecture**

Table de correspondance entre des actions pouvant être effectuées par le processeur, et des valeurs, souvent représentées par des nombres hexadécimaux ou des mnémoniques (symboles courts d'une à quelques lettres, définissant un langage dit **assembleur**)

Bien souvent, une instruction donnée a besoin de **paramètres**, indiqués directement à la suite de l'instruction et avec une taille et position relative dans la mémoire prédéfinies

MIPS32 Add Immediate Instruction



Equivalent mnemonic: **addi** \$r1, \$r2, 350

Communiquer, contrôler à distance un système informatique autonome

- Principe de fonctionnement d'un ordinateur simplifié
 - **Jeu d'instructions/instruction set architecture**

Les actions classiques peuvent être similaires à :

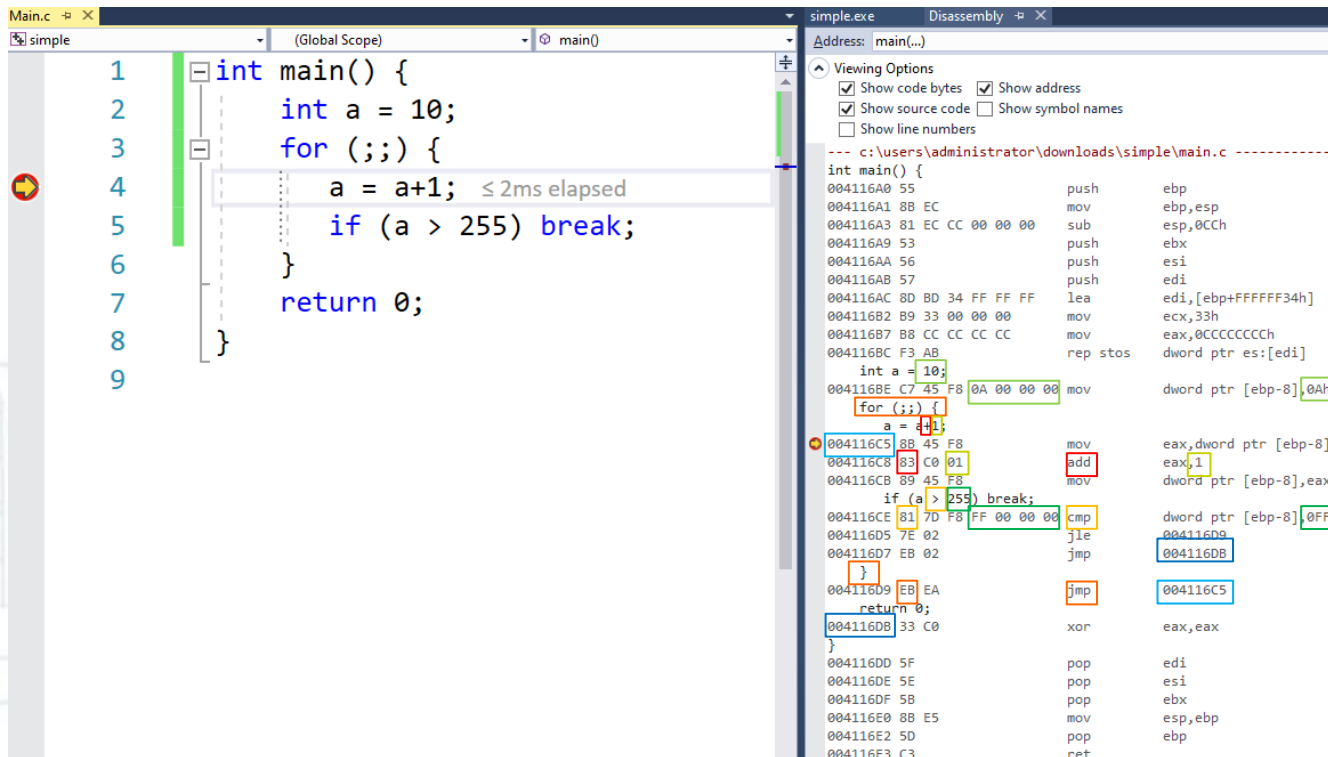
- Ecriture d'une valeur à une adresse donnée
- Copie d'une valeur à une adresse donnée vers une autre adresse donnée
- Addition (et autres opérateurs arithmétiques et logiques usuels) de valeurs à 2 adresses données et enregistrement du résultat à une adresse donnée
- Comparaison de valeurs à 2 adresses données et enregistrement du résultat à une adresse donnée
- Exécution de ce qui se trouve à une adresse donnée si la valeur à une autre adresse donnée est différente de 0
- etc.

Voir par exemple les actions possibles dans le jeu d'instruction de <https://fr.wikipedia.org/wiki/RISC-V> (exemple de Reduced Instruction Set Computer (RISC)), <https://www.felixcloutier.com/x86/> (exemple de Complex Instruction Set Computer (CISC))

Communiquer, contrôler à distance un système informatique autonome

- Principe de fonctionnement d'un ordinateur simplifié
 - **Processeur**

A son démarrage, lit la mémoire et interprète celle-ci comme une suite d'instructions à exécuter en suivant son jeu d'instructions...



```
1 int main() {
2     int a = 10;
3     for (;;) {
4         a = a+1; ≤ 2ms elapsed
5         if (a > 255) break;
6     }
7     return 0;
8 }
9
```

```
simple.exe Disassembly
Address: main(...)
Viewing Options
[ ] Show code bytes [x] Show address
[x] Show source code [ ] Show symbol names
[ ] Show line numbers
----- c:\users\administrator\downloads\simple\main.c -----
int main() {
004116A0 55                push     ebp
004116A1 8B EC            mov     ebp,esp
004116A3 81 EC CC 00 00 00 sub     esp,0CCCh
004116A9 53                push     ebx
004116AA 56                push     esi
004116AB 57                push     edi
004116AC 8D BD 34 FF FF FF lea     edi,[ebp+FFFFFF34h]
004116B2 B9 33 00 00 00  mov     ecx,33h
004116B7 B8 CC CC CC CC  mov     eax,0CCCCCCCCh
004116BC F3 AB            rep stos dword ptr es:[edi]
int a = 10;
004116BE C7 45 F8 0A 00 00 00 mov     dword ptr [ebp-8],0Ah
for (;;) {
a = a+1;
004116C5 8B 45 F8            mov     eax,dword ptr [ebp-8]
004116C8 83 C0 01            add     eax,1
004116CB 89 45 F8            mov     dword ptr [ebp-8],eax
if (a > 255) break;
004116CE 81 7D F8 FF 00 00 00 cmp     dword ptr [ebp-8],0FFh
004116D5 7E 02            jle     004116D9
004116D7 EB 02            jmp     004116DB
}
004116D9 EB EA            jmp     004116C5
return 0;
004116DB 33 C0            xor     eax,eax
}
004116DD 5F                pop     edi
004116DE 5E                pop     esi
004116DF 5B                pop     ebx
004116E0 8B E5            mov     esp,ebp
004116E2 5D                pop     ebp
004116E3 C3                ret
```

Communiquer, contrôler à distance un système informatique autonome

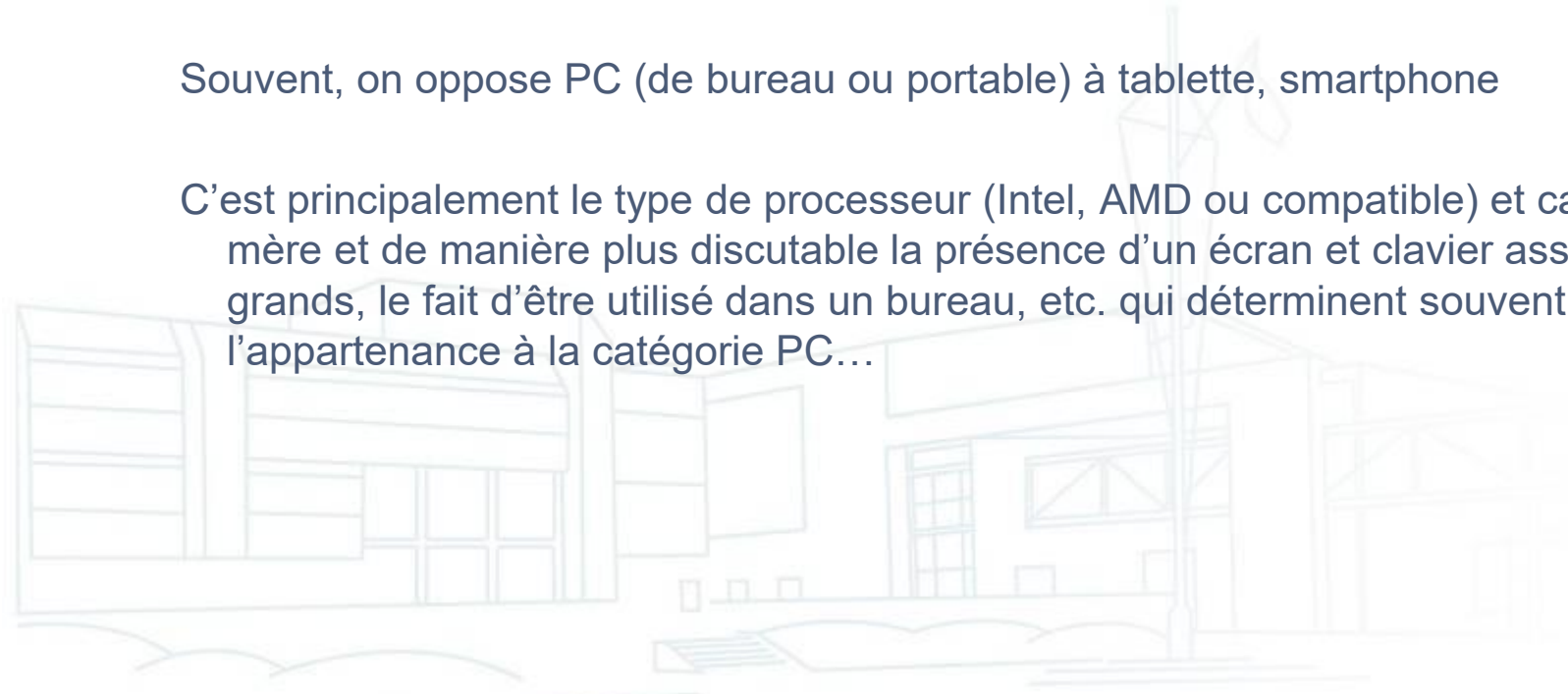
■ Montage d'un PC :

- PC

PC signifie Personal Computer mais est aujourd'hui un sigle assez vague en pratique, presque aussi vague qu'ordinateur (qui peut être vu comme plus large et comprenant aujourd'hui PC, Mac, Raspberry Pi et similaires)...

Souvent, on oppose PC (de bureau ou portable) à tablette, smartphone

C'est principalement le type de processeur (Intel, AMD ou compatible) et carte mère et de manière plus discutable la présence d'un écran et clavier assez grands, le fait d'être utilisé dans un bureau, etc. qui déterminent souvent l'appartenance à la catégorie PC...



Communiquer, contrôler à distance un système informatique autonome

■ Montage d'un PC :

• Architecture, machine...

PC : processeurs **x86/i386/i586/i686/ia32** souvent dits **32 bits** (majorité entre 1990 et 2010), **x64/amd64/x86_64** souvent dits **64 bits** (majorité aujourd'hui) produits principalement par Intel, AMD, VIA

Raspberry Pi, smartphones, Mac (depuis 2021), certaines tablettes ou convertibles : processeurs ARM **armhf, arm64/aarch64** produits par Broadcom, Qualcomm, Apple ou autres

Cas particulier des Mac : série d'ordinateurs d'Apple, à processeur PowerPC dans les années 1990, Intel x86_64 de 2006 à 2021, arm64 (M1, M2... produits par Apple) depuis 2021

Autres : MIPS, s390x

On parle souvent de 32 bit ou 64 bit dans le monde des PCs mais il y a aussi par exemple des processeurs ARM 32 bits (e.g. armv6hf) et ARM 64 bits (e.g. aarch64), et autres...

• Plateforme

Désigne en général le système d'exploitation, e.g. Windows, Linux, macOS

=> Ces différents termes n'ont malheureusement pas vraiment une définition universelle et leur signification varie donc selon le contexte...

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, disque dur, mémoire flash



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](https://www.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 possible 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

Voir aussi https://www.ensta-bretagne.fr/lebars/Share/PC_choice_check-list.txt

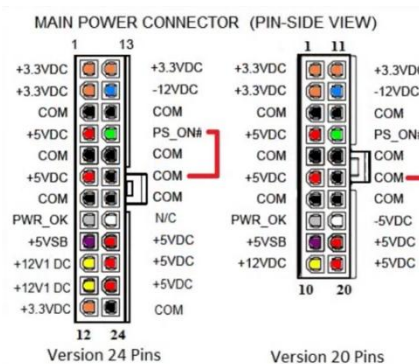
Communiquer, contrôler à distance un système informatique autonome

■ 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 le BIOS/UEFI (si besoin)



Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :

- BIOS ou UEFI :

- Utilitaire de configuration de la carte mère, intégré dedans et lancé dès sa mise sous tension

- C'est lui qui va choisir de lancer le boot loader du système d'exploitation/operating system (OS) selon les disques durs présents et l'ordre de priorité configuré

- Depuis environ 2010, la majorité des PC ont un UEFI pouvant optionnellement se faire passer pour un BIOS (mais cette option est en train de disparaître), de plus la clé de produit Windows y est souvent enregistrée



Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

• BIOS ou UEFI :



Si le BIOS ou 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 ou UEFI!

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

- BIOS ou UEFI :



Touches d'accès classiques (à appuyer/rappuyer à la mise sous tension) :

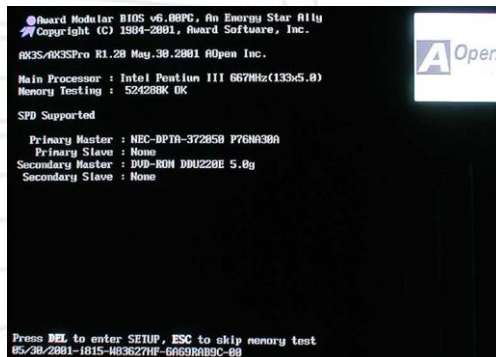
DEL, F2 Enter Setup : page principale de réglages

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

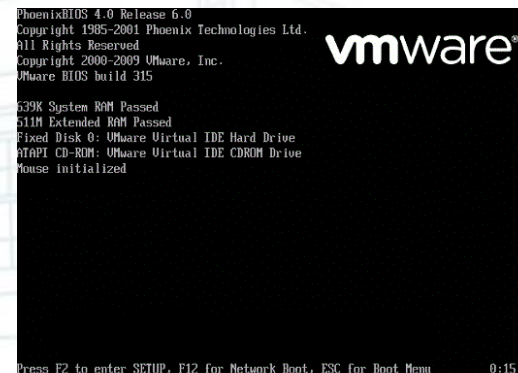
ESC, F8, F10, F11, F12 Boot Menu Override : démarrage sur un périphérique temporaire (voir e.g. sec. 3 <https://q4os.org/dqa014.html>, voir aussi <https://www.tenforums.com/tutorials/21756-boot-usb-drive-windows-10-pc.html>)

F12 Network Boot

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



```
Asus Mother BIOS v6.00PC, An Energy Star Ally
Copyright (C) 1994-2001, Award Software, Inc.
ASUS/MSKSPRO R1.20 May.30.2001 Award Inc.
Main Processor : Intel Pentium III 667MHz(133x5.0)
Memory Testing : 524288K OK
SPD Supported
Primary Master : NEC-DP1A-372B5B P76M3BA
Primary Slave : None
Secondary Master : DVD-ROM DDU22BE 5.0g
Secondary Slave : None
Press DEL to enter SETUP, ESC to skip memory test
65-20-2001-1015-H036271F-6169A1B3C-00
```



```
PhoenixBIOS 4.0 Release 6.0
Copyright 1985-2001 Phoenix Technologies Ltd.
All Rights Reserved
Copyright 2000-2009 VMware, Inc.
VMware BIOS build 315
639K System ROM Passed
51M Extended ROM Passed
Fixed Disk 0: VMware Virtual IDE Hard Drive
ATAPI CD-ROM: VMware Virtual IDE CDROM Drive
Mouse initialized
Press F2 to enter SETUP, F12 for Network Boot, ESC for Boot Menu 0:15
```

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

- BIOS ou UEFI :



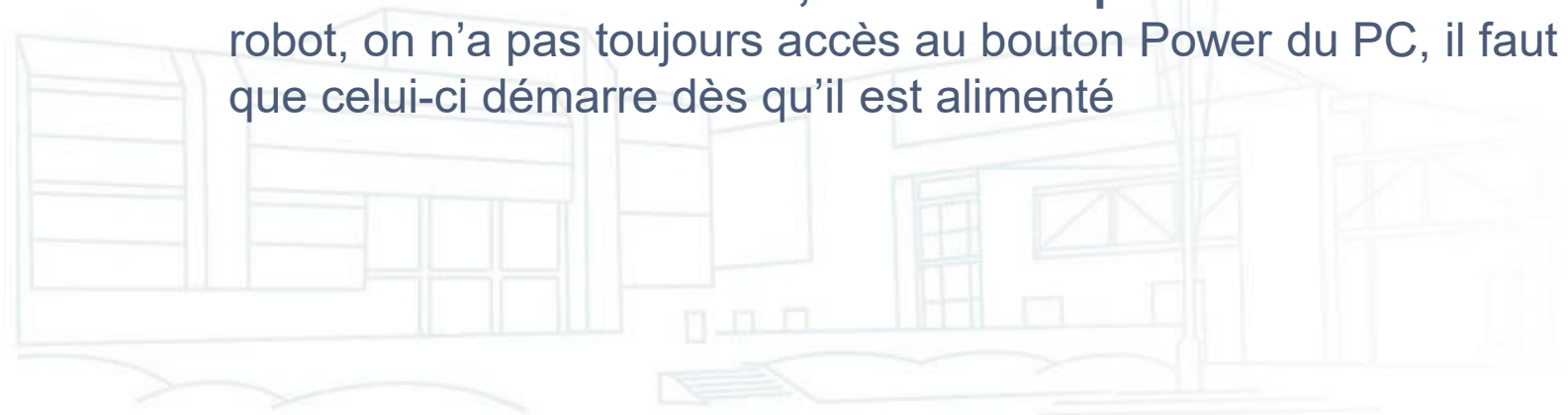
Paramètres à vérifier :

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 ou 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...

Intel VT-x, instructions pour optimiser la **virtualisation** : activer si prévu d'utiliser VMware, Virtual Box, Hyper-V, etc.

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

- BIOS ou UEFI :



Paramètres à vérifier :

AHCI, RAID, IDE, Compatible : RAID n'est pas toujours bien supporté sous Linux (disque non détecté), 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), changer l'option si un OS est déjà installé peut le perturber

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.)

Secure boot : utilisé principalement par Windows (quasi obligatoire à partir de 11) pour empêcher l'installation d'OS ou drivers non validés par Microsoft, désactiver si problèmes d'installation de Linux ou drivers, peut aussi être incompatible avec des versions de [Windows ou Linux pas à jour...](#)

Certains changements peuvent être **interdépendants**, nécessiter **plusieurs redémarrages** ou d'**autres procédures spécifiques** propres au modèle d'ordinateur (e.g. combinaison de touches, mot de passe) 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 côtés 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** (Master Boot Record/secteur de démarrage principal, 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**, pour GUID (Globally Unique IDentifier) Partition Table) et sont notamment compatibles avec des **disques > 2 To** et peuvent aussi proposer de se comporter comme un BIOS avec disques MBR (mais c'est de moins en moins possible)



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 contourner 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-kinda.html>



Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

• Systèmes de fichiers :

Un disque dur (ou équivalent, e.g. SSD) 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 (éventuellement compressée) d'un disque (disque dur, SSD, DVD, etc.)

Montage de partition : attribution d'un chemin d'accès (e.g. **C:**, **D:** sous Windows; **/**, **/media/USB_KEY** sous Linux) 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 :

Un système de fichier forme une arborescence de dossiers et sous-dossiers avec des fichiers comme feuilles, les dossiers n'étant qu'un regroupement de sous-dossiers et fichiers et ne contenant pas de données utiles (à part quelques propriétés telles que les droits d'accès)

Les fichiers par contre contiennent des données utiles (e.g. code binaire exécutable, images, vidéos, texte (i.e. suite d'octets interprétables en suivant généralement le code ASCII, CP-1252 ou Unicode UTF-8 without signature), autres types de données)

Nom de fichier : souvent de la forme **nom.ext**, le .ext s'appelant "extension" et est destiné à caractériser le contenu du fichier (sans garantie qu'il contienne le type de contenu correspondant car l'extension est facilement modifiable...)

Chemin d'accès complet : chemin de la racine jusqu'à l'extension, e.g.
C:\Users\Administrator\Downloads\file.txt, /home/user/Downloads/file.txt

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

• Systèmes de fichiers :

Formats de fichiers classiques (attention: Windows cache souvent les extensions pour lesquelles une action en cas de double-clic existe par défaut) :

.exe : exécutable Windows

.zip .rar .cab .7z .gz .bz2 .xz .zst .tar : fichier contenant un dossier compressé suivant un algorithme particulier (typiquement une compression sans perte d'information), à noter que beaucoup d'autres formats particuliers peuvent faire appel à l'un de ces formats de compression (e.g. .docx, .deb, .nupkg)

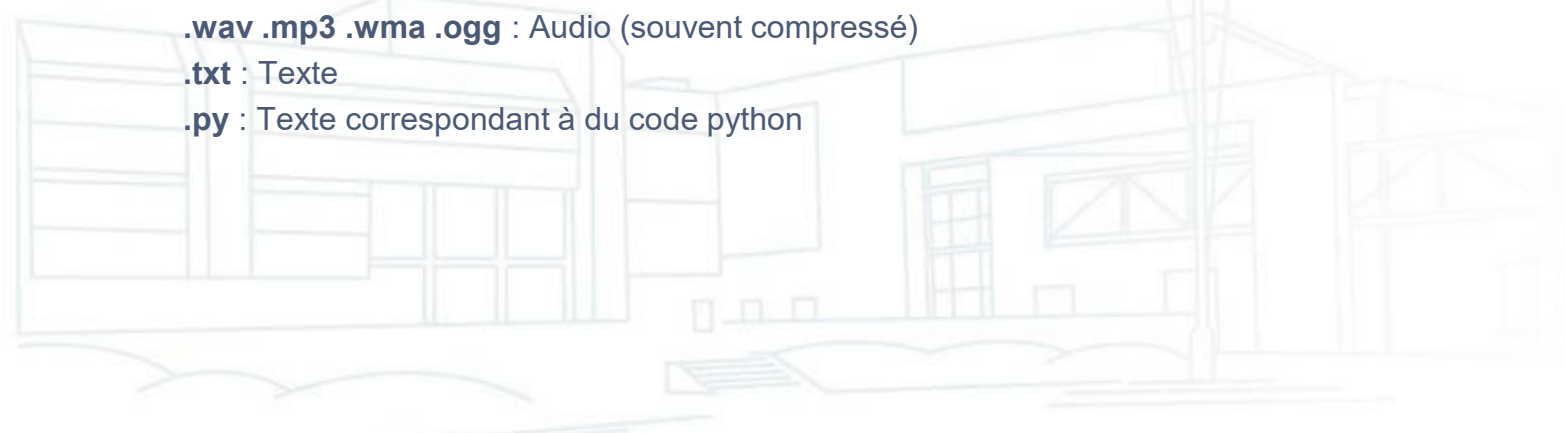
.jpg .png .bmp : Images (parfois compressées avec perte d'information selon le format ou ses options)

.mp4 .wmv .avi .mpeg : Vidéo (souvent compressé)

.wav .mp3 .wma .ogg : Audio (souvent compressé)

.txt : Texte

.py : Texte correspondant à du code python



Communiquer, contrôler à distance un système informatique autonome

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

Données sur les partitions :

Quand on supprime des fichiers (de manière standard) sur une partition, seul la référence vers l'emplacement du fichier est supprimée : le fichier et les données qu'il contient ne sont pas touchés



Pour supprimer vraiment toutes les données sur une partition :

Windows : choisir **low-level format** au moment du formatage, ou e.g. **sdelete -p 1 -z C:**, ou pour supprimer des fichiers spécifiques e.g. **sdelete C:\hiberfil.sys C:\pagefile.sys C:\swapfile.sys**

Linux : si on a déjà supprimé les fichiers de manière standard, **dd if=/dev/zero of=junk bs=1024k count=1000000** va remplir de 0 l'espace libre

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 (de plus ceci peut être souvent fait même si elle est montée)

Mac : les APFS volumes permettent une certaine flexibilité...



Ubuntu Server : voir <https://packetpushers.net/ubuntu-extend-your-default-lvm-space/>

Communiquer, contrôler à distance un système informatique autonome

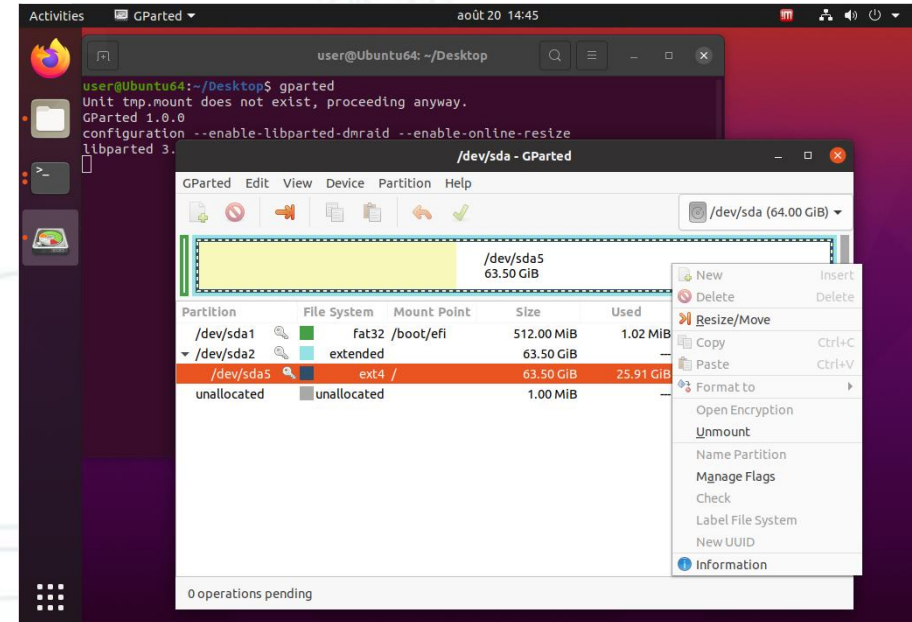
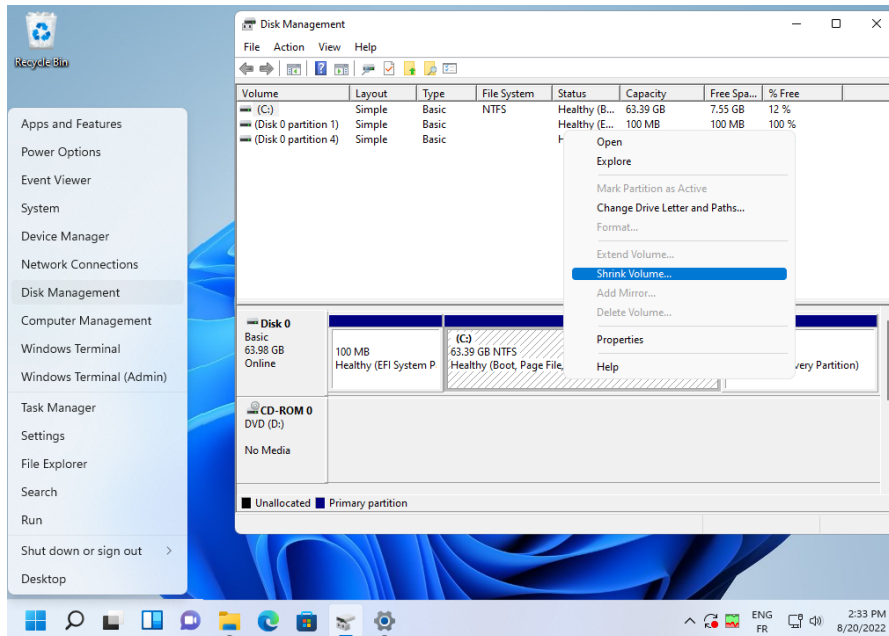
■ Installation d'un PC :

- Systèmes de fichiers :

Manipulations de partitions :

Windows : **Disk Management** (attention, certaines partitions peuvent être cachées), **diskpart**, **mountvol**

Ubuntu : **gparted**, **gpart**, **fdisk**, **/etc/fstab**, **mount**, **unmount**



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 :

- Swap, espace d'échange, mémoire virtuelle, etc.

On appelle **swap** ou encore **espace d'échange** (ou même par abus de langage **mémoire virtuelle** (qui devrait plutôt désigner l'espace RAM+swap)) une zone du disque dur utilisée comme **extension de la mémoire vive (mémoire physique dans la RAM)**. L'idée est que lorsque l'ensemble des programmes lancés a besoin de plus de RAM que ce que le PC dispose, l'OS va quand même essayer de satisfaire leurs demandes d'allocations mémoire supplémentaires en mettant en swap les applications peu utilisées à ce moment, au prix d'un ralentissement dû à la lenteur du disque dur par rapport à la RAM...

Pour **Ubuntu**, il y a parfois une partition complète réservée à cela, avec le système de fichier **linux-swap**, ou un fichier **/swapfile**

Pour **Raspbian**, c'est en général dans un fichier **/var/swap**

Pour **Windows**, c'est dans un fichier **pagefile.sys** ou **swapfile.sys**

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** ou **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 ou GPT valide trouvé**

Le MBR ou 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 et suivants : 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 : https://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

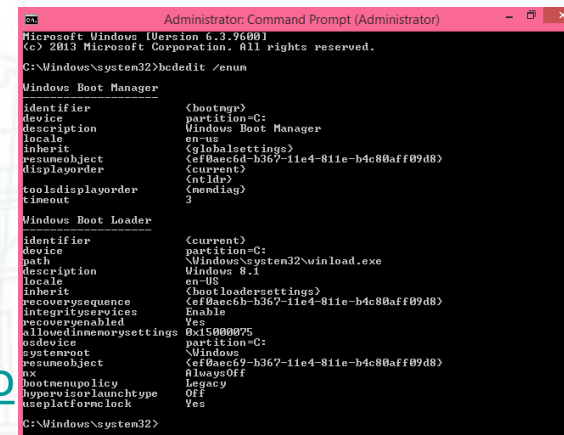
# à partir d'ici à editer pour les autres OS

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

title Le libellé d'OS supplementaires
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
```



```
Administrator: Command Prompt (Administrator)
Microsoft Windows [Version 6.2.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32\bcdedit /enum

Windows Boot Manager
-----
identifiant <bootmgr>
device partition=C:
description Windows Boot Manager
locale en-us
inherits <globalsettings>
resumeobject {ef0aac6d-b367-11e4-811e-b4c80aff09d8}
displayorder <current>
grubid <grub>
toolsdisplayorder <mendiag>
timeout 3

Windows Boot Loader
-----
identifiant <current>
device partition=C:
path \Windows\system32\winload.exe
description Windows 8.1
locale en-US
inherits <bootloadersettings>
recoverysequence {ef0aac6d-b367-11e4-811e-b4c80aff09d8}
recoveryenabled Enable
Yes
allowedinmemorysettings 0x15000075
osdevice partition=C:
systemroot \Windows
resumeobject {ef0aac6d-b367-11e4-811e-b4c80aff09d8}
legacyimageoff Legacy
bootmenupolicy OFF
hypervisorlaunchtype OFF
useplatformlock Yes

C:\Windows\system32>
```

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

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

Vocabulaire

Programme : ensemble d'opérations destinées à être exécutées par un ordinateur

Les termes **application** et **logiciel** sont souvent utilisés de manière interchangeable, on peut dire que c'est un groupe de fichiers exécutables et non exécutables (au moins 1 fichier doit être exécutable). On peut donc dire que c'est un ensemble de programmes

Bibliothèque : fichier (.dll pour Windows, .so pour Linux) pas directement exécutable mais qui contient du code exécutable pouvant être appelé par un fichier exécutable



Communiquer, contrôler à distance un système informatique autonome

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

Vocabulaire

Il y a des applications BIOS ou UEFI (quand l'OS n'est pas lancé, e.g. boot loader), des applications Windows (quand l'OS Windows est lancé), des applications Linux (e.g. quand l'OS Ubuntu est lancé)... Ces différents types d'applications ne sont pas forcément construites de la même façon et sont compilées en assembleur, C/C++, etc. avec des compilateurs différents



Communiquer, contrôler à distance un système informatique autonome

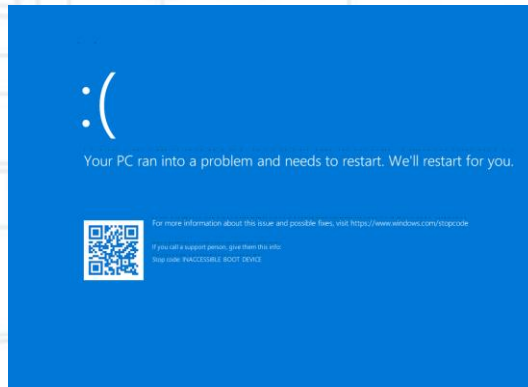
■ Installation d'un PC :

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

Vocabulaire

Noyau/kernel : logiciel central d'un OS, en collaboration avec les drivers il se réserve en général un espace mémoire bien séparé du reste

Pilote/driver : logiciel gérant la **communication bas niveau** entre le **noyau** et un type de **périphérique** spécifique. Lorsque l'un d'entre eux bugge de manière irrécupérable, le noyau de Windows lance l'affichage d'un « **écran bleu** » (« **BSOD** »), sous Linux on a plutôt un message « **kernel panic** ». Dans les cas où le noyau n'a pas su détecter le bug (ou a buggué lui-même), l'ordinateur se retrouve alors dans un état indéfini se traduisant en général par un écran figé, ou arrêt/redémarrage immédiat, etc.



```
Process swapper (pid: 0, threadinfo ffffffff0045a000, task ffffffff00314b60)
Stack: ffffff045a000000 0000000000000000 000001373d030500 ffffff045a030d00
0000000000000007 0000002c00000001 ffffff045a030500 ffffff045a030500
0000000000000001 0000000000000000 000000000000002c 000000169d0dc77
Call Trace:
<IRQ> [ffffffffff801dab24] :vmxnet3:vmxnet3_rq_destroy_all+0x780/0x10c8
[ffffffffff801dab7c] :vmxnet3:vmxnet3_rq_destroy_all+0x740/0x10c8
[ffffffffff8000ca51] net_rx_action+0xac/0x1b1
[ffffffffff80012576] __do_softirq+0x89/0x133
[ffffffffff8005e2fc] call_softirq+0x1c/0x28
[ffffffffff80064636] do_softirq+0x2c/0x7d
[ffffffffff80064c66] do_IRQ+0xee/0xf7
[ffffffffff8006bd43] default_idle+0x0/0x50
[ffffffffff8005d615] ret_from_intr+0x0/0xa
[<EIP>] [ffffffffff8006be1c] default_idle+0x29/0x50
[ffffffffff8040f999] cpu_idle+0x35/0xa8
[ffffffffff80465089] start_kernel+0x220/0x225
[ffffffffff8046522f] _sinit_text+0x22f/0x236

Code: 0f 0b 60 e6 ea 1d 80 c2 a6 85 41 83 7d 00 02 74 0a 0f 0b 60
RIP: [ffffffffff801d9bac] :vmxnet3:vmxnet3_tq_xmit+0xa90/0x1208
RSP: [ffffffffff804a9c30]
<0>Kernel panic - not syncing: Fatal exception
```

Communiquer, contrôler à distance un système informatique autonome

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

Vocabulaire

L'ensemble **kernel+drivers** fournit alors un certain nombre de **fonctions systèmes utilisables** par les **services et applications fournies avec l'OS**, ces derniers ayant pour but de fournir un maximum de fonctionnalités utilisables directement par l'utilisateur ou les applications installées manuellement

Kernel mode vs user mode : on peut dire que les **drivers** sont des **applications en kernel mode**, dans un espace mémoire kernel spécifique, alors que le reste des **applications** de l'OS et installées par l'**utilisateur** sont **en user mode**, dans un autre espace mémoire user...



Communiquer, contrôler à distance un système informatique autonome

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

Vocabulaire

Par abus de langage, on appelle aussi parfois driver un logiciel gérant un protocole de périphérique sans pour autant être en contact direct avec le noyau, e.g. pour les périphériques à port série RS232, un driver de port RS232 est en général fourni avec l'OS, mais il reste à écrire un logiciel pour décoder le protocole spécifique du périphérique...

=> Il est compliqué de fournir une définition précise aux différents termes cités, les frontières et appellations varient selon les OS, leurs versions, les visions des développeurs, leur politique commerciale, etc.



Communiquer, contrôler à distance un système informatique autonome

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

Vocabulaire

Service : **type d'application un peu spécifique** qui respecte certaines contraintes e.g. pouvoir répondre à une demande start, stop, restart, pause définie par le système...

Shell : **application principale** d'un OS **pour l'interaction avec l'utilisateur**. Historiquement, ce n'était qu'une application attendant des commandes textuelles tapées au clavier et affichant une réponse textuelle (appelé aussi **invite de commandes** ou encore **terminal**) mais de nos jours, c'est plutôt une **GUI (interface graphique/Graphical User Interface)** complexe comprenant un concept de bureau et gestionnaire de fenêtres, barre de tâches et notifications, lanceur d'applications, explorateur de fichiers, etc.

Communiquer, contrôler à distance un système informatique autonome

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

Invite de commandes/terminal:

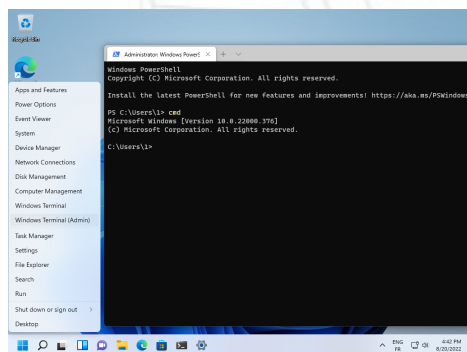
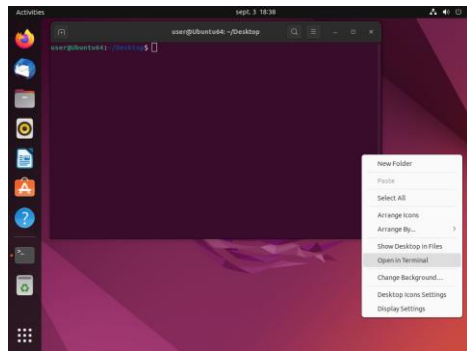
Linux: **bash** (par défaut sous Ubuntu), sh, zsh, etc.

Notes: les appels aux fichiers du dossier courant doivent en général être précédés de **.**, les paramètres des commandes sont souvent de la forme **-a -b value -cvalue -d=value --param**, etc., en particulier **--help**, **-h** servent souvent à afficher de l'aide, **--version** la version de la commande, **-v** la version ou la quantité de détails textuels affichés... Le séparateur de dossiers est **.**. **cd chemin_complet_ou_relatif** permet de changer le dossier courant. Scripts de commandes : souvent fichiers **.sh** mais pas forcément...
Commentaires : **#**.

Windows: **cmd** (Invite de commandes/Command Prompt), **PowerShell** (par défaut dans les versions récentes de Windows)

Notes: les appels aux fichiers du dossier courant doivent en général être précédés de **.** pour PowerShell (pas obligatoire pour cmd), les paramètres des commandes sont souvent d'une forme similaire à Linux mais avec **.** au lieu de **-**, en particulier **/?**, **/help**, **/h**, **-?**, **-help**, **-h**, **--help** servent souvent à afficher de l'aide, ou parfois aussi c'est exactement comme Linux... Le séparateur de dossiers est **.**. **cd chemin_complet_ou_relatif** permet de changer le dossier courant (attention: pour **cmd** il faut rajouter le paramètre **/d** si la destination est sur une autre partition). Scripts de commandes : fichiers **.bat** ou **.cmd** pour **cmd**, **.ps1** pour **PowerShell**.
Commentaires : **rem** ou **::** pour **cmd**, **#** ou **<#** et **#>** pour **PowerShell**.

Pour tous, la touche **TAB** permet en général de faire de l'autocomplétion, ainsi il faut faire attention en recopiant une liste de commandes depuis un fichier et en la collant dans le terminal, les tabulations présentes pouvant être interprétées ainsi, attention aussi aux autres caractères spéciaux évoqués si apparaissent dans des noms de fichiers, de plus certaines commandes peuvent aussi « consommer » ce qui est collé si elles attendent des commandes venant de l'utilisateur...



Communiquer, contrôler à distance un système informatique autonome

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

Vocabulaire

Variable d'environnement : variable dans l'OS. **%PATH%** (Windows) ou **\$PATH** (Linux) est une variable contenant la liste des dossiers (séparés par ';' pour Windows, ':' pour Linux) où chercher un exécutable lorsque son nom est tapé dans un terminal. Pour Linux, **\$LD_LIBRARY_PATH** existe en plus et est spécifique aux bibliothèques .so.

Cache : zone mémoire (e.g. dossier) stockant des données susceptibles d'être réutilisées plusieurs fois et dont la récupération depuis leur zone d'origine (e.g. Internet) pourrait être longue

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!



=> Utiliser le **mode sans échec/safe mode** et autres options de réparation **Windows** (voir <https://support.microsoft.com/en-us/windows/start-your-pc-in-safe-mode-in-windows-10-92c27cff-db89-8644-1ce4-b3e5e56fe234>, touche **F8** avant Windows 8, éventuellement **bcdedit /set {bootmgr} displaybootmenu yes** pour forcer l'affichage du menu de choix à chaque démarrage, voire **bcdedit /set {globalsettings} advancedoptions true**, **bcdedit /set {default} bootmenupolicy legacy** (default is **standard**)*), touche **ESC** pour Ubuntu (appuyer de manière rapide et répétée sur la touche quand le BIOS ou UEFI a fini son chargement et lance l'OS (si on appuie trop tôt on risque d'aller dans des options du BIOS ou UEFI))



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

* Bien préciser explicitement le BCD store du disque interne dans bcdedit si besoin de forcer le mode sans échec depuis Windows PE ou Windows Recovery

Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :

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

A noter qu'après le 1^{er} démarrage de l'OS, il se peut que tout ne soit pas encore initialisé donc il vaut mieux parfois attendre quelques min et redémarrer avant de faire d'autres choses...



Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

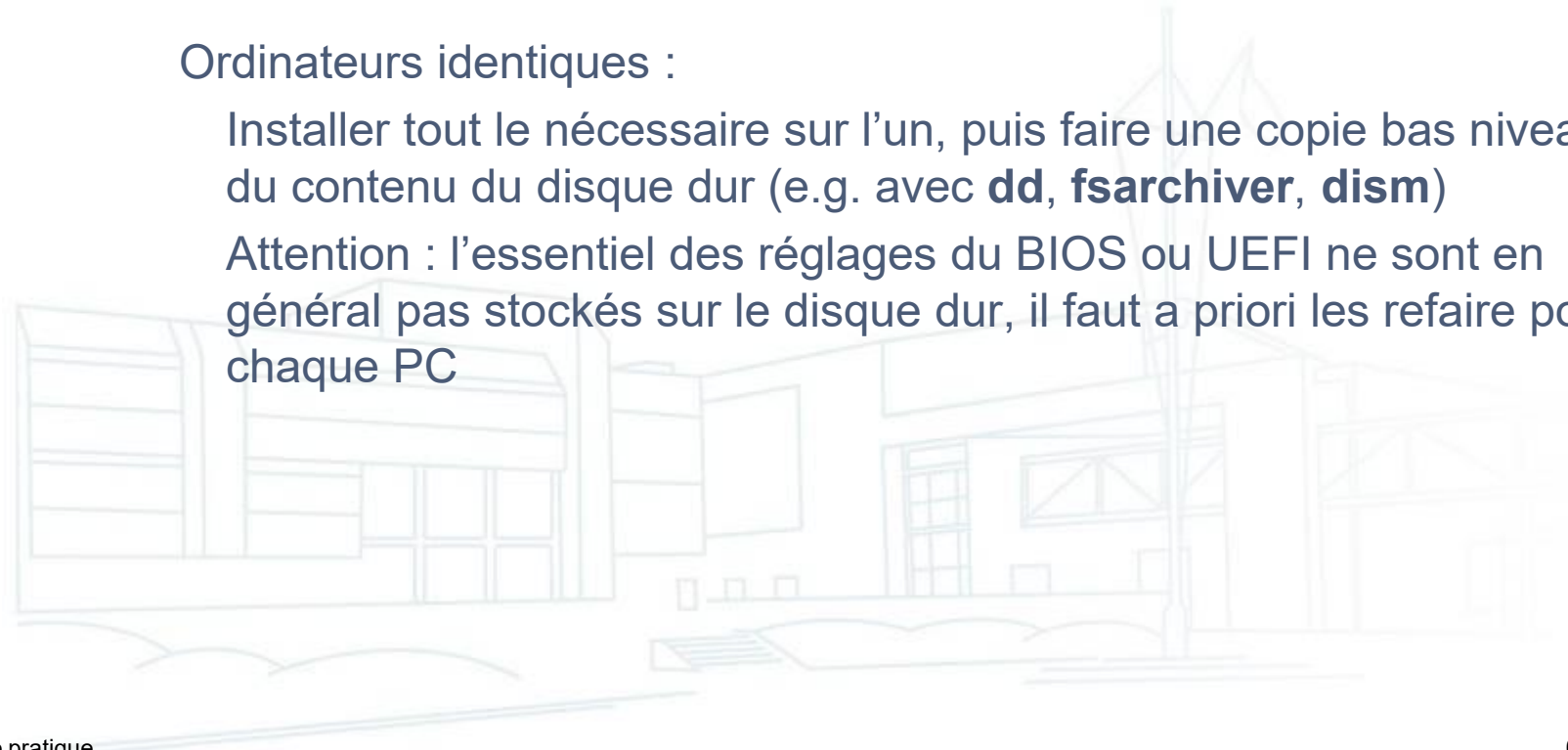
- Automatisation de l'installation d'OS, installations en masse :

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 : l'essentiel des réglages du BIOS ou UEFI 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/AIK, gestionnaire de paquets **choco** 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**, gestionnaire de paquets **apt**.

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

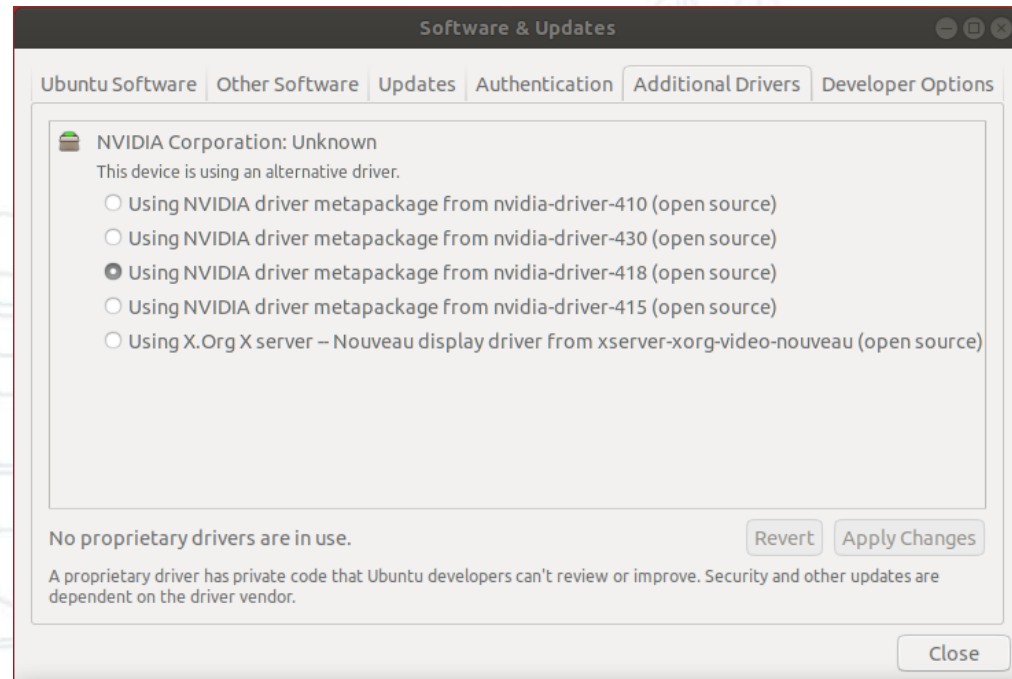
■ Installation d'un PC :

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

Ordinateurs hétérogènes :

Drivers spécifiques (au cas où l'OS ne les trouve pas automatiquement ou s'ils ne fonctionnent pas bien) :

- En général téléchargeables depuis le site du fabricant pour Windows
- Pour Ubuntu :



Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

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

Un **gestionnaire de paquets** est un outil souvent en ligne de commande permettant d'**installer, mettre à jour, supprimer** facilement une suite de **logiciels ou bibliothèques** dans des versions compatibles entre elles **en tenant compte** automatiquement **de leurs dépendances** éventuelles

En général, les logiciels (« paquets ») sont téléchargés depuis Internet

Si un paquet n'est pas disponible pour le logiciel qu'on souhaite (et qu'il n'y a pas d'installateur par ailleurs), il faut souvent installer une chaîne de compilation C++, identifier et récupérer les dépendances, les mettre dans les bons dossiers, régler d'éventuelles variables d'environnement, compiler le tout dans le bon ordre, avec les bonnes options, etc. Voir aussi https://www.ensta-bretagne.fr/lebars/tutorials/Complements_C-C++.pdf

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

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

Gestionnaires de paquets

Exemples :

apt : le plus important sous Ubuntu

choco : Windows n'en ayant pas vraiment par défaut, celui-ci peut être installé et est le plus connu

winget : potentiel futur officiel pour Windows, encore assez limité w.r.t. choco en 2023

snap : autre présent sous Ubuntu

pip : propre à Python



vcpkg : pour du code source de bibliothèques C++ (Windows, Linux, macOS)



ROS : en est aussi un d'une certaine façon



OSGeo4W : pour logiciels en lien avec la geoscience sous Windows

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

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

Gestionnaires de paquets

Ubuntu

sudo apt update # Met à jour sa base de données

sudo apt upgrade # Met à jour les paquets à leur dernière version

sudo apt install build-essential # Installe le paquet « build-essential » (g++, etc.)



Ubuntu garde dans **/var/cache/apt/archives** les paquets .deb téléchargés, ceci peut être utile quand on veut installer les mêmes paquets (sans garantie cependant que tous y soient) 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



Paquets potentiellement utiles : voir <https://www.ensta-bretagne.fr/lebars/Share/unattended.sh>

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

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

Gestionnaires de paquets



Windows

Installer choco : voir <https://chocolatey.org/install>

choco install mingw --version=13.2.0 # Installe le paquet « mingw » (g++, etc.)

choco pin add -n mingw # Empêche la mise à jour du paquet « mingw »

choco list # Liste les paquets installés

choco outdated # Liste les paquets pouvant être mis à jour

choco upgrade all # Met à jour les paquets à leur dernière version

Paquets potentiellement utiles : voir <https://www.ensta-bretagne.fr/lebars/Share/Unattended.bat>

Attention, comme il y a des **limitations sur le nombre de téléchargements concurrents** il vaut mieux être sur une **connexion Internet individuelle**, voir <https://docs.chocolatey.org/en-us/community-repository/community-packages-disclaimer#rate-limiting>

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

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

Gestionnaires de paquets

Risques de problèmes de version des logiciels si plusieurs gestionnaires de paquets utilisés pour installer les mêmes logiciels (attention notamment à leurs dépendances éventuelles)...

Désinstallations

Certains logiciels ont un programme d'installation qui peut faire diverses choses, souvent nécessite les droits admin (mais parfois peut être installé juste pour l'utilisateur courant, sans nécessiter droits admins), et qu'il faut **désinstaller proprement** (ce qui dépend de la manière dont ils ont été installés...) et en général, **pas juste supprimer des dossiers** au hasard...

Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
 - Automatisation de l'installation d'OS, installations en masse :
Désinstallations

Windows

Settings > Apps > Installed Apps

Si échec de la désinstallation, utiliser **regedit** pour chercher l'app dans
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
et **HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall**
et le supprimer de la liste et éventuellement supprimer manuellement les dossiers liés à l'app si on est sûr qu'ils ne servent pas encore à d'autres choses...

choco uninstall -y my_package

winget uninstall my_package

Ubuntu

sudo apt remove my_package

Voir aussi cas avec **snap**, **flatpack**...

Si installé avec **make install**, essayer avec **make uninstall** dans le dossier original de compilation du projet...

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

- Paramètres des OS et applications :

Windows

%UserProfile% : dossier utilisateur, e.g. C:\Users\Administrator

%SystemRoot%=C:\Windows : dossier principal de l'OS

%SystemRoot%\System32, **%SystemRoot%\SysWOW64** : .dll et .exe respectivement 64 bit et 32 bit de l'OS

%ProgramFiles%, **%ProgramFiles(x86)%** : programmes installés par les administrateurs respectivement 64 bit et 32 bit

%AppData%, **%LocalAppData%** : fichiers de configuration et parfois programmes propres à l'utilisateur courant

Registre, via commande **regedit** : contient de très nombreux paramètres de configuration de l'OS et certaines applications

%TMP%, **%TEMP%** : fichiers temporaires

Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
 - Paramètres des OS et applications :

Ubuntu

~ : dossier utilisateur, e.g. /home/user

/root : dossier du super-utilisateur

/etc : fichiers de configuration pour l'OS et communs à tous les utilisateurs

/bin, /sbin : programmes de l'OS, sbin étant pour ceux devant être exécuté avec les droits du super-utilisateur (e.g. via sudo)

/lib : .so de l'OS

/usr : programmes installés via apt

/usr/local, /opt : programmes installés par les utilisateurs

/dev : fichiers représentant les périphériques

/media, /mnt : points de montages pour les partitions

/var : fichiers de log et fichiers temporaires

/tmp : fichiers temporaires

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :



- Lancement automatique de programmes au démarrage de l'OS

Linux :

Configurer l'**auto-login** et **Startup Applications** dans menu Ubuntu, ou fichier `~/.bashrc`

Services avec **systemctl**

cron

Fichiers dans `/etc/init.d...`

Windows :

Configurer l'**auto-login** et **C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp**

Task Scheduler

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

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :



- Démarrage/arrêt/configuration de services

Ubuntu :

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

Eventuellement désactiver le téléchargement et installation automatique sur un système autonome (mais manuellement mettre à jour quand même au moins tous les mois si connecté à Internet!)

Sur un PC en multiboot Windows+Ubuntu, on peut avoir des problèmes d'horloge quand on passe de l'un à l'autre, ce qui peut poser problème notamment pour les mises à jour, pour essayer d'empêcher ce problème :



Disable UTC and use Local Time in Ubuntu to avoid conflicts with Windows

sudo timedatectl set-local-rtc 1 --adjust-system-clock

Il faudra peut-être encore corriger l'heure 1 ou 2 fois après redémarrage sous l'un et l'autre

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

- Problèmes de droits d'accès, blocages, sécurité, etc.

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**

Pour les exécutables, bien activer l'attribut spécifique (**chmod +x a.out** sous Linux)

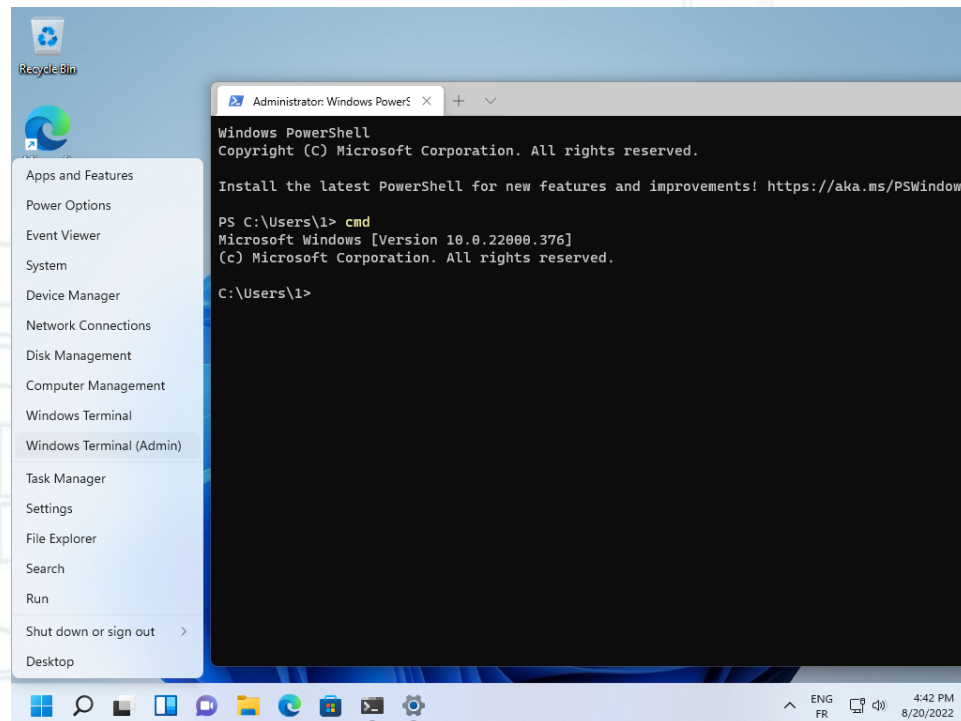


Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

- Problèmes de droits d'accès, blocages, sécurité, etc.

Administrator Terminal sous Windows : utiliser **cmd** pour les commandes bas niveau (sauf mention contraire) car **PowerShell** interprète parfois mal certains caractères spéciaux (e.g. <https://stackoverflow.com/questions/41030701/unable-to-edit-with-bcdedit-filelds-in-powershell-cmd-exe-command-line-fails>)...



Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

- Problèmes de droits d'accès, blocages, sécurité, etc.

Fast startup : à désactiver dans les options d'alimentation de Windows (voir <https://www.tenforums.com/tutorials/4189-turn-off-fast-startup-windows-10-a.html>) car ceci peut empêcher l'accès aux partitions en dehors de Windows (provoque une mise en veille prolongée hybride qui verrouille l'accès aux partitions quand on arrête le PC)

Dans la même logique, le problème se produit aussi quand Windows est mis en veille, de même si Windows a été arrêté brutalement (dans ce cas, **fsck** peut aider)

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

- Problèmes de droits d'accès, blocages, sécurité, etc.

BitLocker/Device encryption : certains PCs peuvent avoir le cryptage de fichiers partiellement activé automatiquement. **Ceci peut empêcher l'accès aux partitions en dehors de Windows ou certains changements peuvent bloquer son démarrage** en demandant une clé de récupération. La clé est généralement associée au(x) compte(s) Microsoft utilisé(s) sur l'ordinateur, voir <https://support.microsoft.com/fr-fr/windows/recherche-de-votre-cl%C3%A9-de-r%C3%A9cup%C3%A9ration-bitlocker-dans-windows-6b71ad27-0b89-ea08-f143-056f5ab347d6>

Si des données ont été cryptées et que les clés de récupération/décryptage sont perdues, les données sont totalement perdues car non décryptables!



Pour désactiver BitLocker pour la partition **C:** : use the command **manage-bde -off C:** and check regularly decryption progress with **manage-bde -status** (may take several minutes, hours...). If you need to shutdown or restart the computer while the decryption is still in progress, use the commands **manage-bde -pause C:** before shutdown/restart, and then **manage-bde -resume C:...**

Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
 - Problèmes de droits d'accès, blocages, sécurité, etc.



TPM :

Un TPM (Trusted Platform Module) est un microcontrôleur spécialisé dans certaines tâches de cryptage, décryptage, authentification

Il peut être requis par certains OS ou logiciels (e.g. Windows 11)

Il peut contribuer à certains blocages pouvant entraîner la perte totale de données si certains changements logiciels ou matériels sont effectués (e.g. si contient des clés de décryptage BitLocker qui n'auraient pas été sauvegardées par ailleurs)



Communiquer, contrôler à distance un système informatique autonome

- Installation d'un PC :
 - Problèmes de droits d'accès, blocages, sécurité, etc.



Drivers non signés sous Windows :

Sous Windows 10 et suivants, les développeurs de drivers doivent les faire valider et signer par Microsoft et ceci est payant. Certains drivers peuvent donc être non signés et bloqués par défaut par Windows, mais il est en général possible de les installer en passant par le **mode sans échec**, voir <https://support.microsoft.com/en-us/windows/start-your-pc-in-safe-mode-in-windows-92c27c27-cff-db89-8644-1ce4-b3e5e56fe234>



Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

- Problèmes de droits d'accès, blocages, sécurité, etc.

If problems accessing a serial port device on Linux, add current user to **dialout** group (vi /etc/group), reboot if needed:

sudo adduser \$USER dialout

Voir aussi <https://www.ensta-bretagne.fr/lebars/Share/Ubuntu.txt> pour renommer un port série selon le périphérique branché (**udev** rules)



Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

• Virtualisation

Il existe des programmes (e.g. **VMware Workstation**, Virtual Box, Hyper-V, qemu) qui permettent de simuler/émuler le fonctionnement d'un ou **plusieurs PC virtuels sur un PC réel**, dans les limites des capacités du PC réel

Tips VMware :

Pour utiliser efficacement la fonction **Compact** pour réduire la taille des fichiers contenant les disques virtuels, il faut parfois créer des partitions temporaires dont l'espace libre est rempli de 0 (e.g. avec **dd if=/dev/zero of=junk bs=1024k count=1000000**)

Communiquer, contrôler à distance un système informatique autonome

■ Installation d'un PC :

- Voir plus d'infos dans le sujet de TD correspondant :

Voir : https://www.ensta-bretagne.fr/lebars/tutorials/TD_wifi_rtk_multiboot_rpi.pdf

- Multiboot Windows+Ubuntu pour VA ROB :

Voir : https://www.ensta-bretagne.fr/lebars/computer_prereq_rob.pdf

- Initiation aux réseaux, configuration pour l'accès à distance, vérification des blocages possibles, etc.

Voir : https://www.ensta-bretagne.fr/lebars/initiation_reseaux.pdf



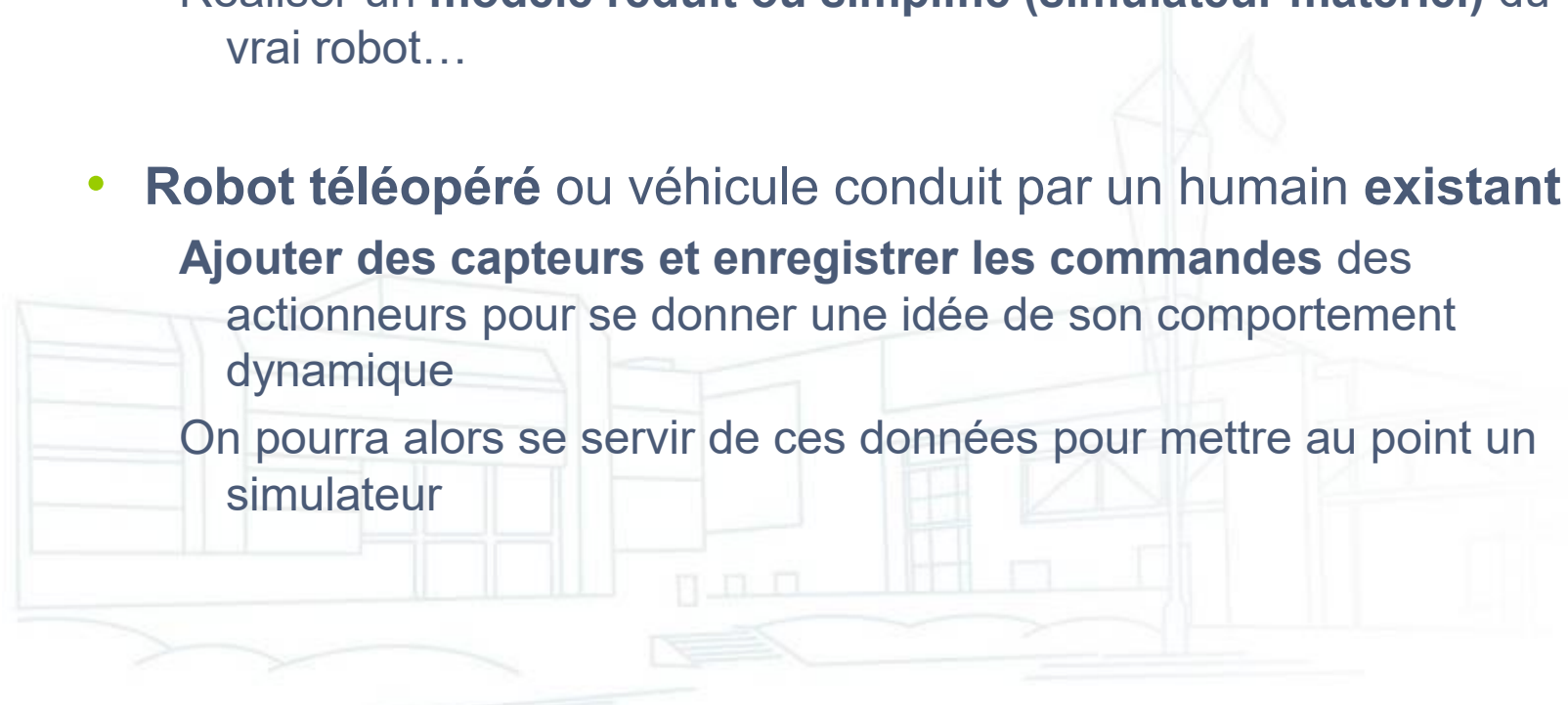


Algorithmes pour l'autonomie

- 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



Algorithmes pour l'autonomie

■ Simulation :

- Intérêts de faire d'abord des simulations :

Moins de risque de casse

Débuggage en général plus facile

etc.

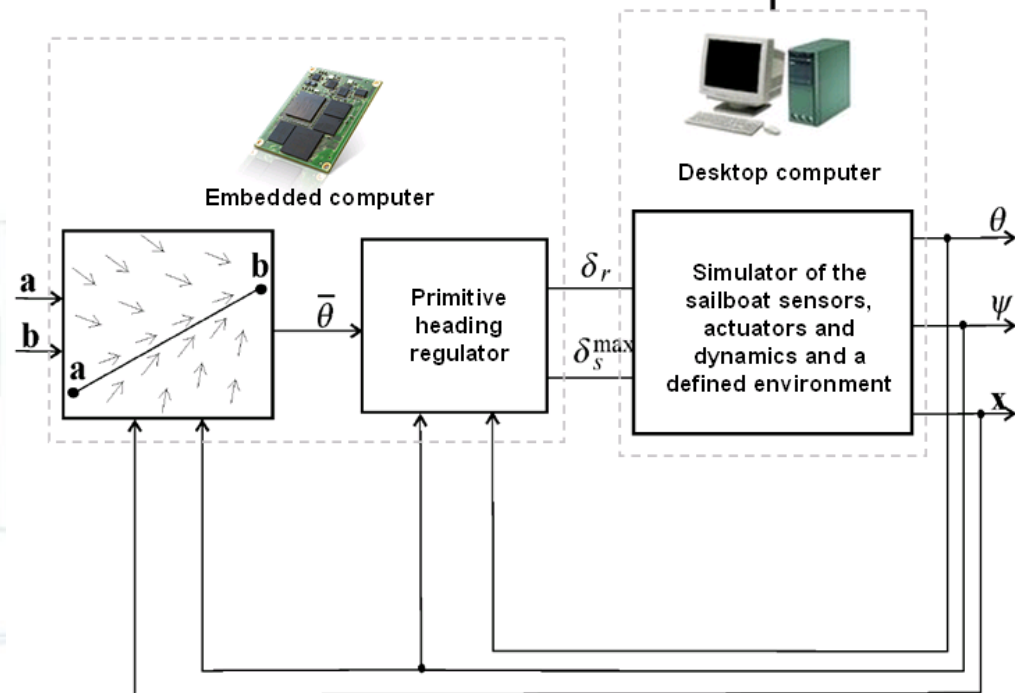
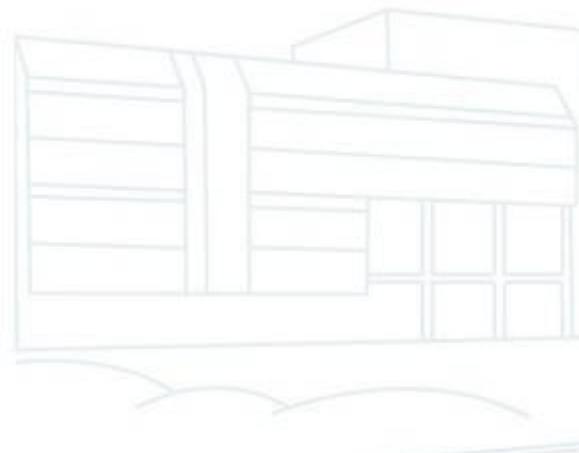
- Inconvénients :

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



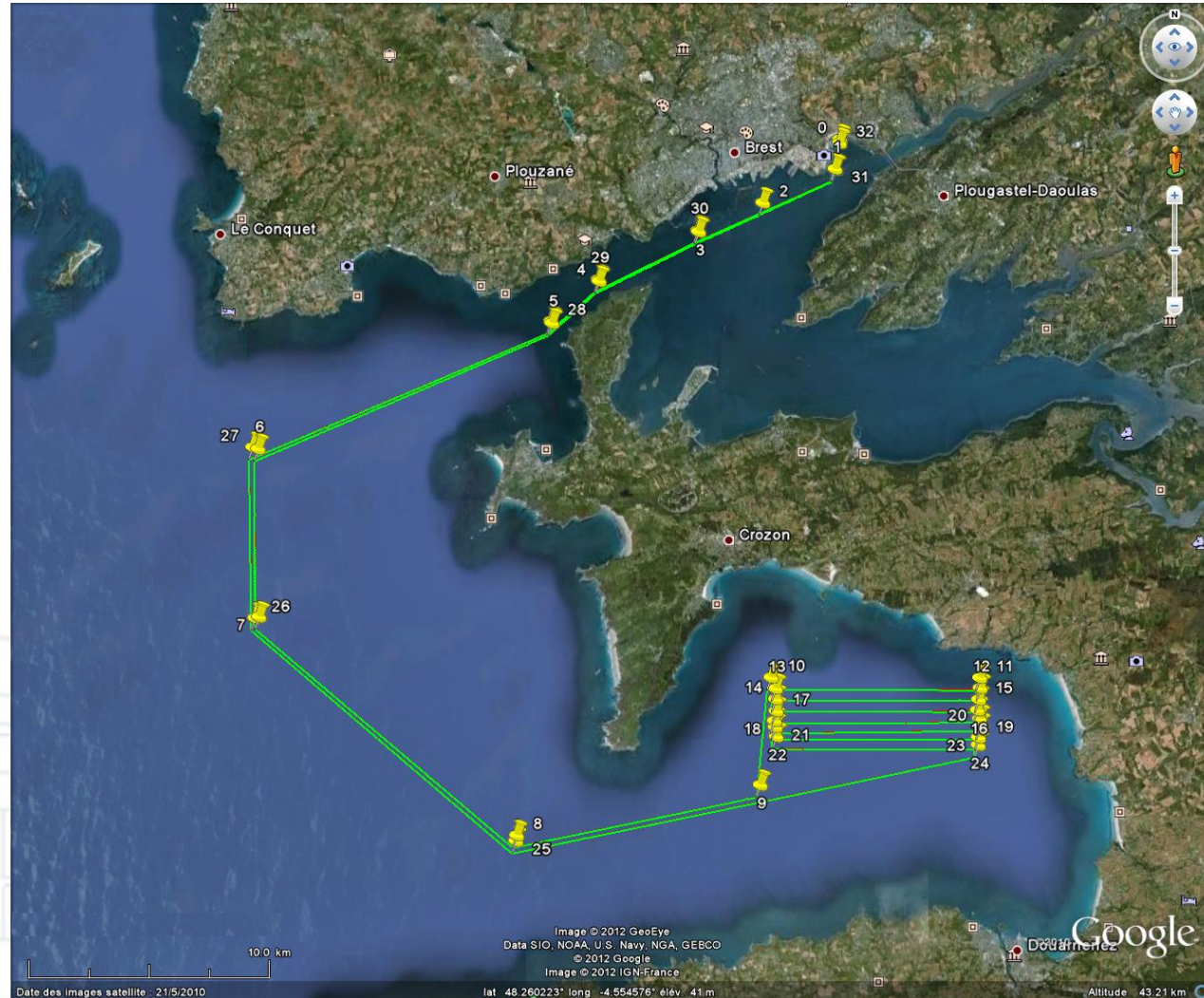
Algorithmes pour l'autonomie

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



Algorithmes pour l'autonomie

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

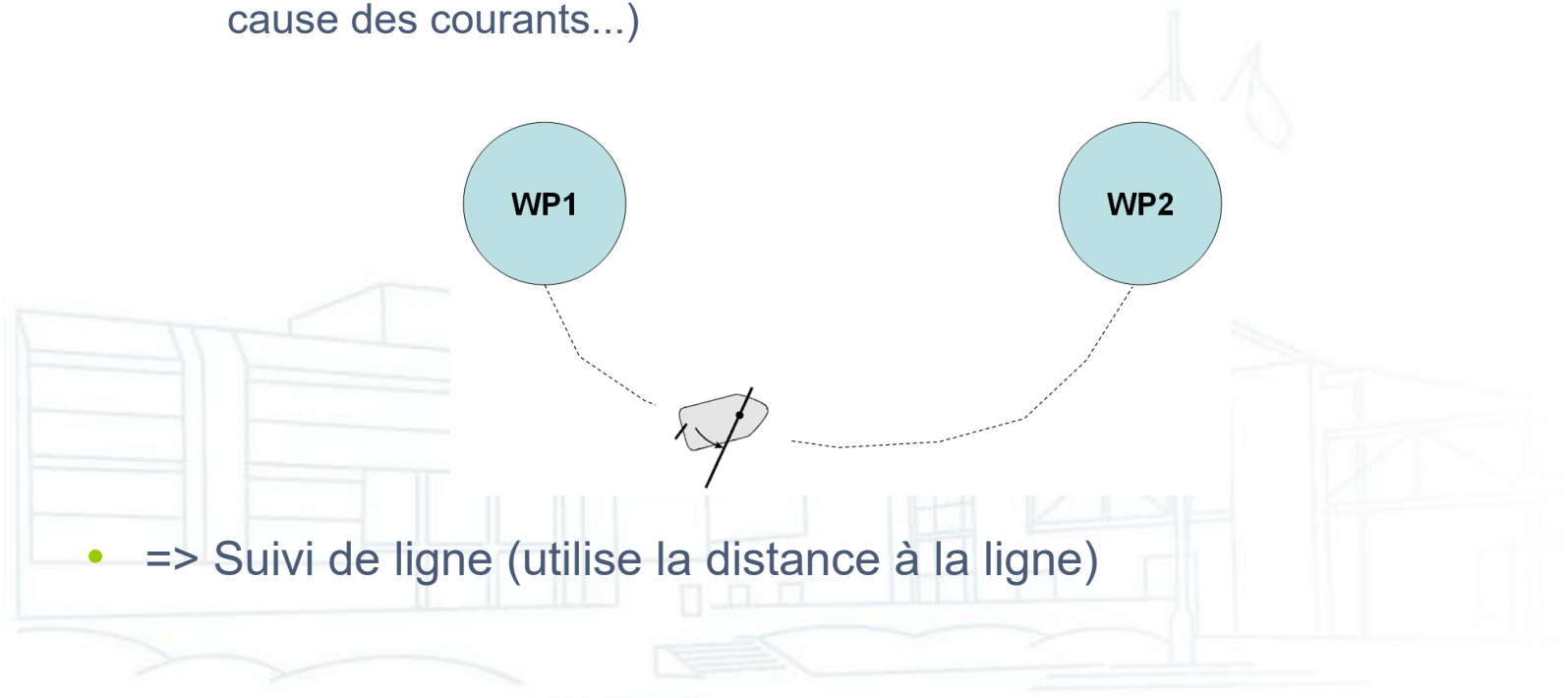


Algorithmes pour l'autonomie

- Simulation de robots en MATLAB :
 - Voir https://www.ensta-bretagne.fr/lebars/simu_robots_matlab.pdf
- Régulation simple de robot pour du suivi de waypoints :
 - Voir https://www.ensta-bretagne.fr/lebars/buggy_control_simple.pdf



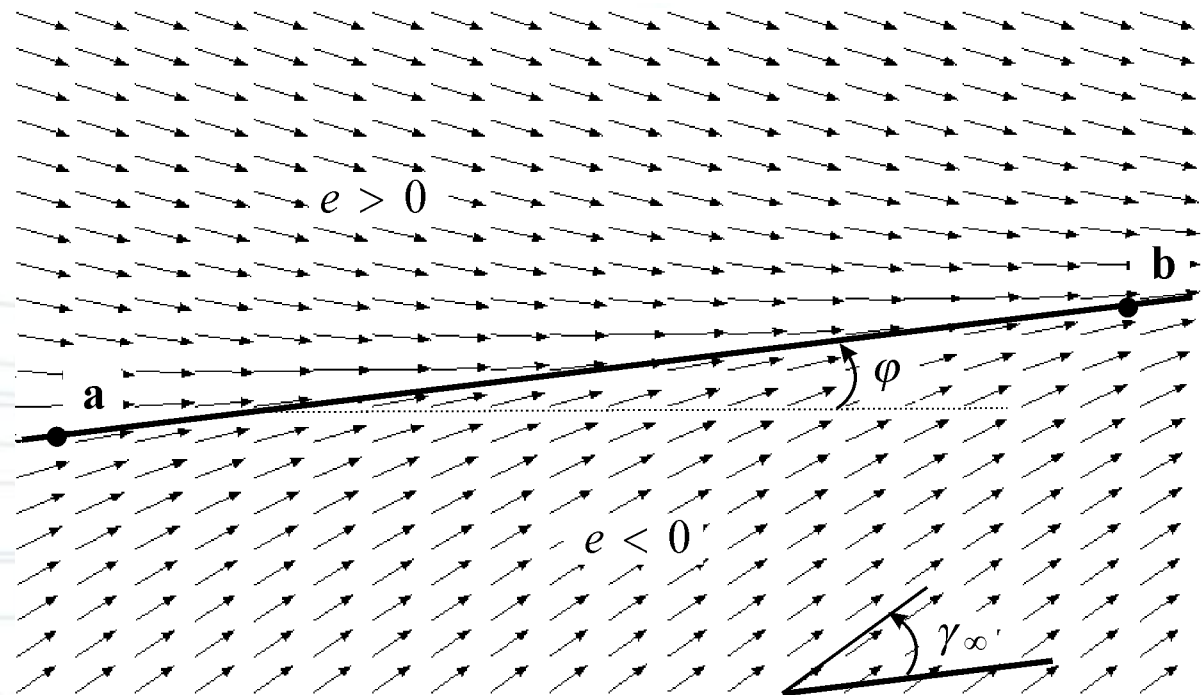
- Du suivi de waypoints au suivi de ligne :
 - Suivi de waypoints de base :
 - Le robot cherche à suivre un cap dans la direction du waypoint
 - Le waypoint est validé quand on est dans un rayon prédéfini
 - Problème : rien n'empêche qu'on dérive entre waypoints (e.g. sur l'eau à cause des courants...)



■ Suivi de ligne :

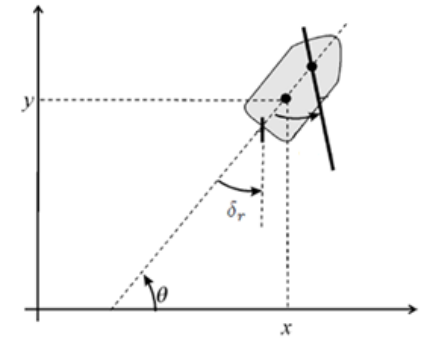
- Le cap à suivre va être la ligne entre les 2 waypoints courants, avec un angle d'attraction à la ligne en fonction de notre écart à la ligne

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



Voir https://www.ensta-bretagne.fr/lebars/Share/line_following.zip

- Suivi de ligne :
 - Un module de régulation 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 régulation en cap en fonction de la ligne courante

$$e = \det\left(\frac{\mathbf{b}-\mathbf{a}}{\|\mathbf{b}-\mathbf{a}\|}, \mathbf{m}-\mathbf{a}\right)$$

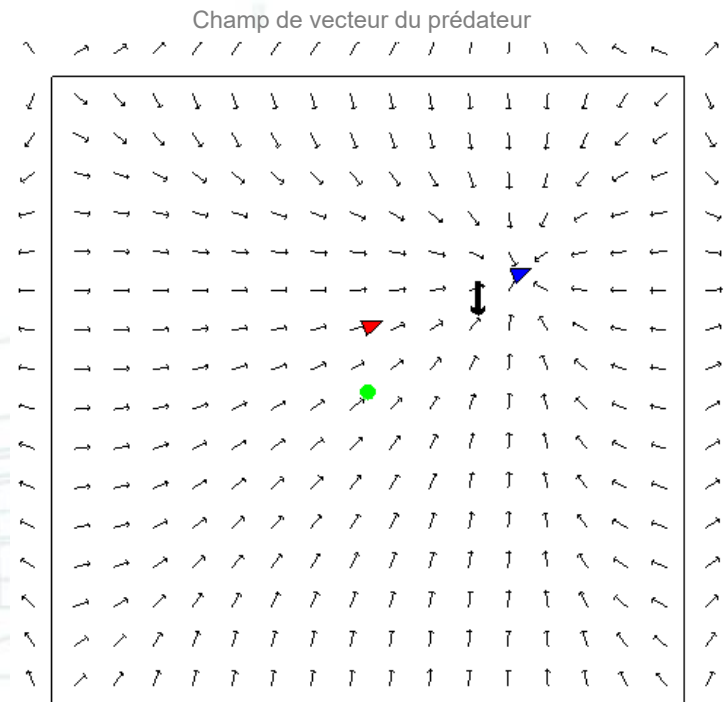
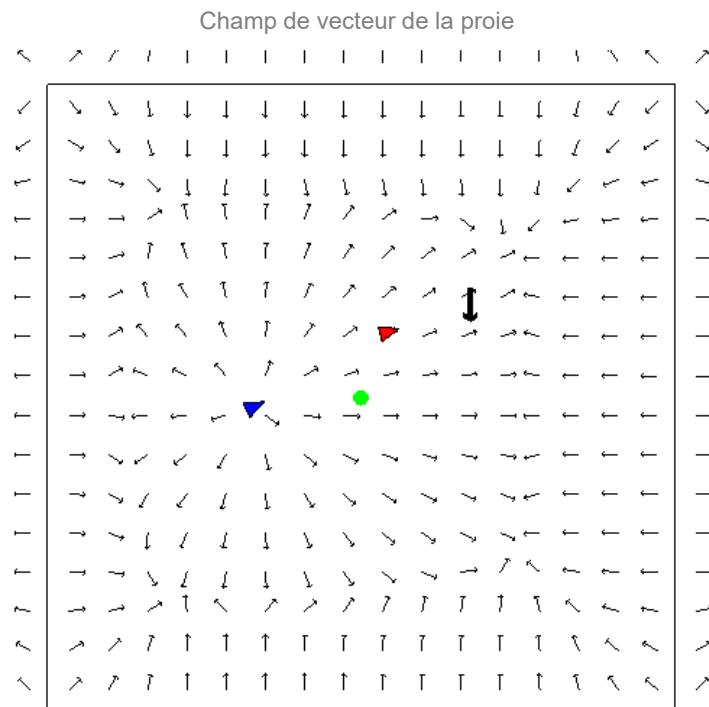
$$\varphi = \text{atan2}(\mathbf{b}-\mathbf{a})$$

$$\bar{\theta} = \varphi - \frac{2 \cdot \gamma_{\infty}}{\pi} \cdot \text{atan}\left(\frac{e}{r}\right)$$

- Un module de gestion de la navigation envoie des lignes au superviseur et décide 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



Algorithmes pour l'autonomie

- Autres régulations :
 - Voir e.g. <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

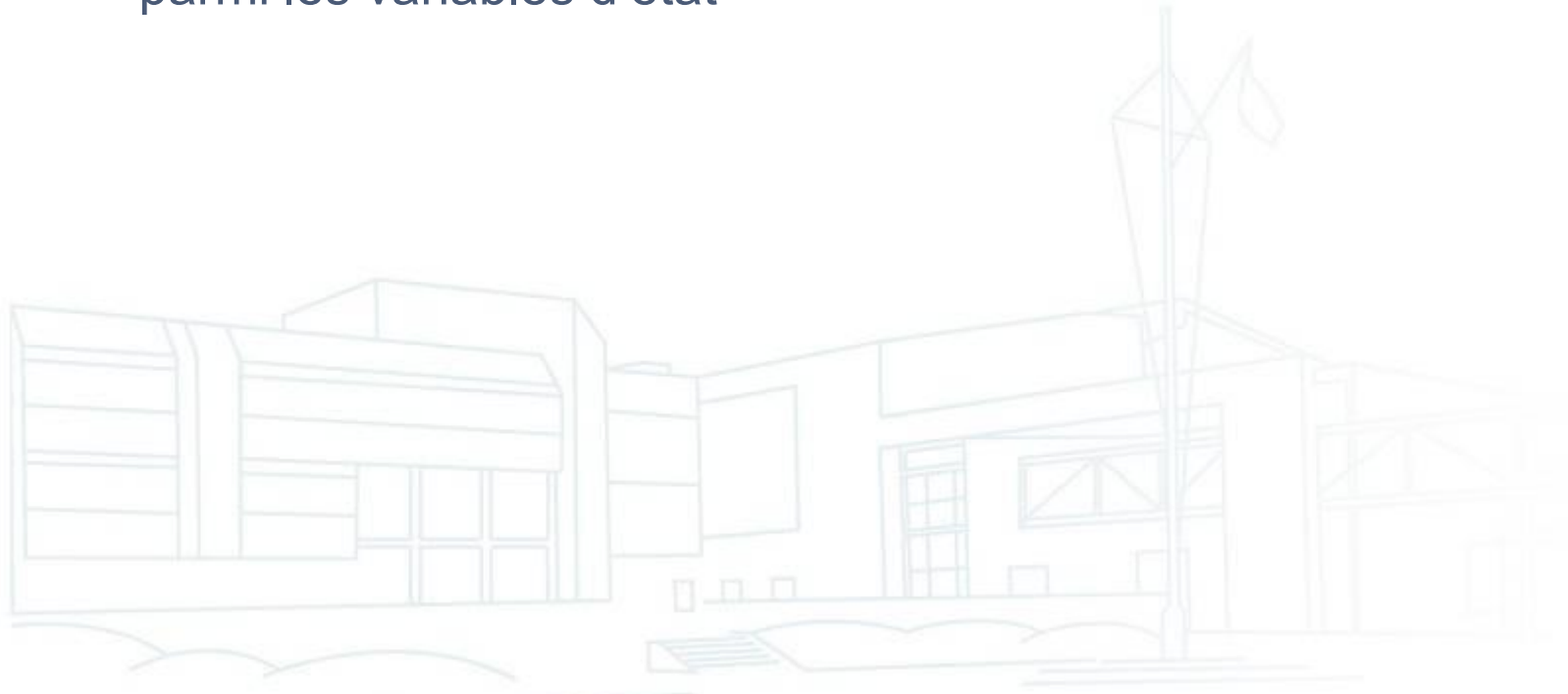
Calculs par relations géométriques : relations de distance et/ou d'angles à des amers (objets de position plus ou moins connue)

Filtre de Kalman : voir e.g. https://www.ensta-bretagne.fr/lebars/Obs_et_filt_de_Kalman.pdf

Méthodes ensemblistes : voir e.g. <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



Algorithmes pour l'autonomie

- Localisation :
 - Conversions de repères courantes :

Repères courants :

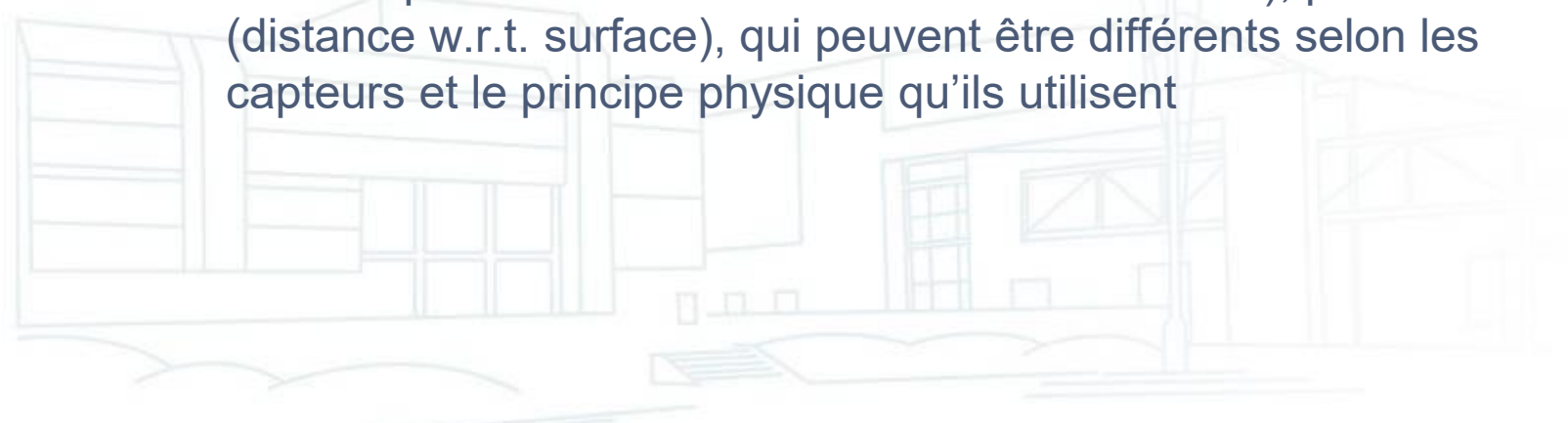
NED (North-East-Down)

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

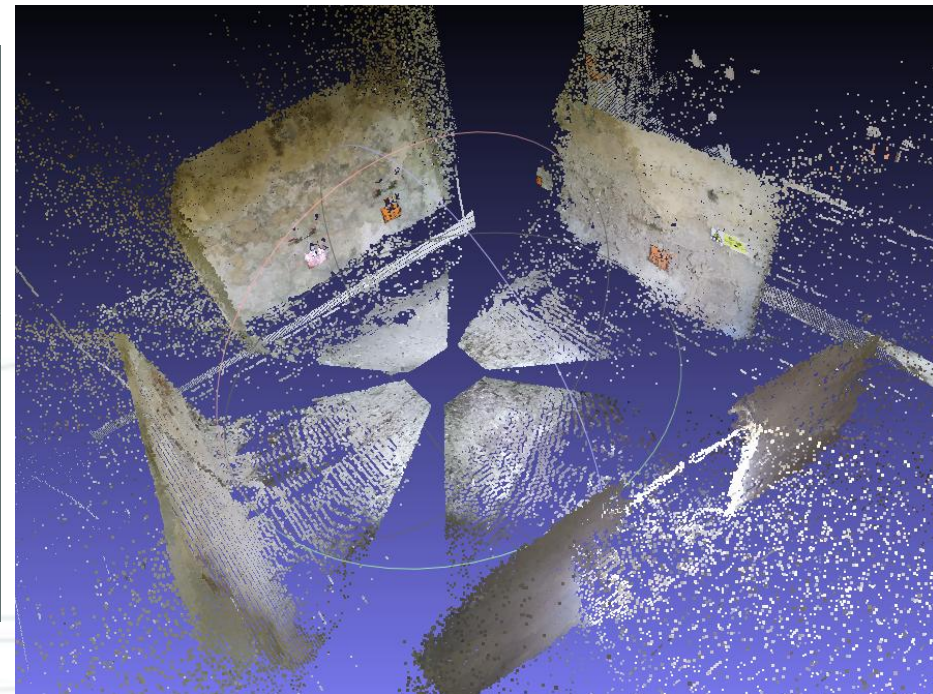
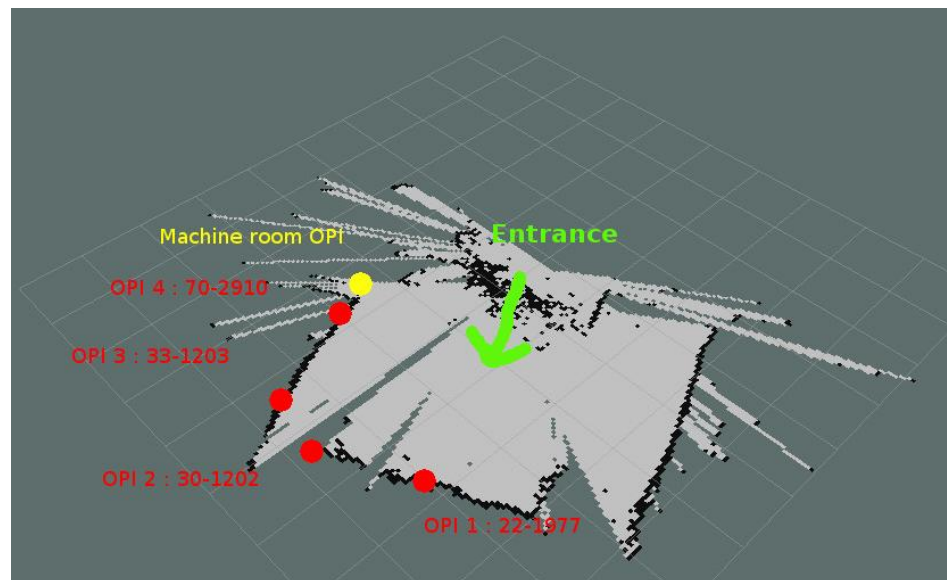


Algorithmes pour l'autonomie

- Localisation :
 - Conversions de repères courantes :
 - NED->ENU** : YPR=90-yaw, -pitch, roll
 - NWU->ENU** : YPR=90+yaw, pitch, roll
 - WGS84** vers repère **ENU local** : voir https://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, 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 :

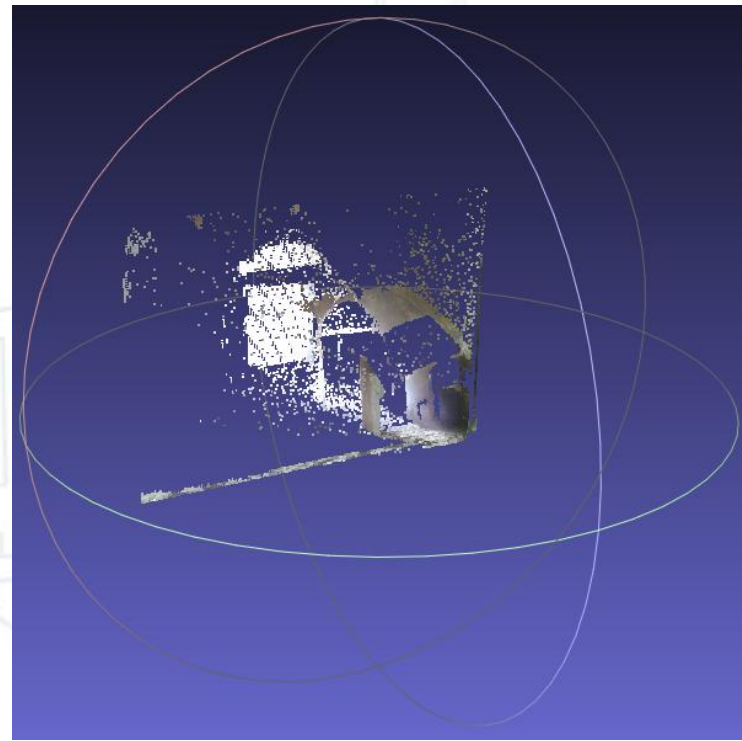
- Point de vue légèrement différent -> zones visibles sur un capteur mais pas sur l'autre
- Résolution et ouverture angulaire (FOV) différentes
- Symétries, etc.



■ Cartographie :

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

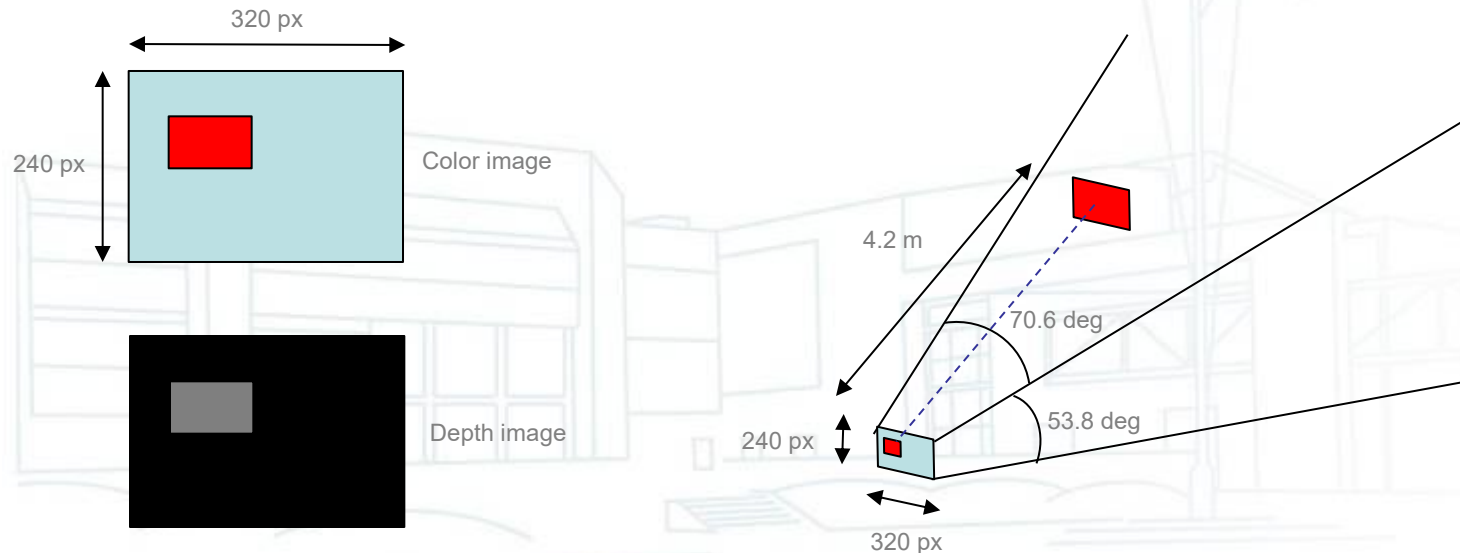
Ensuite, connaissant l'ouverture angulaire de la caméra et la portée en profondeur, on peut combiner les informations de l'image couleur et l'image en profondeur pour dessiner un nuage de points colorés dans l'espace...



■ Cartographie :

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

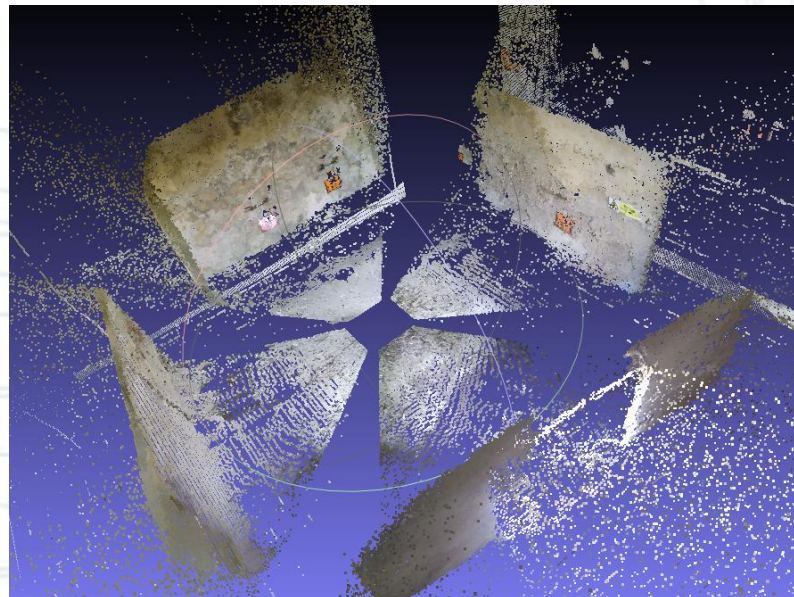
Exemple : représentation dans l'espace d'un **groupe de pixels rouges** de l'image couleur, dont le **niveau de gris** correspondant dans l'image en profondeur se traduit par une **distance de 4.2 m**, si images e.g. de **résolution 320 x 240** avec un **FOV 70.6 x 53.8**



Algorithmes pour l'autonomie

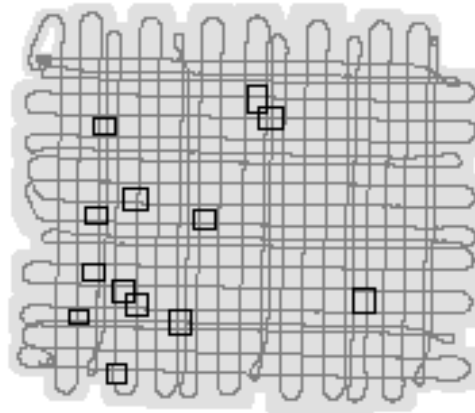


Robot ayant pris des données Kinect
en tournant 4 fois de 90 deg



Algorithmes pour l'autonomie

- 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 amer qu'on voit est un nouvel amer ou un amer 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



Algorithmes pour l'autonomie

■ Divers :

- Voir plus d'infos dans le sujet de TD correspondant :

Voir : https://www.ensta-bretagne.fr/lebars/tutorials/TD_waypoints_object_tracking_kinect2.pdf

- Utilisation de bibliothèques spécifiques en C/C++

Voir : https://www.ensta-bretagne.fr/lebars/tutorials/Complements_C-C++.pdf

- OpenCV

Voir <https://www.ensta-bretagne.fr/lebars/opencv.pdf>



