

Objets connectés et robotique

L3 IFA, S5

Novembre 2025

Vincent Lannurien ¹

`vincent.lannurien@univ-brest.fr`

Inspiré du cours de Yvon Autret et Jean Vareille

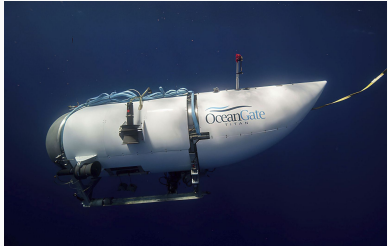
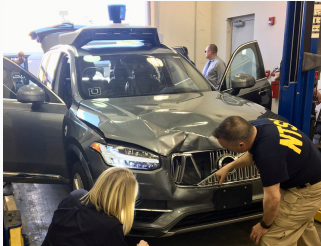
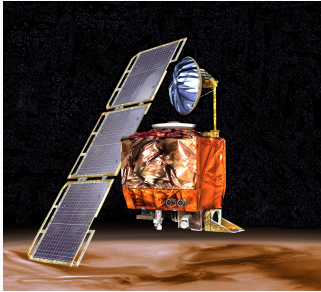
¹ Lab-STICC, CNRS UMR 6285, Université de Bretagne Occidentale, Brest



Université de Bretagne Occidentale



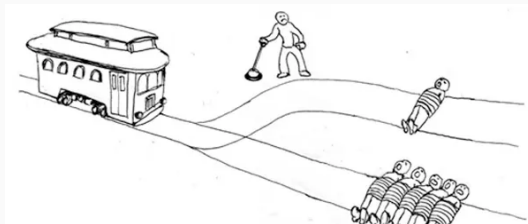
Introduction



- https://en.wikipedia.org/wiki/Mars_Climate_Orbiter
 - problème d'unités (système métrique vs système impérial)
- https://en.wikipedia.org/wiki/Ariane_flight_V88
 - erreur de conversion (*overflow*)
- https://en.wikipedia.org/wiki/14th_Quartermaster_Detachment
 - bug d'arrondi (*overflow*)
- <https://en.wikipedia.org/wiki/Therac-25>
 - erreur de programmation concurrente (*race condition*)
- https://en.wikipedia.org/wiki/Death_of_Elaine_Herzberg
 - détection d'obstacle inopérante
- https://en.wikipedia.org/wiki/Titan_submersible_implosion
 - certifications de sécurité insuffisantes

Commençons par un test...

- *Trolley problem* (https://en.wikipedia.org/wiki/Trolley_problem) : le dilemme du tramway



Que feriez-vous ? → <https://www.moralmachine.net/hl/fr>

Plan de la présentation

1. Organisation de l'UE
2. Projet à réaliser
3. Contexte
4. Informatique embarquée
 - Microcontrôleurs
 - Arduino
 - Programmation
 - Entrées/Sorties
5. Robotique
 - Généralités
 - Commande à distance
 - Interface utilisateur
 - Comportement autonome
 - Calcul déporté



Organisation de l'UE

- Objectifs :
 - Savoir concevoir des systèmes cyberphysiques simples, autonomes
 - Comprendre les compromis entre performance (latence, débit) et consommation (énergie, mémoire)
- Méthode :
 - Le plus de **manipulations** possible!
 - Dimension **matérielle** (un peu d'électronique, un peu de mécanique)
 - Dimension **logicielle** (de l'informatique!)

- Volume horaire :
 - CM : 8h
 - TP : 14h
- Travail en groupes :
 - Binômes dans les groupes de TP
 - Deux binômes par robot
- Examen :
 - Démo lors des TP
 - Rendu de projet
- Intervenants :
 - Vincent Lannurien – vincent.lannurien@univ-brest.fr
 - Alan Le Boudec – alan.leboudec@univ-brest.fr

Projet à réaliser

Objectifs :

- Prise en main de l'environnement de développement
- Implantation d'algorithmes simples sur le robot
- Calcul déporté

Tout ce que l'on ne va pas faire :

- Électronique : montages complexes, conception de carte...
- Traitement du signal : radio, interférences...
- Sécurité : authentification, détection d'intrusion...

Objectifs :

- Prise en main de l'environnement de développement
- Implantation d'algorithmes simples sur le robot
- Calcul déporté

Tout ce que l'on ne va pas faire :

- Électronique : montages complexes, conception de carte...
- Traitement du signal : radio, interférences...
- Sécurité : authentification, détection d'intrusion...

Étapes du projet

- Commande : Fonctionnement de base d'un robot
 - Envoi de messages : avancer, reculer, tourner, etc.
 - Communications : filaire (série), sans-fil (radio)
 - Interface web de commande : Node.js, JavaScript
- Environnement : capteurs et actionneurs
 - Utilisation d'un capteur de distance
 - Fonctionnement des moteurs
- Contrôle : Programmation de missions simples
 - Comportement autonome :
 - Avancer jusqu'à trouver un obstacle
 - Avancer jusqu'à trouver une ouverture
 - Faire un quart de tour
 - Contourner un objet
- En option :
 - Plus de capteurs
 - Plus de cartes

Contexte

Definition

*Cyber-Physical Systems (CPS) are integrations of computation with physical processes.*¹

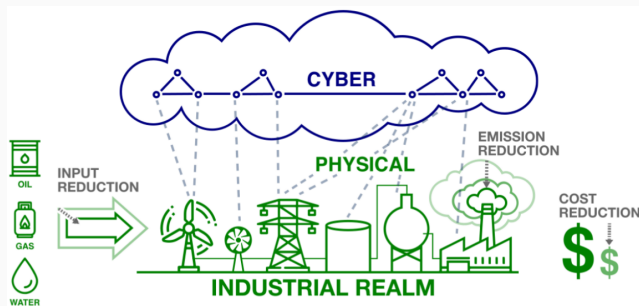
- Énergie : *smart grids*
- Transport : systèmes ferroviaires, aviation civile, automobile autonome...
- Santé : dispositifs médicaux "connectés"
- Industrie : usines automatisées, robots industriels
- Ville "intelligente"

1. Edward A. LEE. « Cyber Physical Systems : Design Challenges ». 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC). 2008.

Définitions – Systèmes cyberphysiques

Definition

Cyber-Physical Systems (CPS) are integrations of computation with physical processes.²

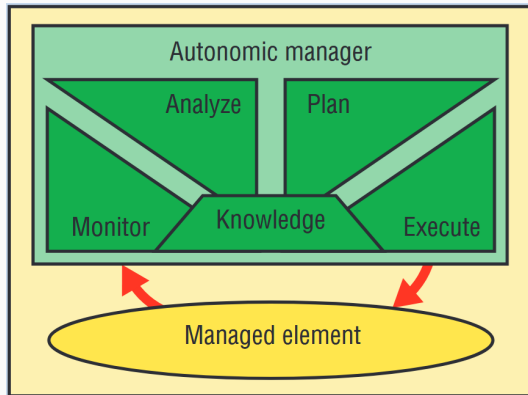


2. Edward A. LEE. « Cyber Physical Systems : Design Challenges ». 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC). 2008.

3. [1] Oliver Inderwildi et al. « The Impact of Intelligent Cyber-Physical Systems on the Decarbonization of Energy ». *Energy & Environmental Science* (2020)

Définitions – Systèmes autonomes

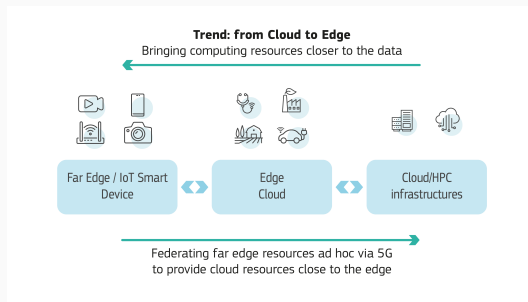
- Robotique, systèmes autonomes⁴ :
 - **Capteurs** : produisent une vue de l'environnement du système
 - **Actionneurs** : commandent le système selon son état et sa consigne
- Objets connectés, *Internet of Things (IoT)* : contrôle, commande à distance



⁴[2] J.O. KEPHART et D.M. CHESS. « The Vision of Autonomic Computing ». *Computer* (2003)

Quelques tendances [réf.]

- 41 milliards d'appareils *IoT* déployés en 2025
- Intelligence artificielle de plus en plus déployée en périphérie de réseau (Apple Intelligence, Google AI Edge)
- Défis : interopérabilité, durabilité



Numérique – Quels impacts? (2022) [réf.]

Pour chaque Français, le numérique représente :

- Consommation électrique : 10%
- Empreinte carbone : 2.5%
- Déchets : 299 kg/habitant/an
- Fabrication : 78% de l'impact
- Utilisation : 21% de l'impact



10 %

de la consommation électrique annuelle vient des services numériques.

Cela représente, pour chaque Français :

- L'équivalent de la consommation électrique d'un radiateur de 1000 W alimenté sans interruption pendant 30 jours.
- Le même impact environnemental qu'un trajet de 2259 km parcouru en voiture.



2,5 %

de l'empreinte carbone de la France est liée au numérique. C'est un peu plus que le secteur des déchets (2 %).



20

millions de tonnes de déchets sont produits par an sur l'ensemble du cycle de vie des équipements à l'échelle de la France. Soit : 299 kg/habitant.



62,5

millions de tonnes de ressources sont utilisées par an pour produire et utiliser les équipements numériques.

QUELS SONT LES OUTILS NUMÉRIQUES LES PLUS IMPACTANTS?



Les terminaux utilisateurs : téléviseurs, ordinateurs, smartphones, tablettes... (65 à 90 % de l'impact environnemental, selon l'indicateur environnemental considéré)



Les centres de données (data center) (entre 4 % et 22 %)



Les réseaux (entre 2 % et 14 %)



78 %

de l'impact environnemental du numérique sur les émissions de gaz à effet de serre est lié à l'étape de fabrication. Celle-ci nécessite une extraction importante de métaux rares et est surtout effectuée dans des pays au mix énergétique fortement carboné.

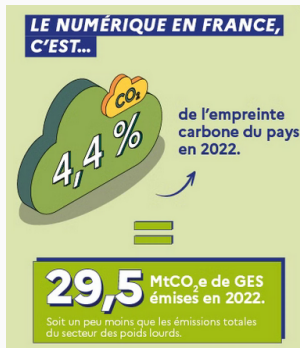


21 %

concerne la phase d'usage.

Des chiffres qui confirment l'importance des politiques visant à allonger la durée d'usage des équipements numériques à travers la durabilité des produits, le réemploi, le reconditionnement, l'économie de la fonctionnalité ou la réparation.

Numérique – Quels impacts? (2025) [réf.]



QU'EN EST-IL DE L'AUDIOVISUEL ?

5,6

MtCO₂e émises par la consommation de contenus audiovisuels en France en 2022 :

TV linéaire, streamings audio et vidéo à la demande..., soit autant que les émissions de 4 041 073 véhicules par an.

+29 %

en 2030, si l'on suit la tendance actuelle (moins de télé en direct, mais plus de vidéos à la demande et de streaming vidéo).



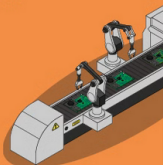
FABRICATION DES ÉQUIPEMENTS : UN IMPACT IMPORTANT

17,8

MtCO₂e de GES émises en 2022. La fabrication de ces équipements est ce qui pèse le plus dans l'impact global du numérique.

117 Mt/an

de ressources mobilisées pour produire ces équipements : métaux, minerais, plastiques, eau, terres excavées, etc.



1,7 t
par an et par personne en France. 🇫🇷

- Hausse de l'impact des **centres de données** : 46% de l'empreinte carbone du numérique

Informatique embarquée

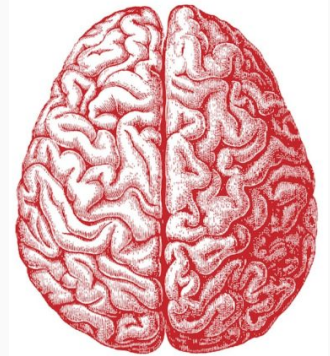
Informatique embarquée

Microcontrôleurs

Microcontrôleurs : Utilisations et limites



- **Circuit intégré** : calcul (processeur), mémoire (mémoire vive et non-volatile), interfaces d'Entrées/Sorties
- **Non-programmable** : son comportement est gravé dans le silicone
- **Utilisation** : partout où un processeur généraliste n'est pas nécessaire
 - + : Plus simple à comprendre, à vérifier
 - + : Consommation d'énergie minimale
 - - : Peu flexible
- **Coût** : en 2018, en moyenne, \$0.03 pour un μc 8-bit, \$1 pour un μc 32-bit $\rightarrow$ ratio $\frac{1}{100}$ par rapport à un PC grand public

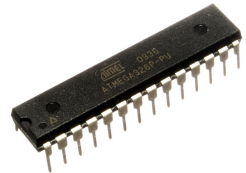
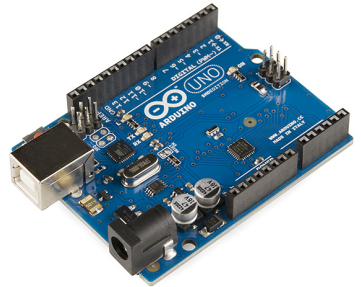


Informatique embarquée

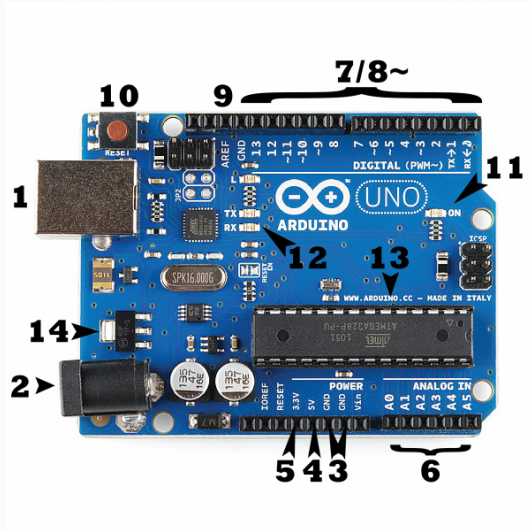
Arduino

Arduino Uno : caractéristiques (ATmega328P)

- Plateforme de **prototypage**
- Permet de reprogrammer le système
 - Alimentation : 7 à 12V ou via port USB (40 mA min.)
 - Horloge : 16 MHz
 - Mémoire SRAM : 2 kB
 - Mémoire EEPROM : 1 kB
 - Mémoire flash : 32 kB
 - Broches d'E/S : D0 à D13 (40 mA par broche)
 - Entrées analogiques : A0 à A5
 - Bus série, I2C et SPI
 - Gestion des interruptions
 - Dimensions : 74 x 53 x 15 mm

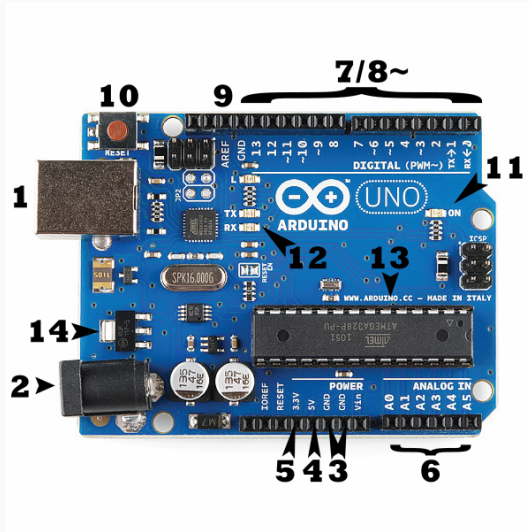


Arduino Uno : détails [réf.]



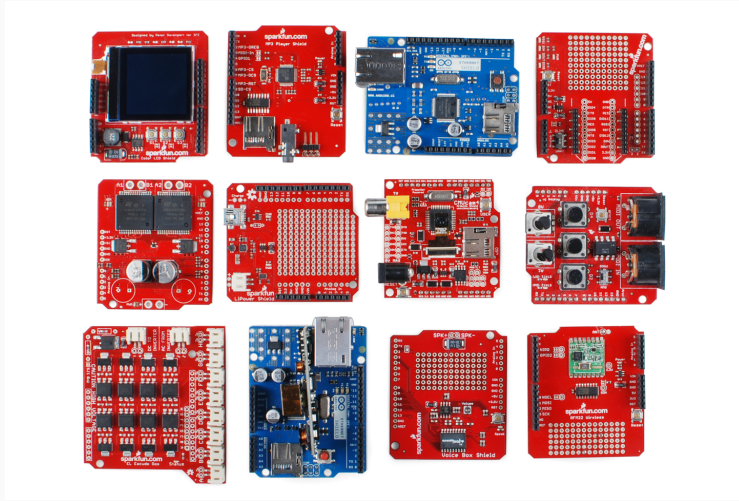
1. ...
2. ...
3. ...
4. ...
5. ...
6. ...
7. ...
8. ...
9. ...
10. ...
11. ...
12. ...
13. ...
14. ...

Arduino Uno : détails [réf.]



1. Connecteur USB
2. Connecteur *barrel jack*
3. Pins **GND** (masse) : référentiel 0V
4. Pin 5V
5. Pin 3.5V
6. Pins **analogiques** (Entrées) – A0 à A5
7. Pins **numériques** (E/S) – 0 à 13
8. Pins **numériques** (PWM) – 3, 5, 6, 9, 10, 11
9. Pin **AREF** : référentiel plafond en tension
10. Bouton *reset*
11. Témoin d'alimentation
12. Témoins Tx / Rx
13. Circuit intégré **ATmega328P**
14. Régulateur de tension (sécurité)

Arduino Uno : shields



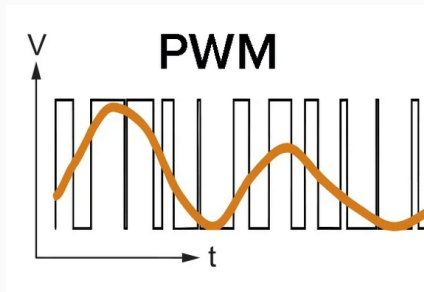
<https://www.shieldlist.org/>

Informatique embarquée

Programmation

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

- **GPIO** : *General-Purpose Input Output*
- Lecture analogique : tension sur un *pin*
 - Conversion analogique vers numérique (ADC)
 - $[0.0, 5.0[V \rightarrow \text{int}[0, 1024[$
 - Précision : 10 bits, résolution : 4.9 mV/valeur
- Lecture/Écriture numérique : états binaires
 - 0 (**LOW**, 0V)
 - 1 (**HIGH**, 3.3V ou 5V)
- Écriture analogique ?
 - Le microcontrôleur ne peut pas générer une tension arbitraire...



- Décrire un phénomène continu par des modulations
- Technique : *PWM*

Un premier programme Arduino

On utilise le simulateur :

- <https://wokwi.com/projects/new/arduino-uno>

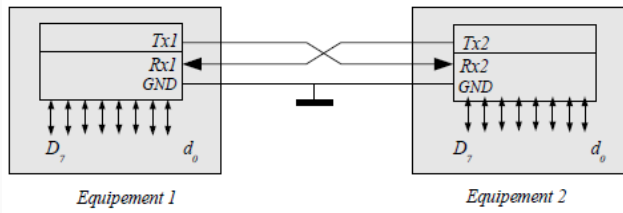
1. **TODO** Écriture numérique : allumer puis éteindre une LED sur le pin 5;
2. **TODO** Écriture analogique : faire varier l'intensité de la LED pour produire un effet de fondu, en boucle.

Informatique embarquée

Entrées/Sorties

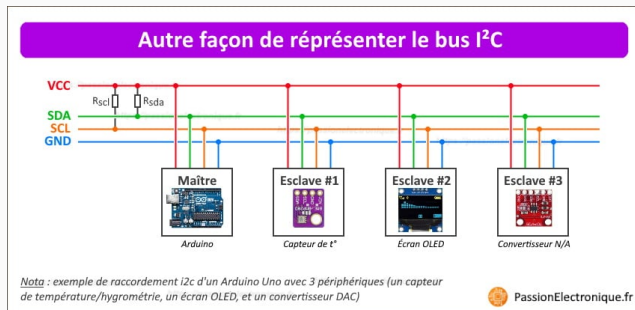
Communications série : UART [réf.]

- **UART** : *Universal Asynchronous Receiver Transmitter*
- Convertit des données *parallèles* (un octet en mémoire) en données *série* (des bits sur un fil)
- Liaison avec 3 fils : **Tx** (*transmit* de l'émetteur), **Rx** (*receive* du récepteur), **GND** (masse)
- *Full duplex* : émission et réception simultanées; point à point : 2 éléments
- Vitesse : en général 9600 bauds (bits/seconde), au maximum 115200 bauds (environ 14 kB/seconde...)
- "Masqué" derrière le port USB de l'Arduino!



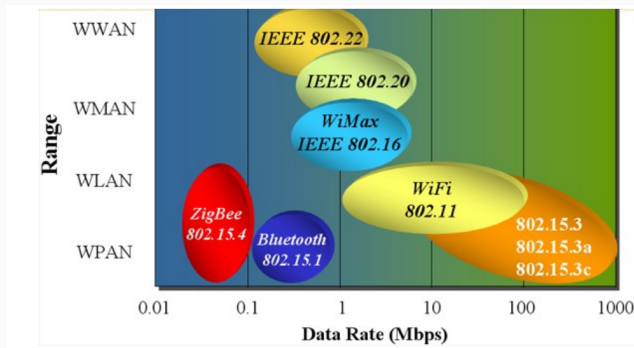
Communications série : I2C [réf.]

- **I2C** : *Inter-Integrated Circuit*
- Topologie "maître-esclave", adresses uniques, bits de lecture/écriture
- Liaison avec trois fils : **SDA** (signal de données), **SCL** (signal d'horloge), **GND** (masse)
- *Semi duplex* : communications bidirectionnelles, mais pas simultanées; multi-points
- Vitesse : en général 100 kbps (10x plus rapide que l'UART), au maximum 3.4 Mbps (environ 425 kB/seconde)



Communications sans-fil : Zigbee

- **Zigbee** : protocole / norme ouverte IEEE 802.15.4
- Topologie en maillage (*mesh*)
- Communications sans-fil courte distance (10 mètres @ 1 mW)
- Bande de fréquences non réservée : pas de licence
- Vitesse : 2.5 à 30 kB/seconde; très faible consommation d'énergie



Comparaison des protocoles Zigbee, Bluetooth et Wi-Fi

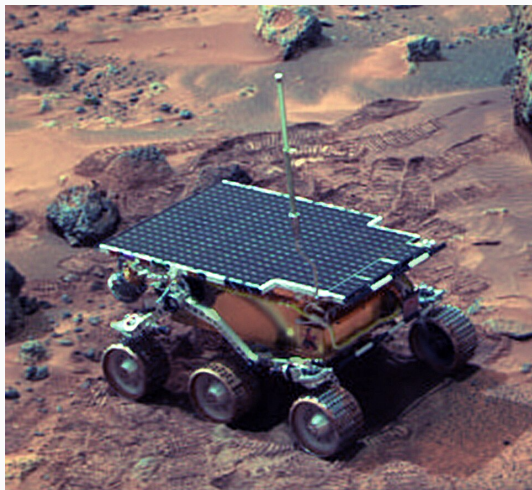
Caractéristique	Zigbee	Bluetooth Low Energy	Bluetooth	Wi-Fi	Z-Wave
IEEE	802.15.4		802.15.1 (BT 1.x)	802.11a/b/g/n/ac/ax/be	
Besoins mémoire	4-32 ko		250 ko +	1 Mo +	
Autonomie avec pile	Années	Années	Mois	Jours	
Nombre de nœuds	65 000+	illimité	255	256+	232
Vitesse de transfert	20-250 kb/s	1 Mb/s	1-3 Mb/s	11-54-108-320-1000 Mb/s	9,6 à 100 kbit/s
Portée (environ)	10 m	10 m	10 m	10 m (802.11a) à > 300 m (802.11b)	50 m

Robotique

Robotique

Généralités

Qu'est-ce qu'un robot ?



Sojourner sur la surface de Mars, décembre 1997 [réf.]

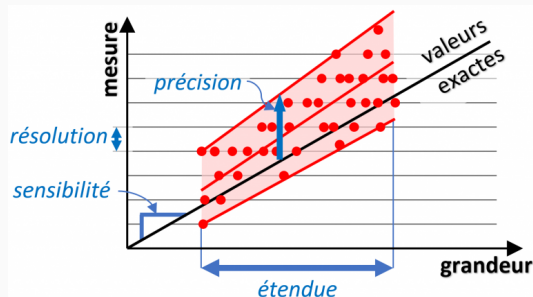
Qu'est-ce qu'un robot ?

Robot = système de systèmes

- **Mobilité** : énergie, déplacement (propulsion), orientation (direction)
- **Charge utile** : capteurs, informations, matériels, etc.
- **Intelligence** : contrôle, commande, communication, décision

Caractéristiques d'un capteur :

- Temps de réponse
- Fréquence d'échantillonnage
- Précision, finesse, résolution, stabilité
- Coût
- Taux d'erreur (dispersion, aberrations, etc.)
- Robustesse (environnement)
- Charge de calcul induite
- Poids, consommation d'énergie, dimensions, etc.



- Éléments contrôlés en boucle fermée : **asservissements**
- Système automatique qui ajuste son comportement en fonction de **consignes** et de **mesures**
- Exemple :



Boucle d'asservissement

1.  **Consigne** : objectif à atteindre (ex : avancer à 10 cm du mur)
2.  **Mesure** : capteur qui lit la position (GPS), la distance réelle (LiDAR), etc.
3.  **Erreur** : différence entre consigne et mesure
4.  **Contrôleur** : calcule la **commande** à partir de l'erreur
5.  **Commande** : signal envoyé au système (ex : augmenter la vitesse des moteurs)

Robotique

Commande à distance

- Architecture :
 - PC : Envoi de commandes filaire → USB → Robot Arduino
 - PC : Envoi de commandes sans fil → XBee → Robot Arduino
- Peu importe le type de liaison, le même programme s'exécute sur l'Arduino
- → Il faut programmer l'Arduino pour **recevoir**, **décoder** et **exécuter** des lignes de commande.

Sur nos robots...

- De quels capteurs disposons-nous?
- De quels actionneurs disposons-nous?

Communications

- Objectifs : avancer, reculer, tourner...
- Quels éléments va-t-on commander ?
- Sur la base de quelles informations ?
- → De quelles **commandes** avons-nous besoin ?

- Une ligne de commande peut contenir :
 - Une commande élémentaire : `[cmd]`
 - Une séquence de commandes élémentaires : `[[cmd_1][cmd_2]]`
- Une commande peut prendre un paramètre : `[cmd param]`
 - Séparation par un espace

Exemple de ligne de commande

```
[[mda 255][mga 255]]
```

Robotique

Interface utilisateur

- Langage : JavaScript
- Programmation asynchrone, ou événementielle

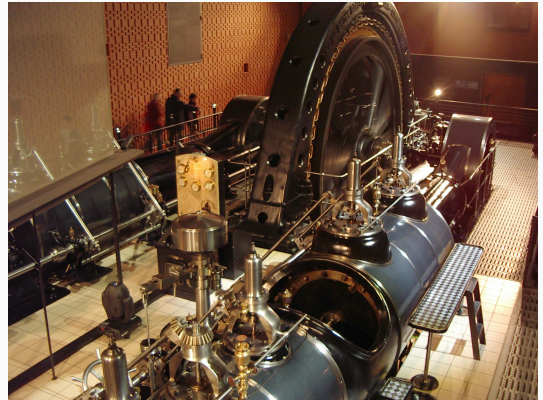
```
// Charger une bibliothèque  
const { Something } = require('something');  
  
// Instancier un objet  
const thing = new Something({ "some": "value" });  
  
// Écouter un événement  
thing.on('someEvent', (data) => {  
    // Faire quelque chose...  
    console.log(data);  
});
```

Robotique

Comportement autonome

Comportement autonome – Origines historiques

- Sans régulation : la machine peut **accélérer dangereusement** ou **ralentir**
- Problèmes de sécurité et de fiabilité
- → Comment stabiliser la vitesse des machines à vapeur (moulins, locomotives, etc.)?



Machine à vapeur Sulzer et alternateur Brown Boveri
(Electropolis, Mulhouse)

Comportement autonome – Théorie du contrôle



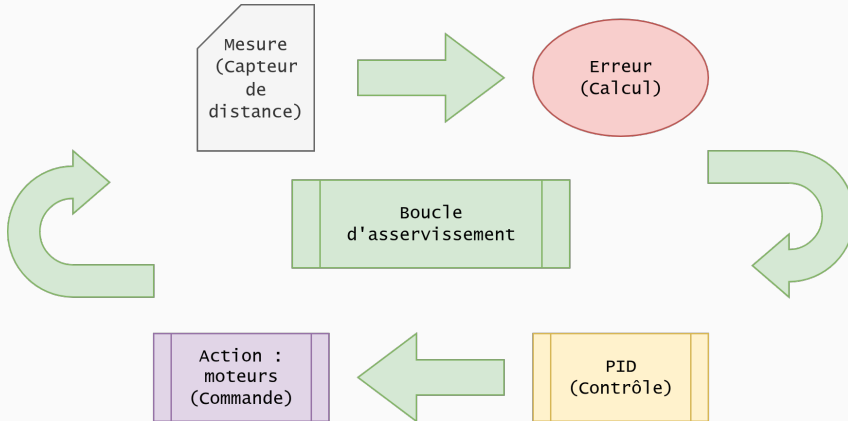
- Comment stabiliser la vitesse des machines à vapeur (moulins, locomotives, etc.)?
- **Régulateur à boules** pour machines à vapeur (James Watt  (XX^{ème})
- Plus la machine accélère, plus les boules s'écartent, réduisant l'arrivée de vapeur → diminution de la vitesse
- Initialement proposé par Christian Huygens  (XVII^{ème})
- Principe purement mécanique
- Limites :
 - Réponse lente
 - Pas de correction d'erreur persistante
 - Pas d'anticipation des variations rapides
 - Pas adapté aux systèmes numériques!

- **PID** : mis en équation en 1922 par Nicolas Minorsky 🇷🇺
- Testé sur un cuirassé de l'US Navy, l'USS New Mexico
- En pilotage automatique, en partant d'une erreur de 2 degrés, le navire a pu faire mieux qu'un barreur humain en maintenant son cap avec une précision d' $\frac{1}{3}$ de degré



Contrôle – Fonctionnement du PID

Feedback loop : stabiliser automatiquement des systèmes en s'appuyant sur les comportements passés



Équation du PID en version discrète (pour une implantation numérique) :

$$u[n] = K_p \cdot e[n] + K_i \cdot \sum_{i=0}^n e[i] + K_d \cdot (e[n] - e[n-1]) \quad (1)$$

Équation du PID en version discrète (pour une implantation numérique) :

$$u[n] = K_p \cdot e[n] + K_i \cdot \sum_{i=0}^n e[i] + K_d \cdot (e[n] - e[n-1]) \quad (2)$$

- Composante p : correction proportionnelle à l'erreur
 -  Problème : erreur persistante → délai de correction très long
- Composante i : correction proportionnelle à la somme des erreurs précédentes
 -  Problème : instabilité de la trajectoire → oscillations
- Composante d : correction prédictive basée sur la variation de l'erreur
 -  L'erreur augmente → on "accélère" la correction
 -  L'erreur diminue → on "freine" la correction

Sans PID

Réactions brutales

Oscillations

Dépassement

Avec PID

Réactions fluides

Stabilité

Précision

Robotique

Calcul déporté

- Il est possible de réaliser une partie des calculs sur le PC qui commande le robot
- Mais...

Avantages, inconvénients?

- Quels sont ici les **compromis** à réaliser?
- Comment appelle-t-on un tel système?

- Compromis techniques :
 - **Temps de calcul** : PID simple adapté à l'Arduino Uno ?
 - **Mémoire** : quel historique des erreurs peut-on stocker ?
 - **Énergie** : gains avec le PID ?
 - **Latence** : calcul local = réponse rapide

Bibliographie

- [1] Oliver Inderwildi et al. « The Impact of Intelligent Cyber-Physical Systems on the Decarbonization of Energy ». *Energy & Environmental Science* 13.3 (2020), p. 744-771. ISSN : 1754-5692, 1754-5706. DOI : **10.1039/C9EE01919G**. (Visité le 19/10/2025).
- [2] J.O. Kephart et D.M. Chess. « The Vision of Autonomic Computing ». *Computer* 36.1 (jan. 2003), p. 41-50. ISSN : 0018-9162. DOI : **10.1109/MC.2003.1160055**. (Visité le 05/12/2024).
- [3] Edward A. Lee. « Cyber Physical Systems : Design Challenges ». *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. Orlando, FL, USA : IEEE, mai 2008, p. 363-369. ISBN : 978-0-7695-3132-8. DOI : **10.1109/ISORC.2008.25**. (Visité le 17/09/2025).