



**Licence 1 portail
Informatique
Electronique**

UE DIE – Découverte Informatique Electronique

Samuel CRAND
Bât 6 – ISTIC
Tél : 02 23 23 65 04
Samuel.Crand@univ-rennes1.fr

Equipe pédagogique

Sophie.Allain@univ-rennes1.fr

Christian.Brousseau@univ-rennes1.fr

Samuel.Crand@univ-rennes1.fr

Clement.Ferise@univ-rennes1.fr

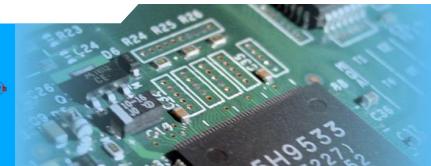
Souhir.Gabsi@univ-rennes1.fr

Patrick.Lamy@univ-rennes1.fr

Jordane.Lorandel@univ-rennes1.fr

Christophe.Moy@univ-rennes1.fr

Anne-Claude.Tarot@univ-rennes1.fr





Campus Scientifique de Beaulieu

BEAULIEU NORD

- 21 Médecine préventive Universitaire (SIMPS) - Handicap - CLAEH
- 22 Recherche Physique
- 23 Institut Mathématique de Rennes
- 24 Recherche Chimie
- 25 Recherche Biologie
- 26 Enseignement - Archéologie
- 27 Salle de cours et de T.D.
- 28 Langues (SCELW)
- 29 Salle d'informations, salles de cours; de T.D., CCNE, antenne SMU
- 30 Salle d'informations et T.D.
- 31 Amphithéâtre G. H. LILL et V. Bach
- 32 TP Sciences naturelles
- 33 TP Physique
- 34 TP Chimie
- 35 TP Génétique
- 36 Salle de cours et de T.D.
- 37 TP Biologie
- 38 TP Géosciences
- 39 Salles de cours et T.D.
- 40 TP Physique
- 41 TP Chimie
- 42 TP Biologie
- 43 SMU et Bureau des Sections syndicales et des Associations du Personnel
- 44 Salle de biostatistique
- 45 S.C.D. (section scientifique)
- 46 Salles de cours et T.D.
- 47 Amphithéâtre A. et N.
- 48 Bureau des Associations étudiantes, Cafétéria

BEAULIEU SUD

- 1 Administrateur
- 2 Enseignement, salles de cours et de T.D.
- 3 Amphithéâtre A, B, C, D, E et L, Amphi
- 4 T.P. Informatique
- 5 T.P. Biologie
- 6 T.P. Biologie
- 7 T.P. Géosciences
- 8 Salles de cours et T.D.
- 9 T.P. Physique
- 10 T.P. Chimie
- 11 U.T.E. (Département G.E.A. et C.S.)
- 12 SMU et Bureau des Sections syndicales et des Associations du Personnel
- 13 Salle de biostatistique
- 14 S.C.D. (section scientifique)
- 15 Salles de cours et T.D.
- 16 Amphithéâtre A, M et N
- 17 Bureau des Associations étudiantes, Cafétéria

**Nous sommes là...
Bât 2A**

**Bâtiment 6
Salle de TP**

BEAULIEU SUD

- 9 U.T.I. (Espace des Technologies Innovantes)
- 10A Recherche Chimie
- 10B Recherche Chimie - Recherche Micrométrie
- 10C Recherche Chimie
- 11A Recherche Physique - CRMPO
- 11B Recherche Physique - CCMO
- 11C Recherche Physique et Electronique IETR
- 12A IFASIC - RISA - CII - INRIA (Informatique)
- 12B Accès IFASIC et CII au 12D
- 12C Accès RISA au 12C
- 13A Recherche Biologie - Zoologie
- 13B Secrétariat UFR S.N.E.
- 14A Recherche Biologie - Environnement
- 14B Centre Américain de Recherche en Environnement - CAREN
- 15A Recherche Géosciences
- 16A Atelier de Physique - INRA
- 17A Locaux Produits Chimiques
- 18A Messagerie - Dépôt Colis - Magasin STI
- 19A Messagerie - Dépôt Colis - Magasin STI
- 20A Service Technique Immobilier
- 20B

UNIVERSITÉ DE RENNES 1

September 2007

Bâtiment administratif (hall 1) :

- Accès
- Vice-Présidence
- Salle des Thèses
- Service Intérieur de Beaulieu
- Service Secrétariat
- Secrétariat UFR GRAN
- Archives Muséologiques
- Service Imprimerie
- Magasin Papier

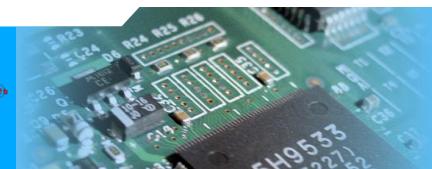
Lieu des TP

4



Plan

- 1. Les objectifs du module**
- 2. Le contexte**
- 3. Un système électronique**
- 4. La carte capteurs**
- 5. Le microcontrôleur**
- 6. Les fonctions électroniques**
- 7. L'environnement logiciel Arduino**
- 8. Les capteurs**



1. Objectifs du module

Objectifs globaux module DIE

- . Comprendre les applications de **l'informatique et de l'électronique par la pratique**
- . Découvrir l'ensemble des notions sous jacentes à la mise en œuvre **d'applications mobiles** et de l'usage de **capteurs**.

Objectifs de la partie électronique du module DIE

- . Apprendre à effectuer des mesures électriques
- . Lire et interpréter des schémas fonctionnels et électriques
- . Étudier des fonctions électroniques primordiales par la pratique
- . Utiliser des capteurs de la vie courante
- . Réaliser une application globale permettant d'interagir avec le monde extérieur en relevant des informations sur l'environnement pour mener des actions



1. Objectifs du module

Organisation

- . 2 cours de 1,5h en amphi
- . 2 journées de TP de 7h : 8h30 – 12h et 14h – 17h30

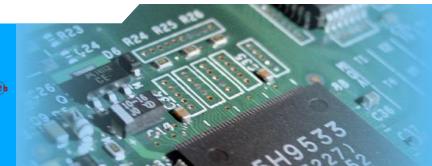
- . 2 cours : ½ promotion du portail Informatique Electronique
 - . A l'issue de chaque cours => Dépôt du diaporama sur MOODLE
 - . Cours => préparation des TP
 - . Interaction possible et souhaitée



TP : Préparation individuelle avant les séances

Quelques questions à l'issue de chaque séance de TP

Examen après les 2 séances de TP d'électronique



1. Objectifs du module

Organisation des TP

- . TP par binôme → Un binôme = 2 personnes qui travaillent
- . Le polycopié de TP : disponible à la 1ère séance de TP
- . Diapositives du cours, vidéos, poly de TP et codes disponibles sur Moodle
- . Quelques questions à répondre individuellement à l'issue de chaque journée
- . Ce document est corrigé/commenté par un enseignant pour la journée

suivante

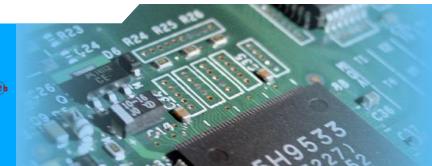
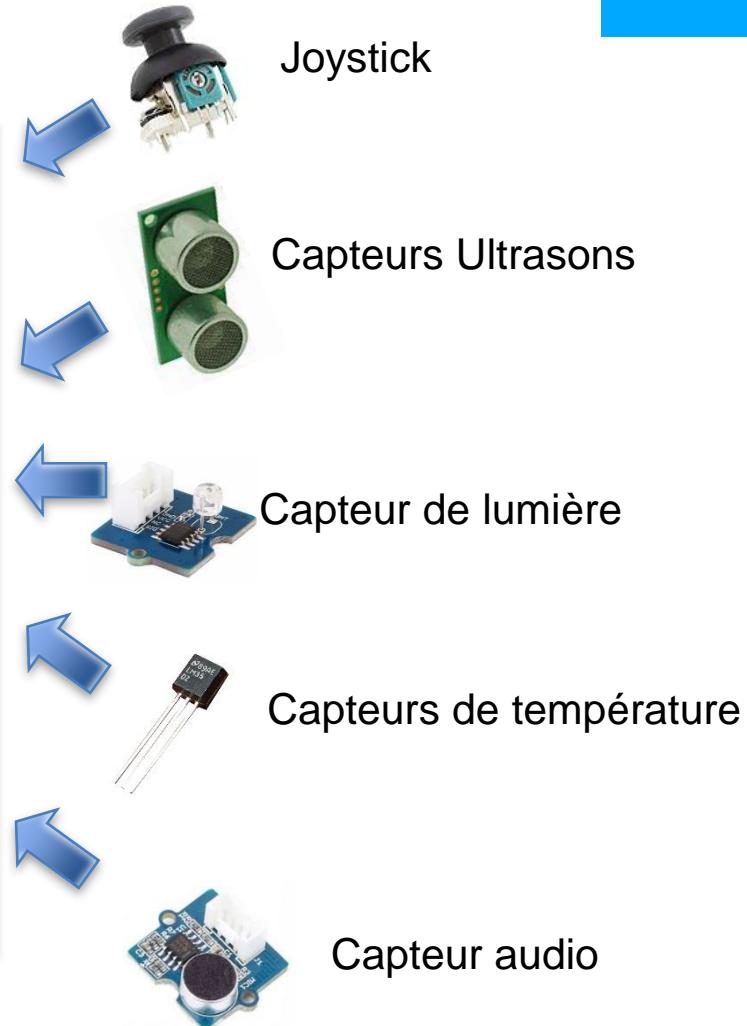
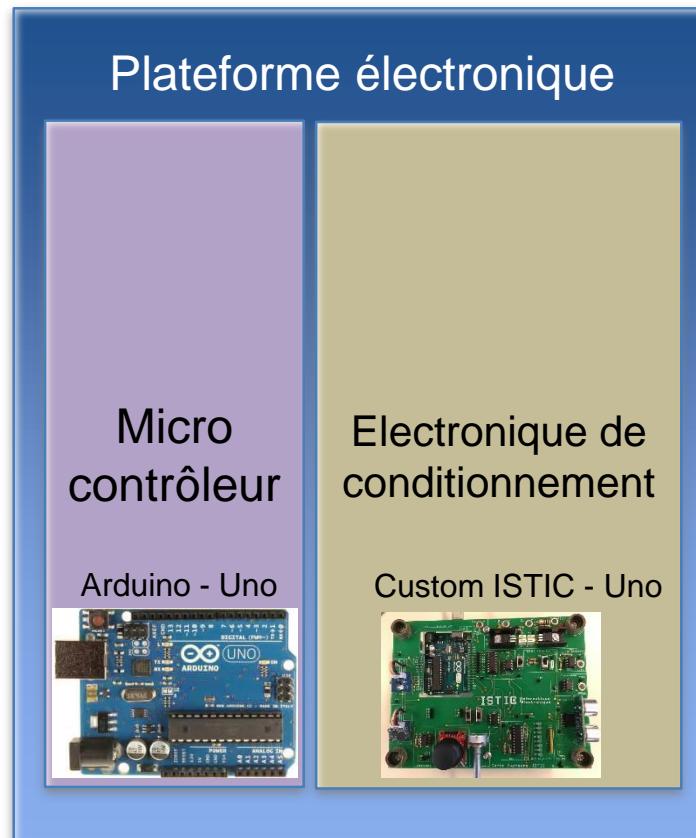
- . Les TP pourront être « agrémentés » de moments de type TD en petits groupes



1. Les objectifs du module



Exemple d'infos transmises :
*joystick,x,514
joystick,y,123
joystick,bp,0
temperature,19
ultrason,25
son,72
lumiere,2*



1. Les objectifs du module

Plateforme électronique

Micro contrôleur

Arduino - Uno



Electronique de conditionnement

Custom ISTIC - Uno

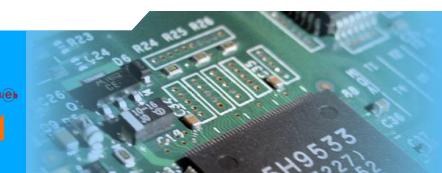


Electronique de conditionnement

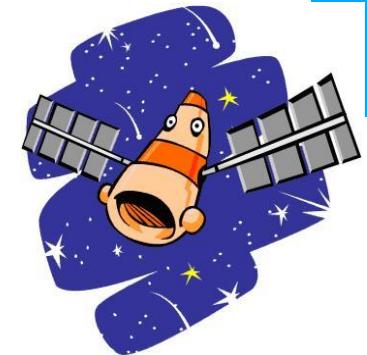
- Interrogation des capteurs
- Mise en forme des signaux issus des capteurs
 - . Filtrage
 - . Amplification
 - . Conversion analogique numérique
- Visualisation des signaux
 - . Conversion numérique-analogique
 - . Leds

Microcontrôleur

- Communication avec l'extérieur (PC)
- Interrogation des capteurs
- Mise en forme des signaux issus des capteurs
- Stockage/analyse



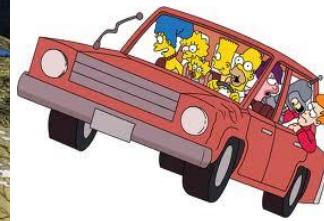
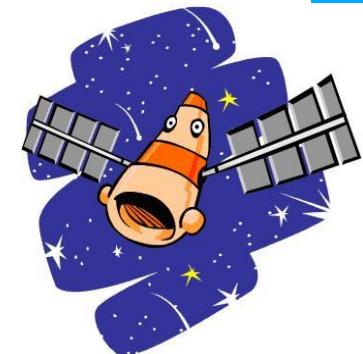
2. Le contexte



Mais où est l'électronique ?



2. Le contexte



2. Le contexte

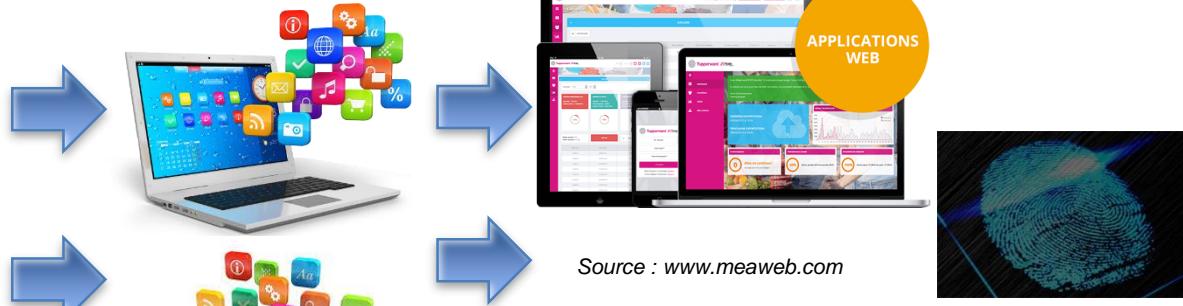
Informatique ?

- . Automatiser l'exécution de tâches et applications
- . Mise en œuvre de logiciels (Software)
- . Information traitée par l'exécution de programmes (codes) sur des machines

Données => fichiers...



Source : studiohookipa.com



Source : www.meaweb.com



Source : www.secure-ic.com

Des codes ... des langages...



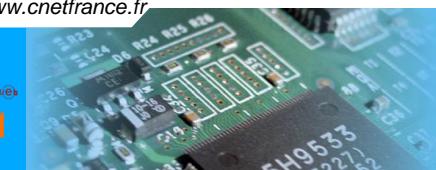
Source : www.developpez.net



Source : www.eficom.net



Source : www.cnetfrance.fr



2. Le contexte

Electronique ?

- . Traiter et gérer des signaux électriques
- . Mise en œuvre de composants matériels (Hardware)
- . Signaux (tension/courant) traités par des circuits constitués de composants

Signaux => composants



Source : www.lextronic.com



Source : www.microprocessor-fr.com



Source : fr.audiofanzine.com



Source : www.adnauto.fr



Source : wikipedia



Source : AFP



2. Le contexte

Mais à quoi sert l'électronique ?

Faciliter l'interaction avec l'environnement



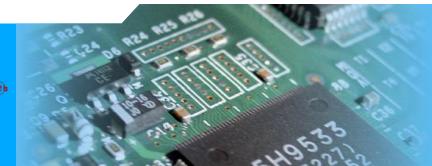
- . Collecter des informations
- . Recevoir des informations
- . Traiter des informations
- . Mémoriser des informations
- . Interpréter des informations
- . Exploiter des informations
- . Agir en fonction des informations
- . Transmettre des informations



Informations ?

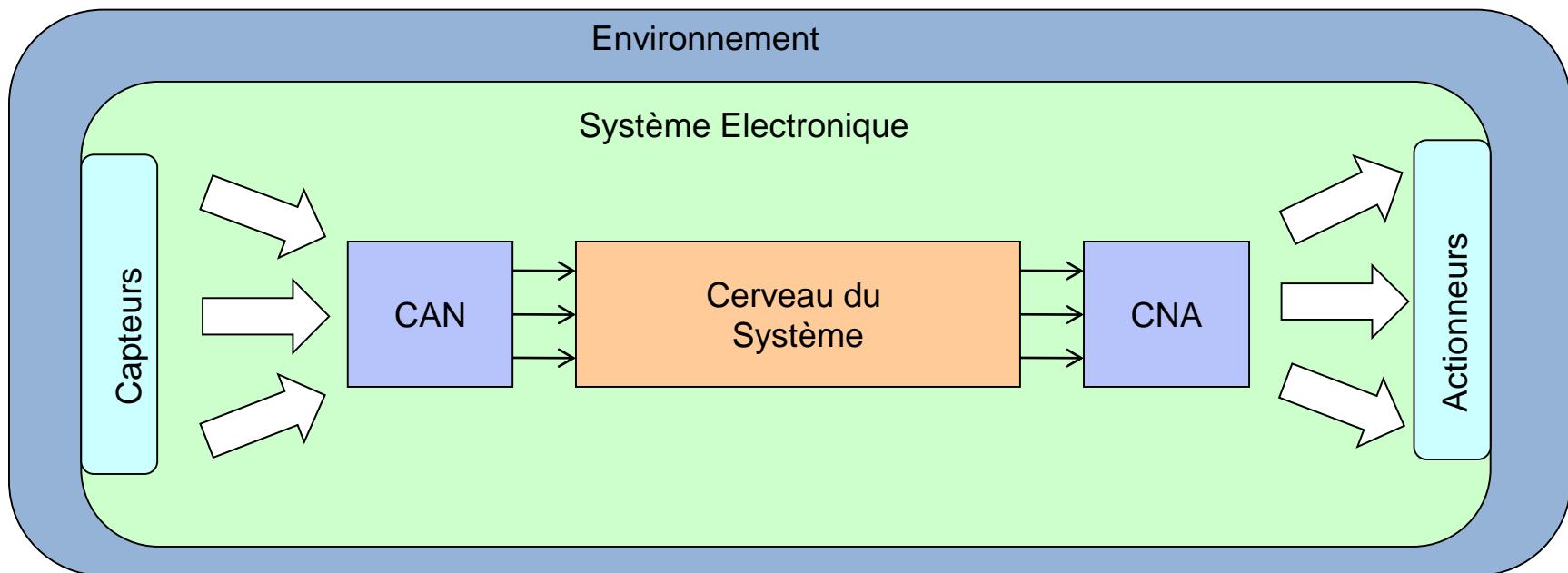
Signaux électriques

Tension et courant



3. Un système électronique

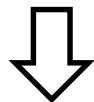
Définition : un système électronique permet d'interagir avec son environnement



3. Un système électronique

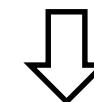
Capteurs

Données environnementales



Température
Mouvement : déplacement,
vitesse, accélération
Gaz
Lumière
Son
Humidité
...

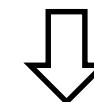
Signaux électriques



Tension
Courant



Analogique



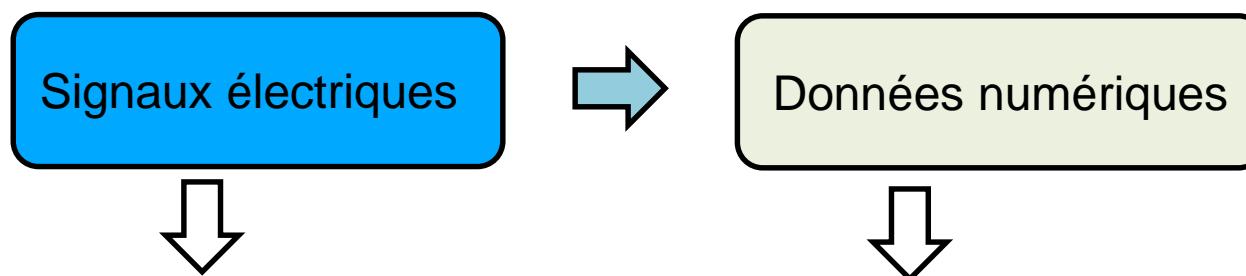
Infinité de valeurs
(espace et temps)



3. Un système électronique

CAN

Convertisseur Analogique Numérique



Tension
Courant

Signaux binaires : 2 valeurs possibles => 0 ou 1
Association de signaux binaires : octet
1 octet = 8 bits => 256 valeurs possibles (2^8)

Pourquoi traduire les signaux électriques en données numériques ?



Faciliter leurs manipulations



3. Un système électronique

CAN

Convertisseur Analogique Numérique
(ADC) Analog Digital Converter

Comment faciliter leurs manipulations ?



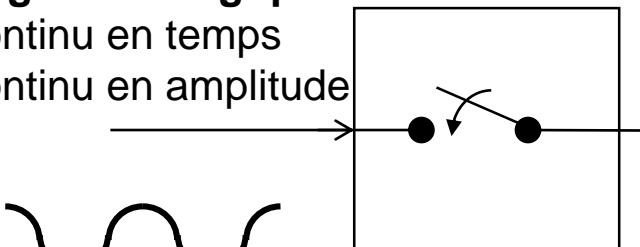
Réduction du nombre de données
échantillonnage
Réduction de leur représentation
quantification



Echantillonnage :
discrétisation temporelle
Quantification :
discrétisation d'amplitude

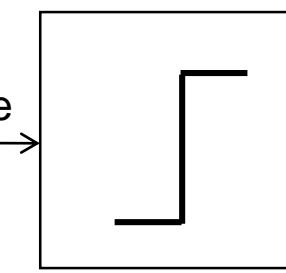
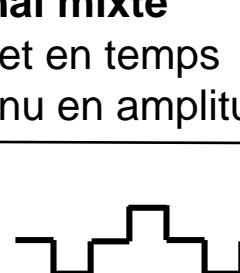
Signal analogique

- continu en temps
- continu en amplitude



Signal mixte

- discret en temps
- continu en amplitude

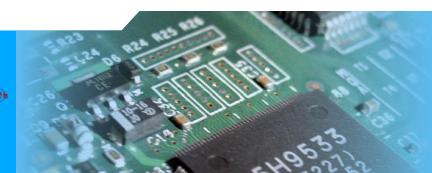


Signal numérique

- discret en temps
- discret en amplitude

010 001 001
010 111 111

Quantificateur

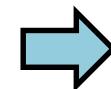


3. Un système électronique

CNA

Convertisseur Numérique Analogique
(DAC) Digital Analog Converter

Données numériques



Signaux électriques

Signaux binaires : 2 valeurs possibles => 0 ou 1

CNA 8 bits : 256 données analogiques différentes en sortie

Sortie : tension ou courant

Tension
Courant

Pourquoi traduire les données numériques en signaux électriques ?



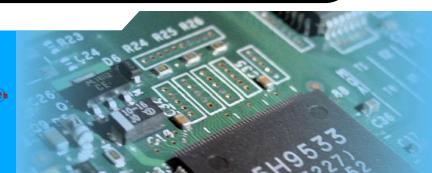
Agir sur l'environnement



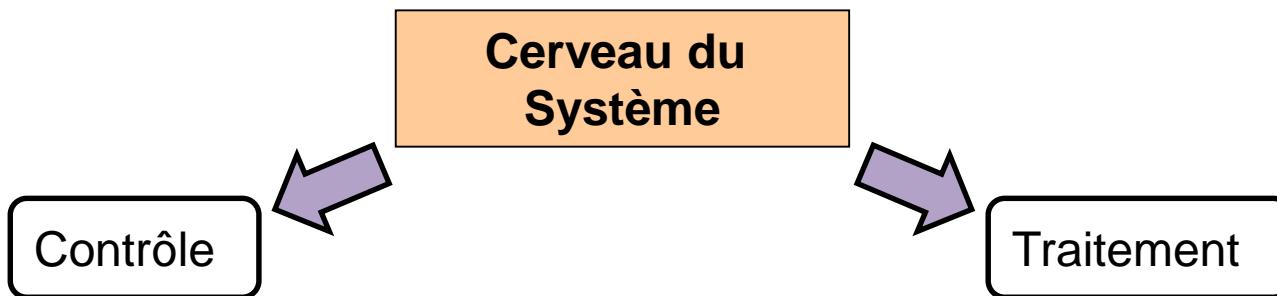
analogique



Infinité de valeurs
(espace et temps)

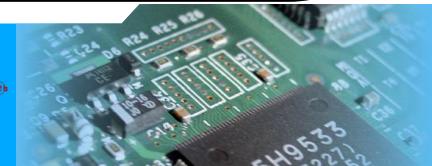


3. Un système électronique

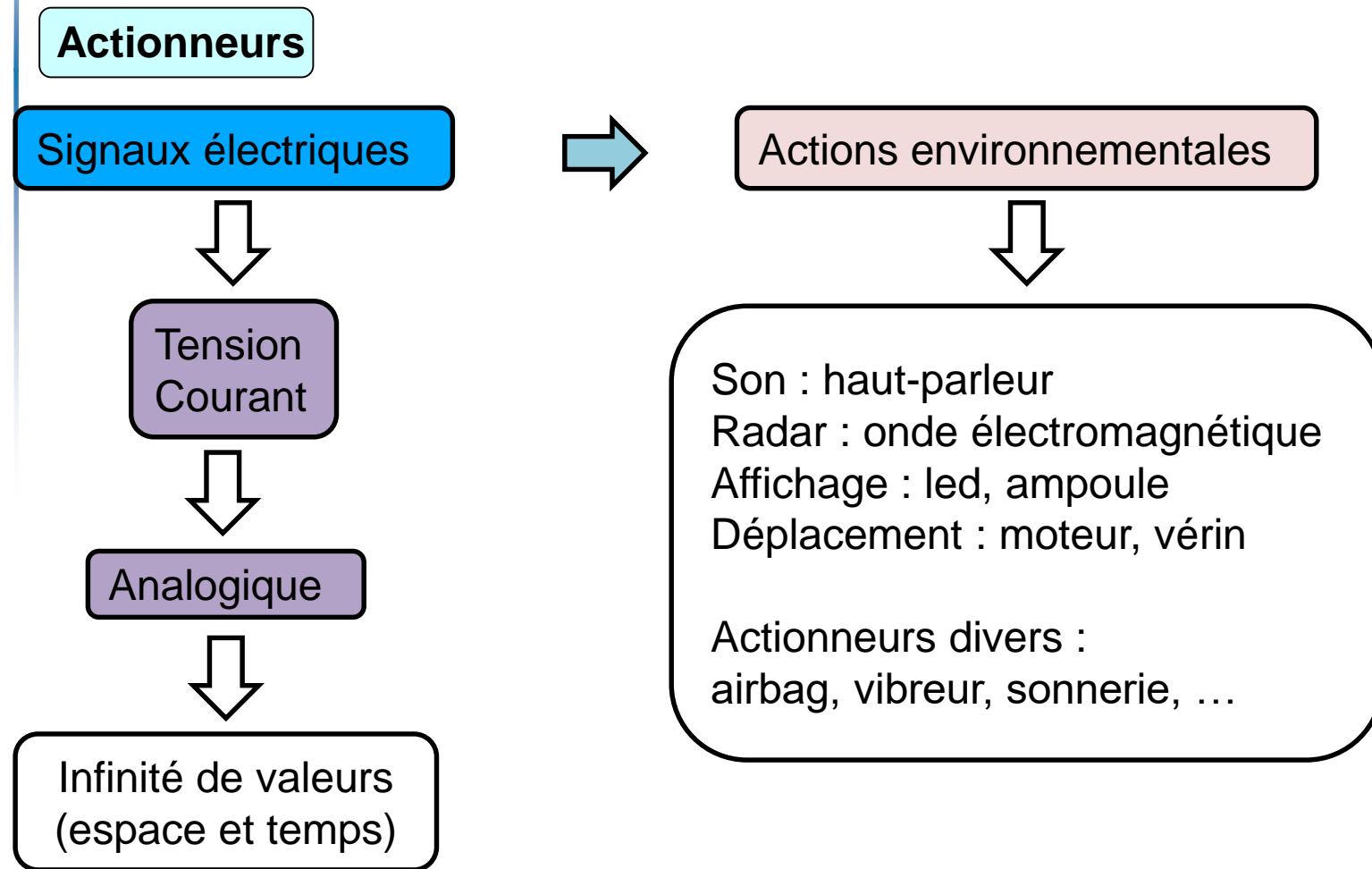


- . Piloter les CAN et CNA
 - Quand ?
 - Comment ?
 - Quoi ?
- . Recevoir les informations
- . Mettre en forme les informations pour faciliter leurs traitements
 - Ex : Conversion série/parallèle
- . Mettre en forme les informations pour faciliter leurs transmissions

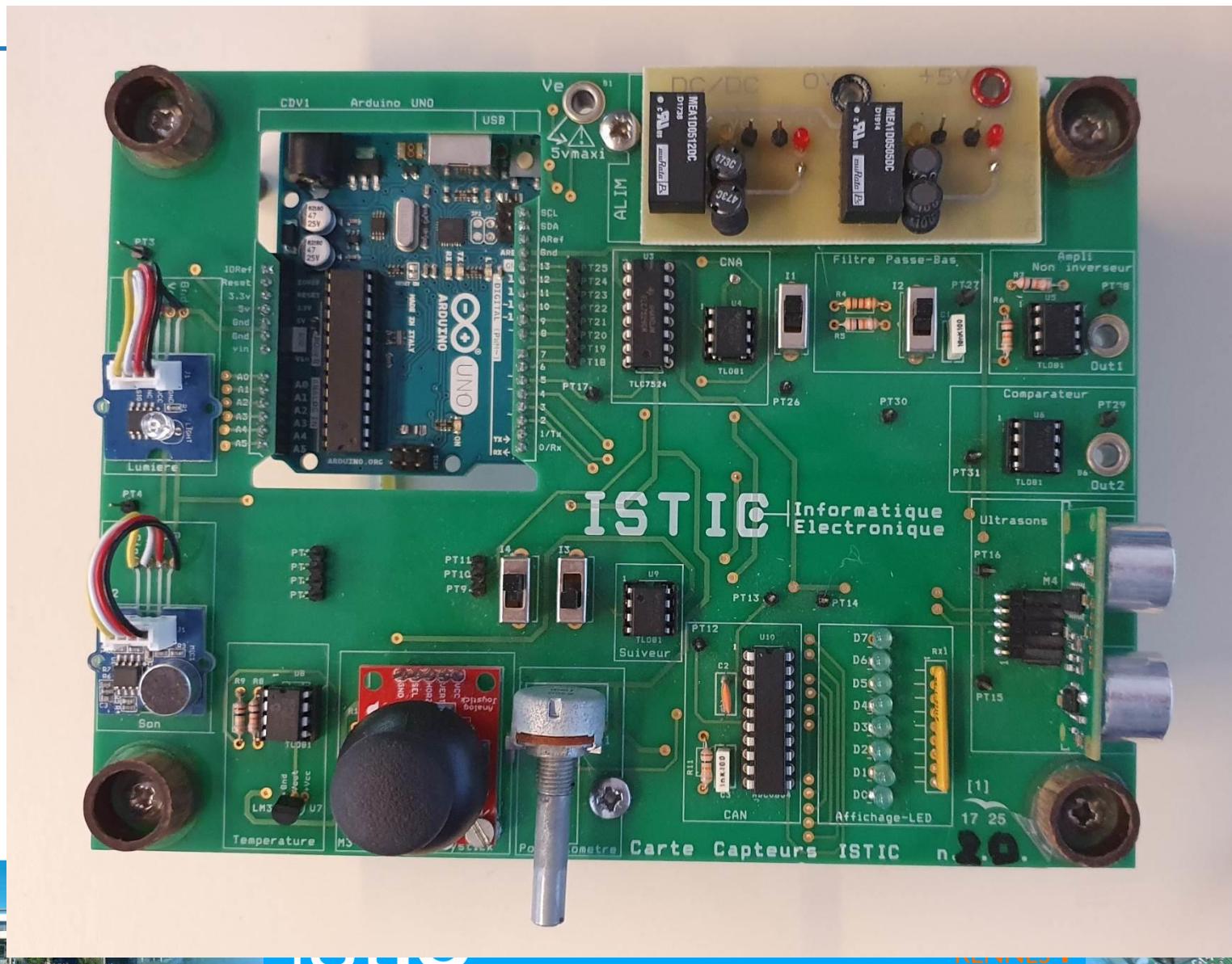
- . Collecter les informations
- . Modifier les informations
- . Mémoriser les informations
- . Interpréter les informations
- . Exploiter les informations
- . Agir en fonction des informations
- . Transmettre les informations



3. Un système électronique

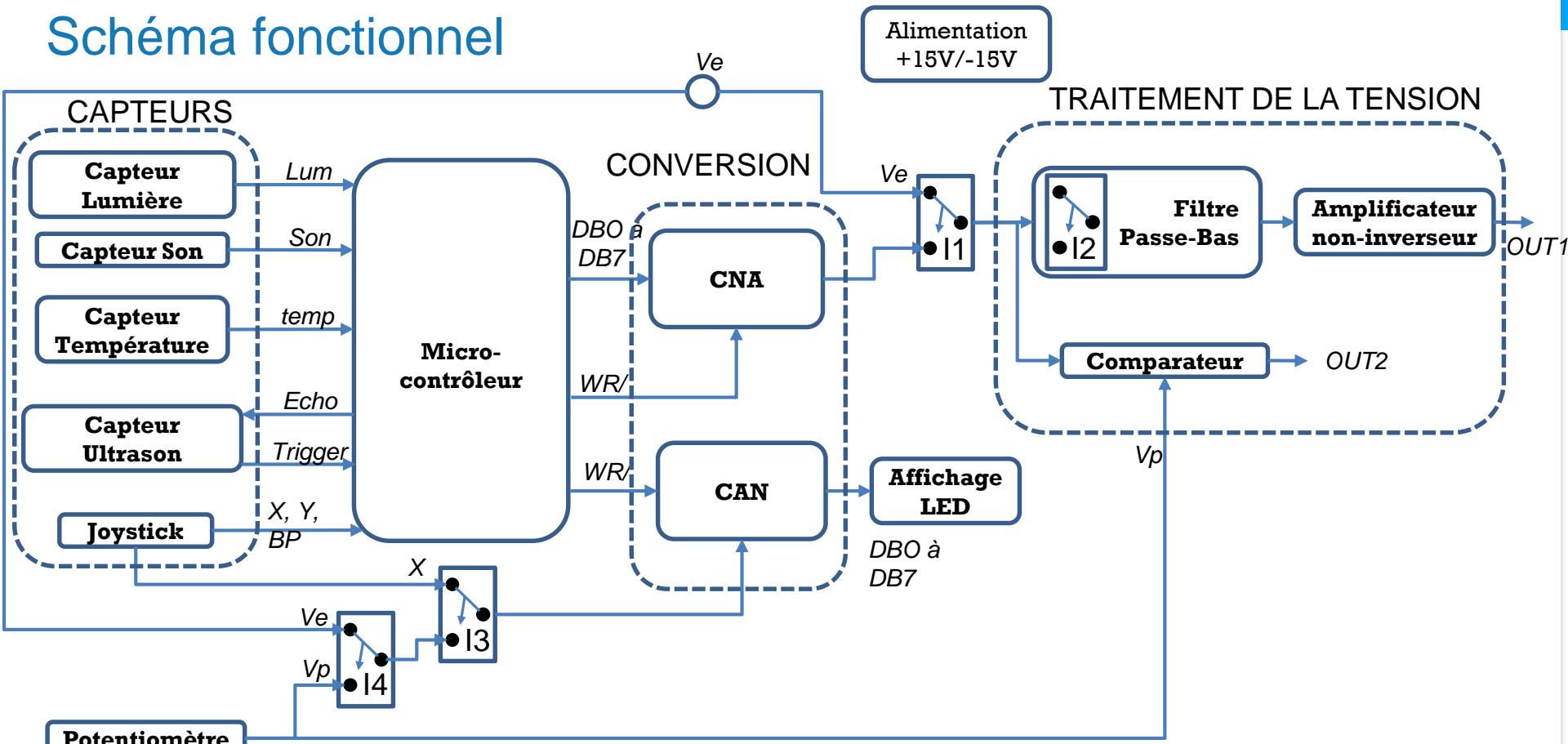


4. La carte capteurs



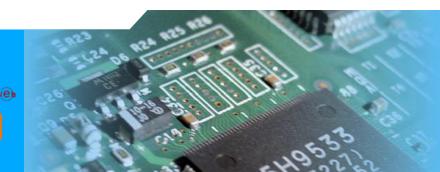
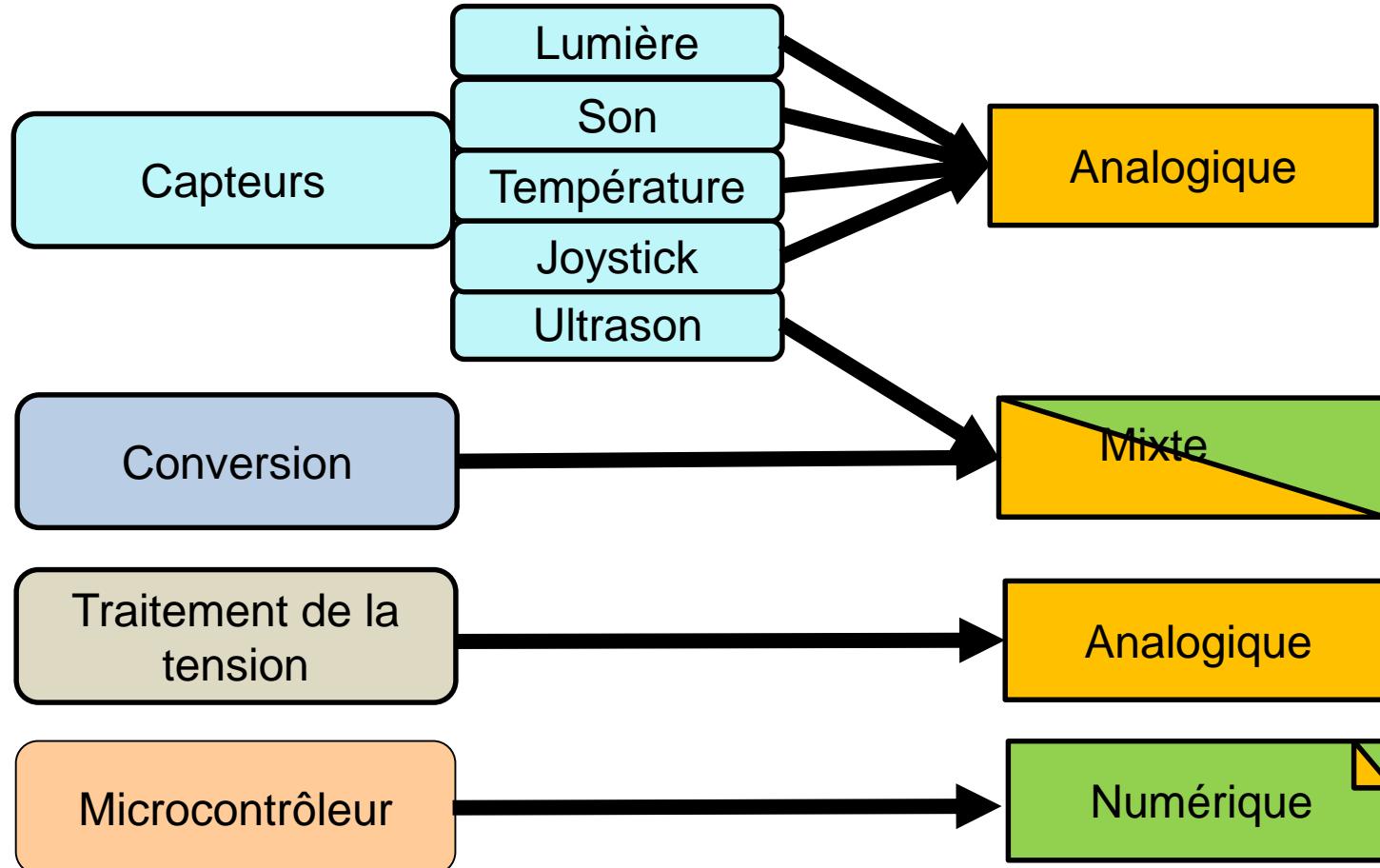
4. La carte capteurs

Schéma fonctionnel

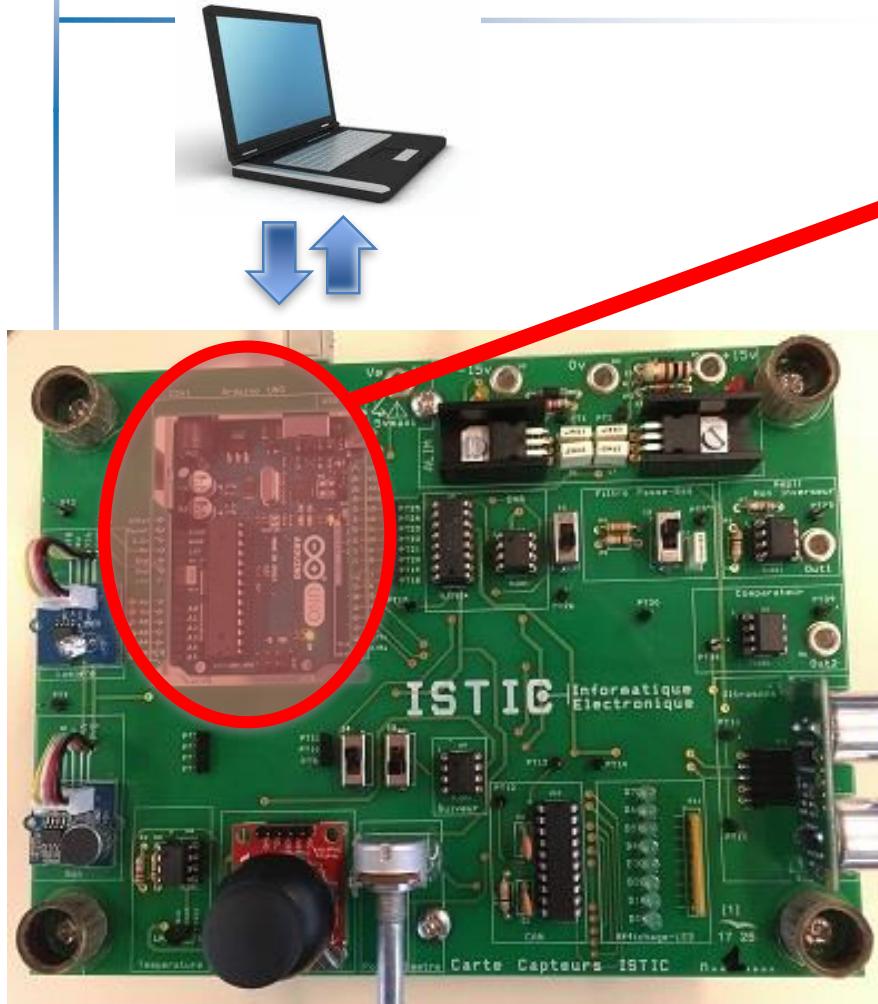


Source : sophie.allain@univ-rennes1.fr

4. La carte capteurs

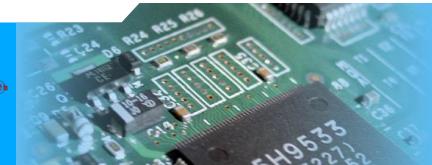


5. Le microcontrôleur



- Collecter les données des capteurs
- Stocker les données des capteurs
- Interpréter les données des capteurs

-
- Assurer la communication entre l'extérieur (PC par exemple) et la carte capteurs
 - Transmettre les données des capteurs

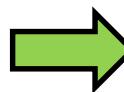


5. Le microcontrôleur

Un microcontrôleur = 1 Ordinateur « comme à la maison ou à l'université » ?

OUI

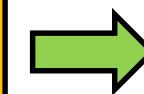
Unité Centrale (Processeur)
Mémoire vive
Mémoire morte
Interfaces pour communiquer



Ca se programme donc ?... OUI

NON

Ressources très limitées

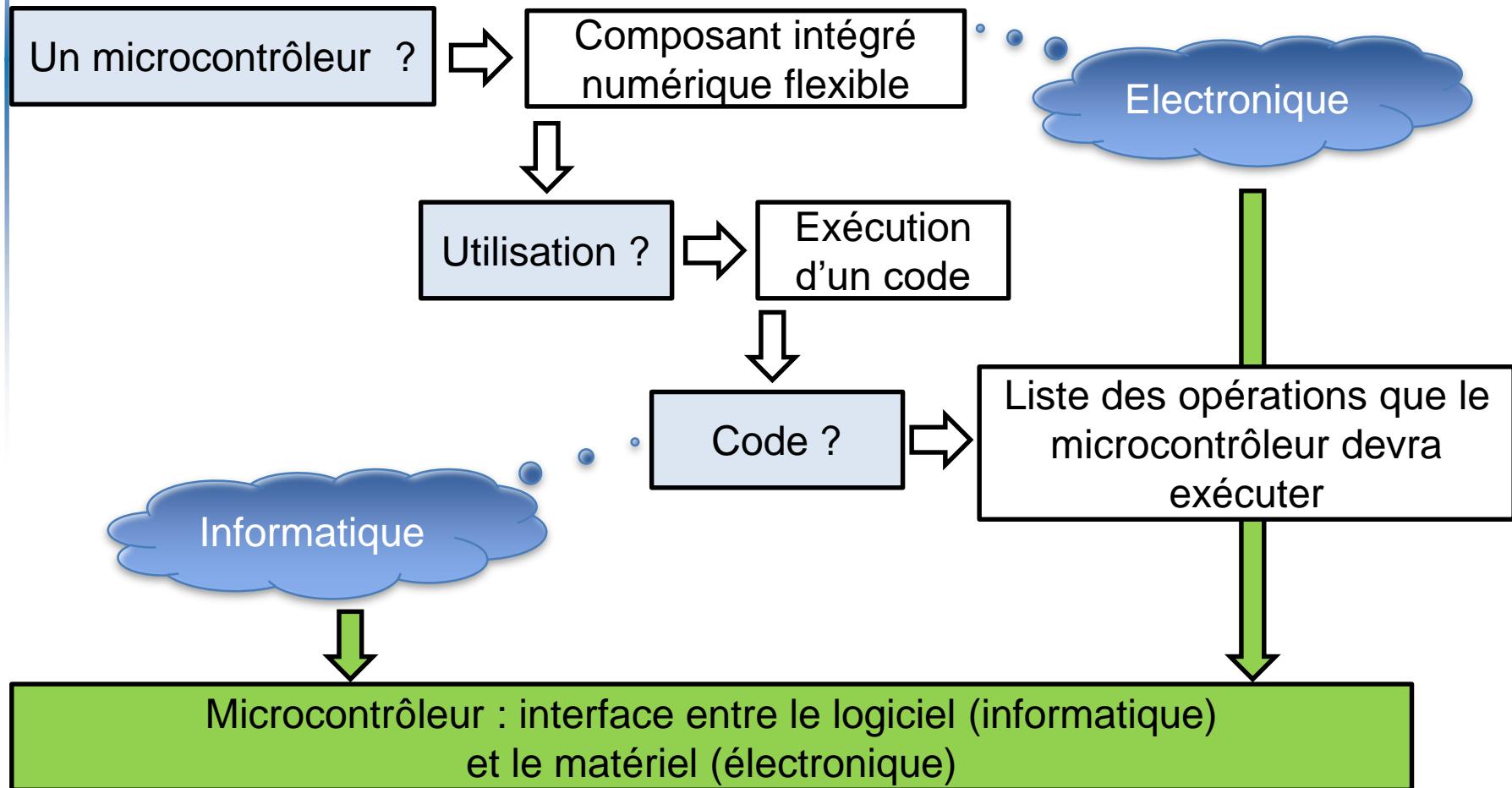


C'est peu utilisé ?... NON

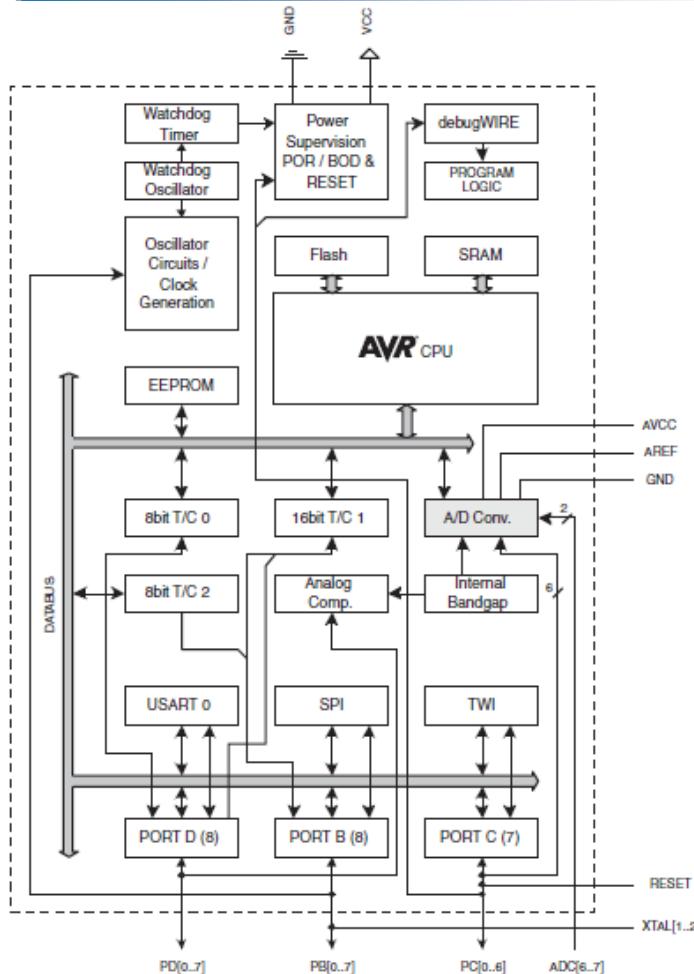
Il y en a « presque » partout... dans une voiture : plus d'une centaine de microcontrôleurs



5. Le microcontrôleur



5. Le microcontrôleur



ATmega328 et non... Arduino



- Architecture RISC (Reduced instruction set computer)
- Low Power
- In System programmable system
- 20 MIPS (Mega Instructions Per Second)
- 3 types de mémoire (Flash, SRAM, EEPROM)
- 6 A/D Converters de 10 bits
- 6 PWM (*Pulse Width Modulation*)

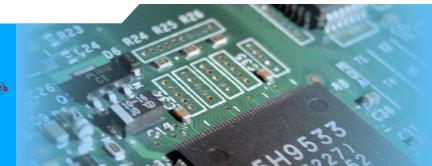
6. Les fonctions électroniques

Méthode :

Présentation rapide en cours

Expérimentations en TP « à votre rythme »

1. Conversion tension/courant
2. Conversion tension/lumière
3. Filtrage
4. Amplification
5. Comparaison
6. Conversion analogique numérique
7. Conversion numérique analogique



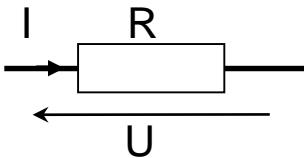
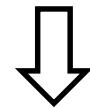
6.1. Fonction 1 : conversion courant/tension

Conversion courant/tension et tensions/courant : Fonction de base de l'électronique

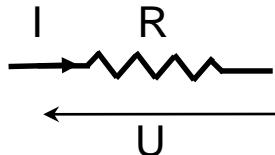
- => Courant et tension : 2 grandeurs à manipuler qui transportent l'information utile
- => Informatique : monde numérique

Objectif du convertisseur : Convertir un courant en tension et vice versa

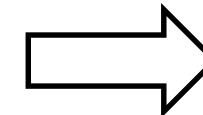
Convertisseur : Impédance et dans notre cas une résistance



ou

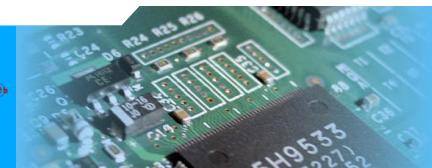


Loi d'ohm



$$U = R \cdot I$$

Remarque : U et I ont la même forme



6.1. Fonction 1 : conversion courant/tension

Courant

- Quantité d'électricité qui parcourt un conducteur
- Débit de charges électriques dans ce conducteur.
- Appelé aussi intensité (électrique)
- Noté I et exprimé en Ampère (A).

Analogie avec une rivière :

- le lit de la rivière = le conducteur électrique
- le débit d'eau = l'intensité électrique

Tension

- Différence de 2 potentiels électriques
- Liée à une énergie potentiellement disponible (voir l'analogie avec la rivière)
- $U_{AB} = V_A - V_B$ et exprimée en Volt (V).

Analogie avec une rivière :

- l'altitude = le potentiel électrique
- la différence d'altitude entre le début et la fin de la rivière (le dénivelé) = la différence de potentiels (la tension)

Source : sophie.allain@univ-rennes1.fr



6.1. Fonction 1 : conversion courant/tension

Applications

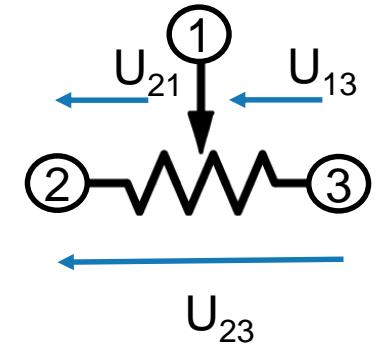
- Gérer le volume audio sur un ampli



Source : www.marantz.com



Potentiomètre



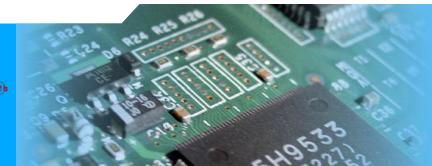
$$U_{23} = U_{21} + U_{13}$$



$$U_{23} = \text{constante}$$



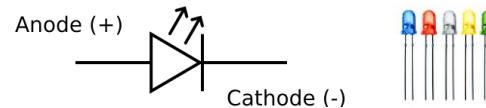
Déplacement du point 1 vers la gauche : U_{21} diminue et U_{13} augmente
 Déplacement du point 1 vers la droite : U_{21} augmente et U_{13} diminue



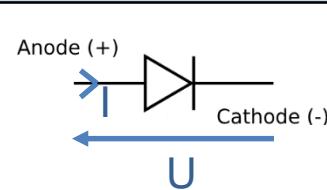
6.2. Fonction 2 : conversion tension/lumière

TP - Carte capteurs : Affichage-LED

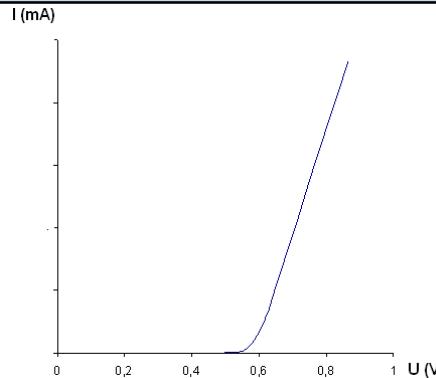
LED ? => Light-Emitting Diode (ou DEL : Diode Electroluminescente)



Diode ? => Composant électronique : dipôle asymétrique



Si $U >$ Tension de seuil, la diode conduit ($I > 0$ A)
 Sinon la diode est dite bloquée ($I = 0$ A)...



Source : openclassrooms.com

Une LED = une diode => Si elle est conductrice => elle éclaire !!!



6.3. Fonction 3 : filtrage

Filtrage : Fonction très utilisée dans les systèmes électroniques

- => Traitement du signal
- => Modifications des caractéristiques spectrales d'un signal électrique

Objectif du filtrage : Extraire une information utile

« *Le but du filtrage est d'extraire une partie de l'information, afin de restituer un signal plus intelligible* »

Source : Techniques de l'ingénieur

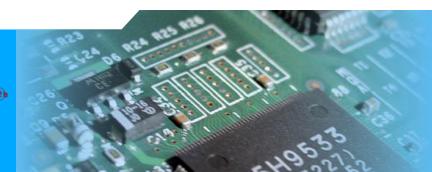
Filtre : Ensemble de composants électroniques permettant de modifier la forme d'un signal électrique (spectre de fréquence et/ou phase)

Filtres : **analogiques** et numériques



$$H(f) = \frac{V_2}{V_1}$$

Fonction de transfert $H(f)$: Rapport entre la tension de sortie et la tension d'entrée



6.3. Fonction 3 : filtrage

Rappels :

signal sinusoïdal => amplitude, fréquence, phase

$$S(t) = A \sin(\omega t + \varphi)$$

avec

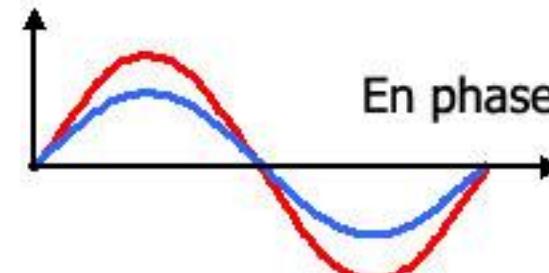
A : Amplitude du signal (en volts si S est une tension)

ω : pulsation en $\text{rad}^{-1}/\text{s} = 2\pi f = 2\pi/T$

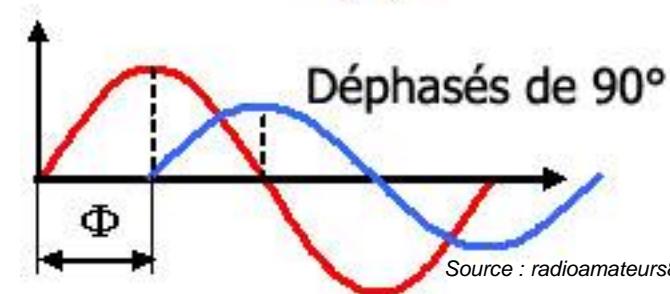
F : fréquence en Hz

T : période en s

φ : phase à l'origine en rad



Déphasage entre 2 signaux : retard



6.3. Fonction 3 : filtrage

Caractérisation d'un filtre : deux courbes en fonction de la fréquence permettent d'expliciter les modifications de forme entre V_2 et V_1

- . amplitude/fréquence => gain en fonction de la fréquence
- . phase/fréquence => déphasage en fonction de la fréquence

Fonction de transfert $H(f)$:

$$\text{Gain} = |V_2/V_1|$$

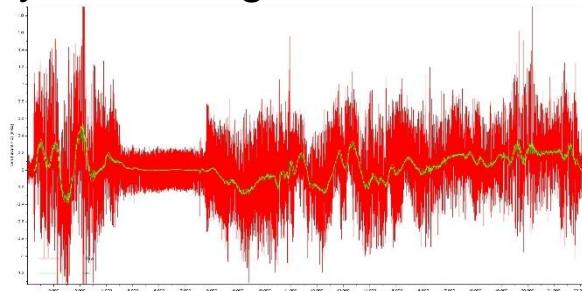
$$\text{Phase} = \text{Arg}(V_2/V_1)$$

$$H(f) = \frac{V_2}{V_1}$$



6.3. Fonction 3 : filtrage

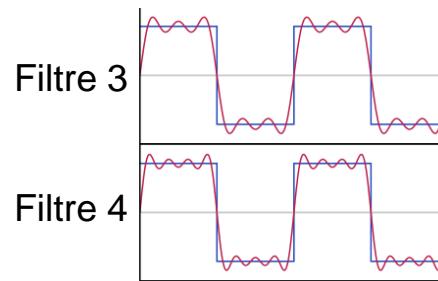
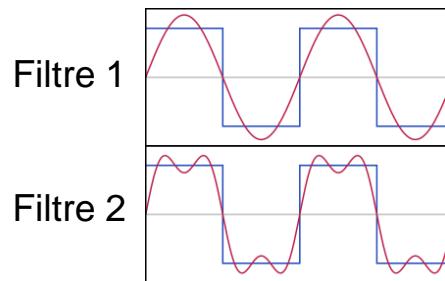
Applications : - « Nettoyer » un signal bruité



- Signal non filtré
- Signal filtré

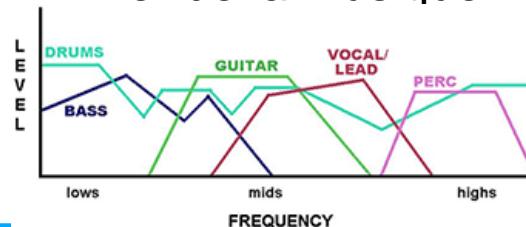
Source : www.developpez.net

- Modifier la forme d'un signal

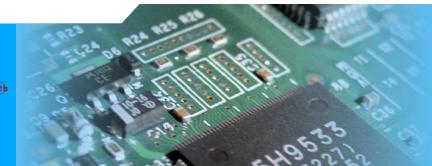


- Signal non filtré (original)
- Signal filtré (modifié)

- ### - Mixer de la musique

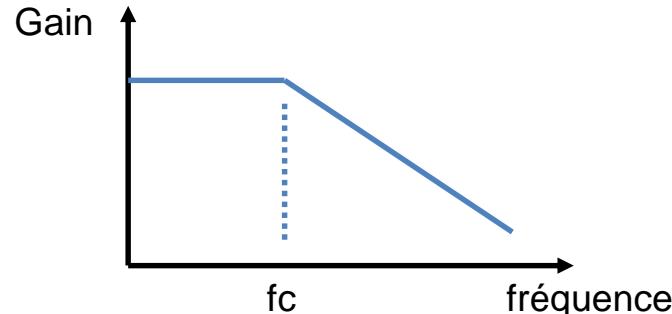


Source : fr.audiofanzine.com



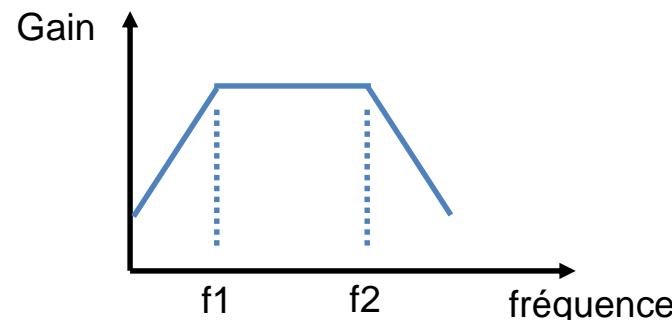
6.3. Fonction 3 : filtrage

passe-bas



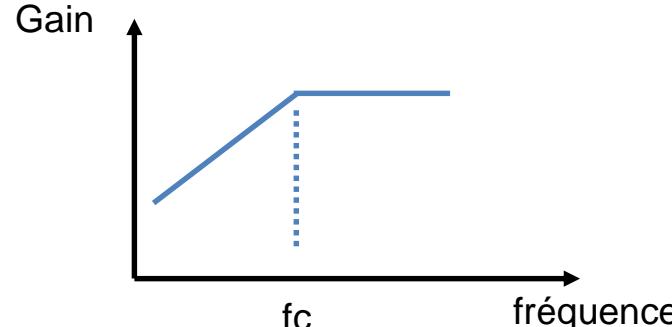
signal atténué lorsque $f > f_c$

passe-bande



signal atténué lorsque $f < f_1$ et $f > f_2$

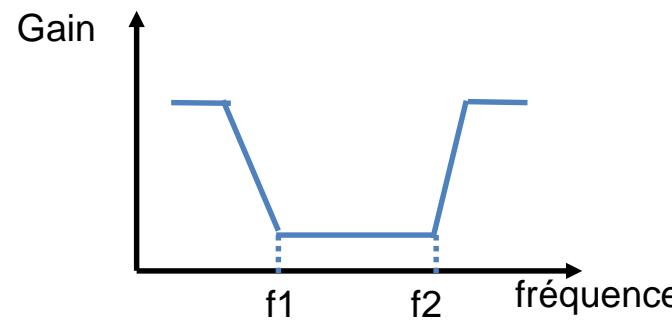
passe-haut



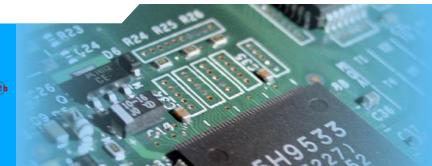
signal atténué lorsque $f < f_c$

F_c : fréquence de coupure

coupe bande ou réjecteur



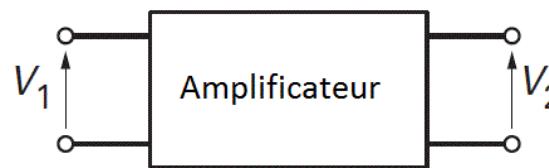
signal atténué lorsque $f_1 < f < f_2$



6.4. Fonction 4 : amplification

Objectif de l'amplification : Augmenter l'amplitude d'un signal sans en modifier la forme

Amplificateur : Composants électroniques permettant de multiplier un signal par une constante, appelée gain

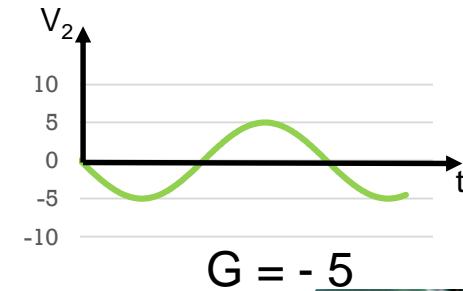
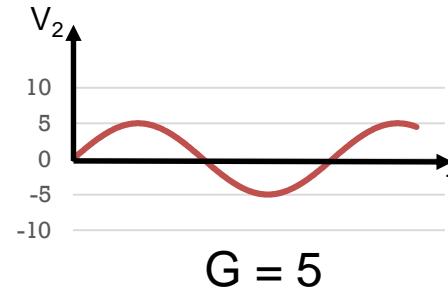
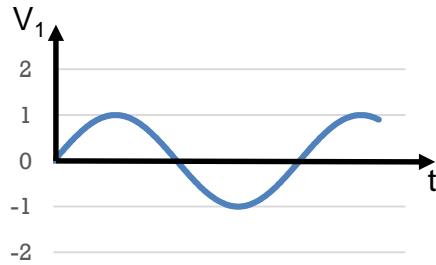


$$G = \frac{V_2}{V_1}$$

$G = \text{Constante}$

Si $G > 0 \Rightarrow$ Amplificateur non inverseur

Si $G < 0 \Rightarrow$ Amplificateur inverseur



6.4. Fonction 4 : amplification

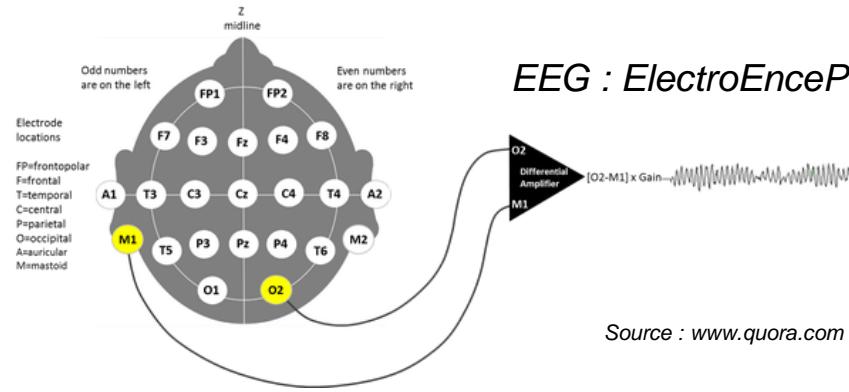
Applications :

- Amplificateur de guitare



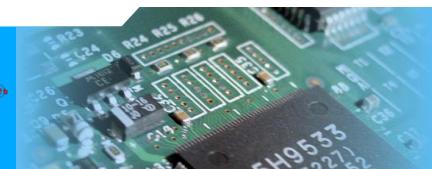
Source : fr.audiofanzine.com

- Amplificateur biomédical => Signaux électrophysiologiques



EEG : ElectroEncePhalogramme

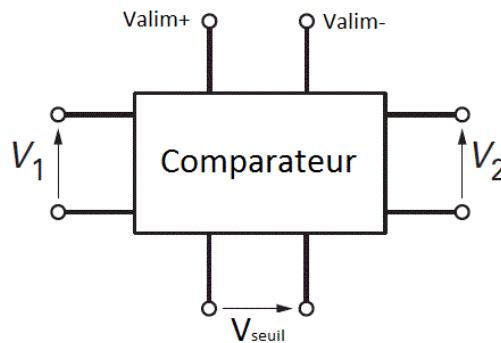
Source : www.quora.com



6.5. Fonction 5 : comparaison

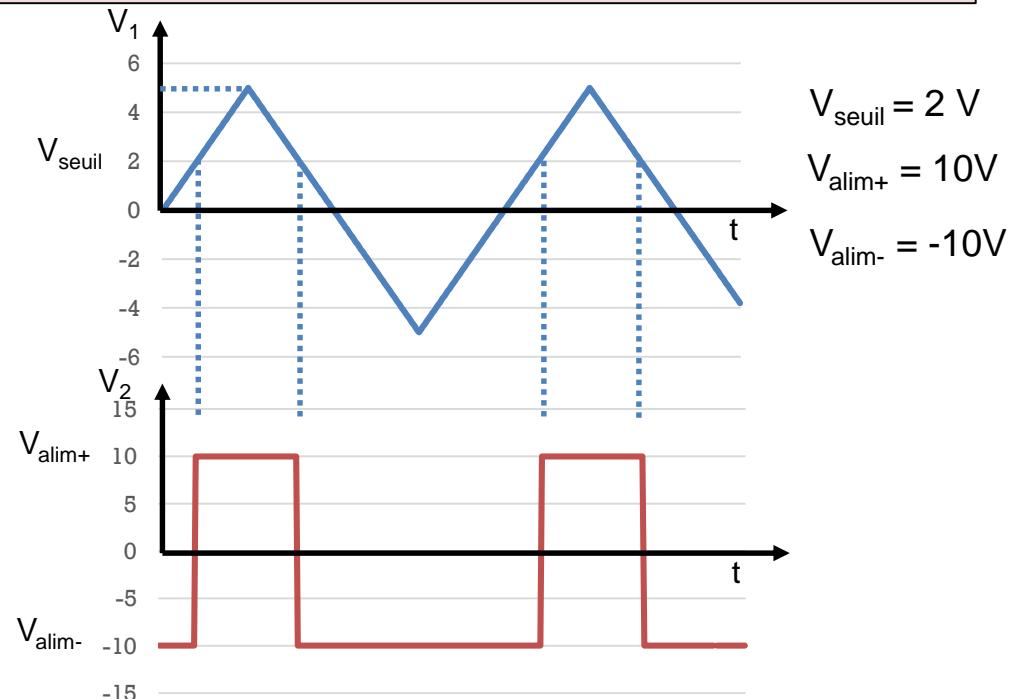
Objectif de la comparaison : Comparer l'amplitude d'un signal à une amplitude de référence appelée seuil

Comparateur : Composants électroniques permettant de générer une sortie avec uniquement 2 états (haut et bas)



Si $V_1 > V_{\text{seuil}} \Rightarrow V_2 = V_{\text{alim}+}$

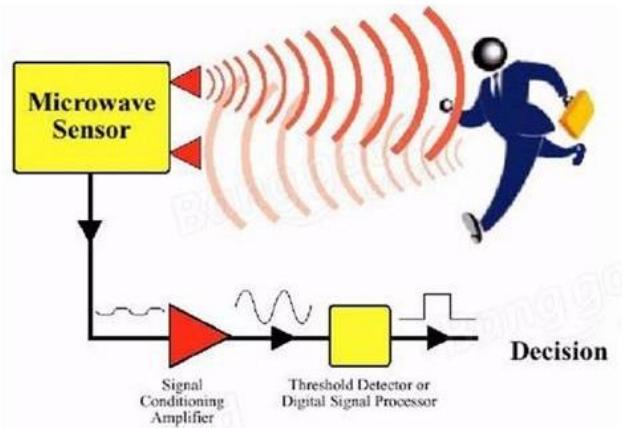
Si $V_1 < V_{\text{seuil}} \Rightarrow V_2 = V_{\text{alim}-}$



6.5. Fonction 5 : comparaison

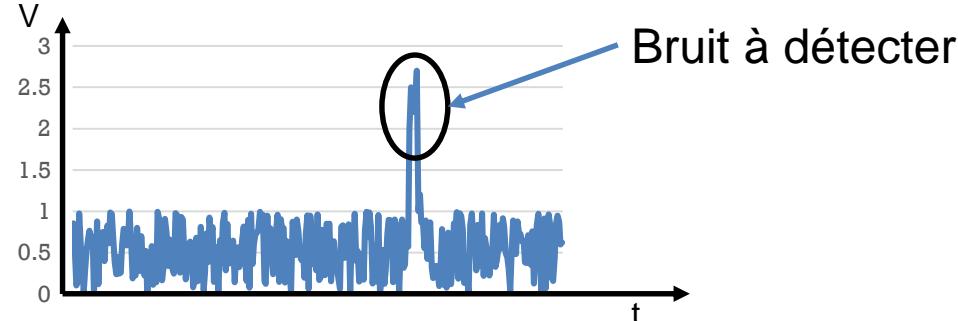
Applications :

- DéTECTeur de mouvement

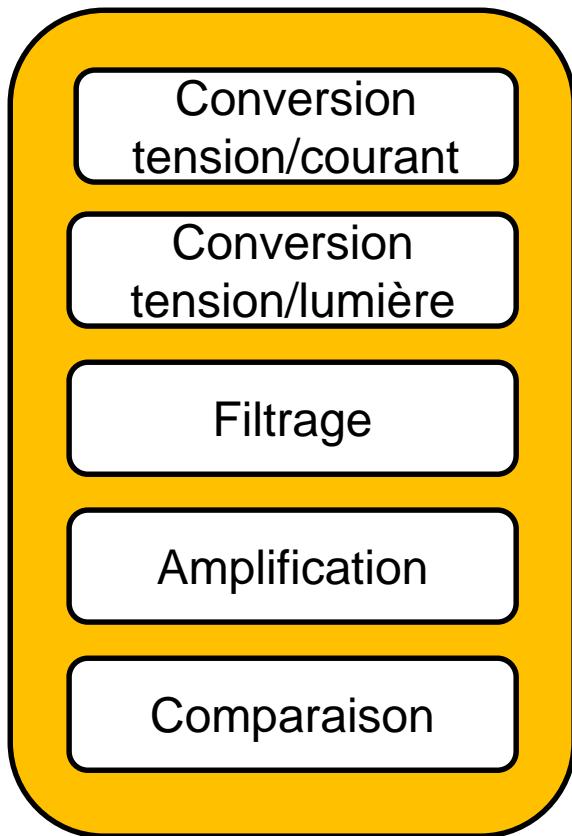


Source : www.banggood.com

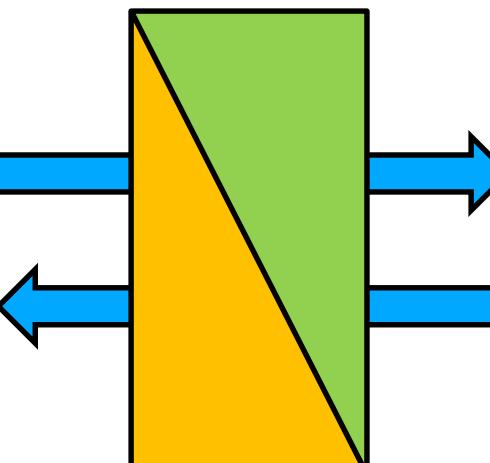
- DéTECTeur d'un bruit bref



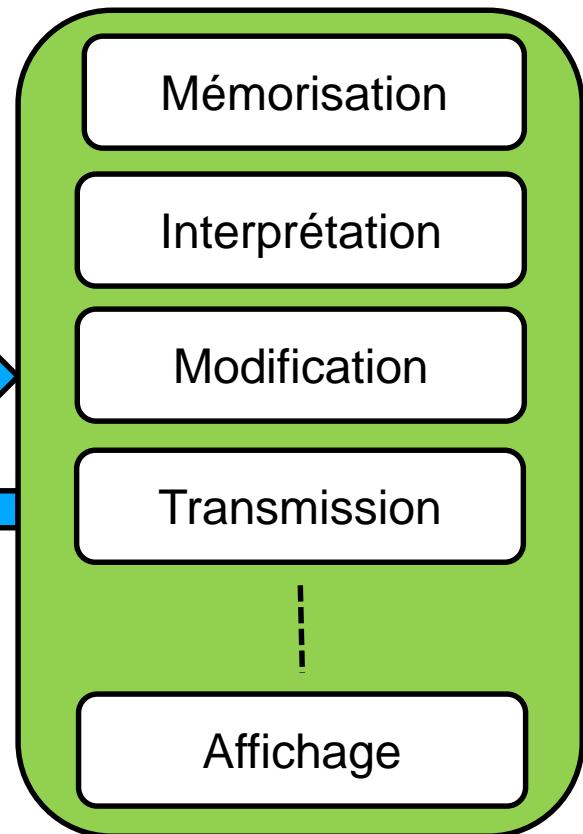
Fonctions analogiques



Conversion
analogique/numérique



Fonctions numériques



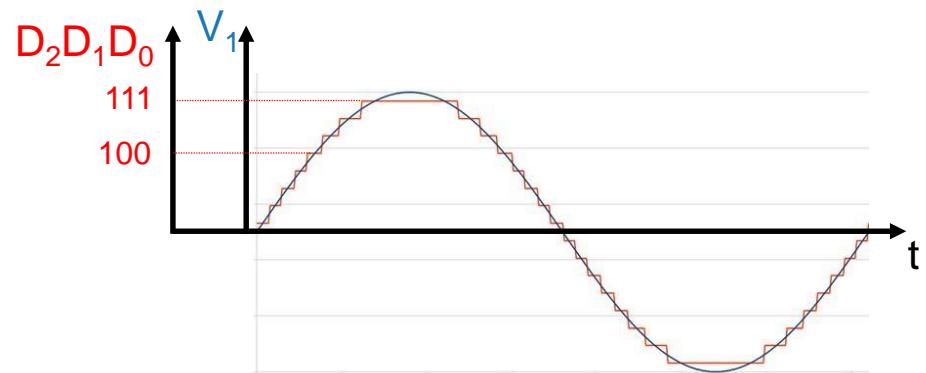
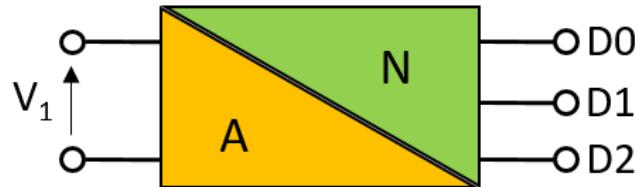
6.6. Fonction 6 : Conversion analogique numérique

Conversion analogique numérique : Fonction indispensable pour interagir avec l'environnement

- => Environnement : monde analogique
- => Informatique : monde numérique

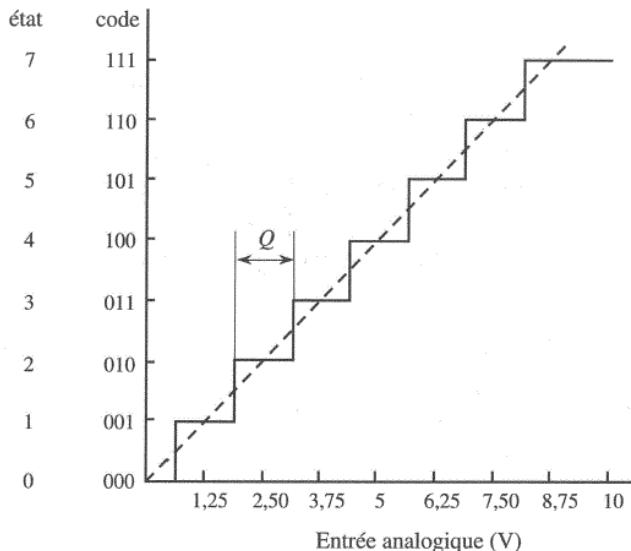
Objectif du convertisseur : Convertir un signal analogique (tension ou courant) en une donnée numérique

Convertisseur : Ensemble de composants analogiques et numériques



6.6. Fonction 6 : Conversion analogique numérique

Caractéristique de transfert d'un CAN



PE : Pleine échelle ou plage de conversion
 Code : Représentation de la donnée numérique
 Q : Quantum ou pas de quantification

CAN n bits : 2^n-1 niveaux analogiques de décision

Avec

Pleine échelle

$$Q = \frac{\text{Pleine échelle}}{\text{Nombre d'états de sortie}}$$

Nombre d'états de sortie



Source : JP Troadec, Dunod

Erreur de quantification : Perte d'information

Caractérisation de transfert d'un CAN

3 bits : $2^3 = 8$ états possibles

Pleine échelle : 10 V



$$Q = 10/8 = 1,25 \text{ V}$$

6.6. Fonction 6 : Conversion analogique numérique

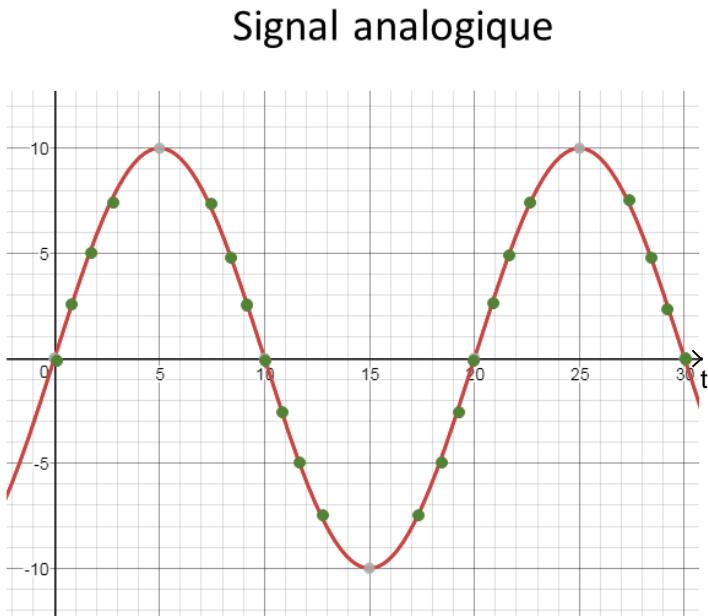
Résolution, nombre d'états et valeur du LSB en fonction de la pleine échelle

Résolution N	Nombre d'états 2^N	Valeur du LSB (% de la pleine échelle)	Valeur du LSB avec 10 V de pleine échelle (mV)
8	256	0,390 625	39,062
10	1 024	0,097 656	9,765
12	4 096	0,024 414	2,441
14	16 384	0,006 104	0,610
16	65 536	0,001 526	0,152
18	262 144	0,000 381	0,038
20	1 048 576	0,000 095	0,009
22	4 194 304	0,000 024	0,002
24	16 777 216	0,000 006	0,000 596
26	67 108 864	0,000 001	0,000 149

Source : techniques de l'ingénieur

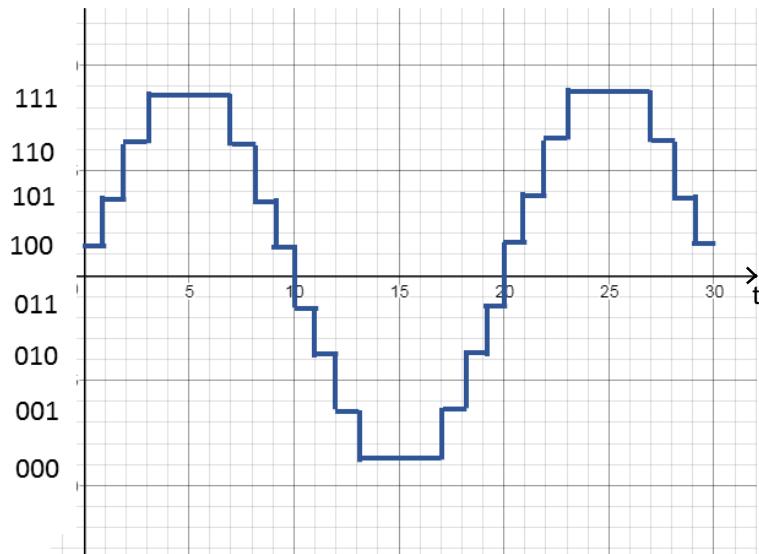
6.6. Fonction 6 : Conversion analogique numérique

Entrée du CAN en LSB



Sortie du CAN en LSB

Numérisation 3 bits



Source : <http://electroniqueamateur.blogspot.com>

Signal numérique \neq Signal analogique \rightarrow Perte d'information

6.6. Fonction 6 : Conversion analogique numérique

TP - Carte capteurs : utilisation d'un ADC0804

Caractéristiques

Résolution : 8 bits

- . 256 données différentes possibles en sortie

Plage de conversion : 5 volts

- . possibilité de convertir une tension U avec $0 < U < 5 \text{ V}$

Temps de conversion : $100 \mu\text{s}$

- . possibilité de convertir une tension toutes les $100 \mu\text{s}$
- . fréquence de conversion = $1/100 \mu\text{s} = 10 \text{ kHz}$

Boîtier (chip) : 20 broches (pins)

4 signaux de contrôle

8 signaux de données => 8 bits

2 signaux d'entrée => Tension à convertir

2 alimentations

3 signaux de configuration

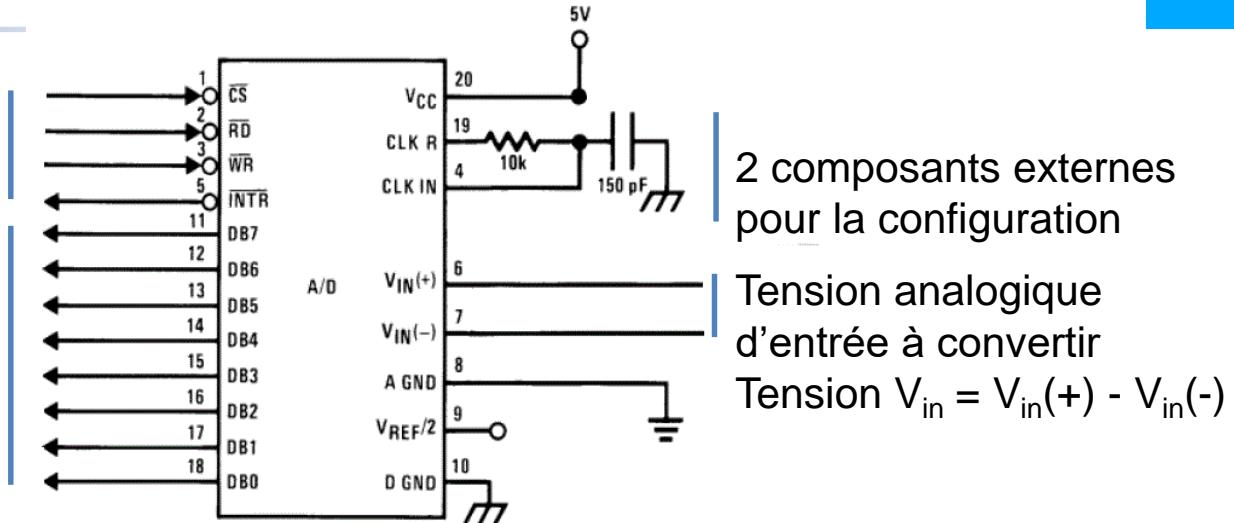


6.6. Fonction 6 : Conversion analogique numérique

TP - Carte capteurs : utilisation d'un ADC0804

4 signaux de contrôle
3 entrées et 1 sortie

8 données numériques de sortie



2 composants externes pour la configuration

Tension analogique d'entrée à convertir
Tension $V_{in} = V_{in}(+) - V_{in}(-)$

TP : chaque donnée numérique (sortie) est connectée à une LED

Exemple : si DB7 = 0 alors la LED D7 est allumée

si DB7 = 1 alors la LED D7 est éteinte

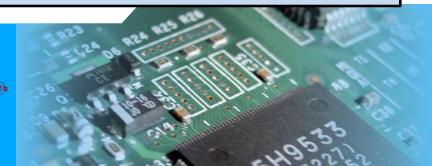
TP : entrées de contrôle /CS et /RD mises à 0 => Connectées au GND (0 volt)

entrée /WR : signal généré par un composant externe au CAN

- ordre de conversion (écriture) si /WR passe de 1 à 0 => **début conversion**

sortie /INTR : signal généré par le CAN

- information de fin de conversion si /INTR = 0 => **conversion terminée**



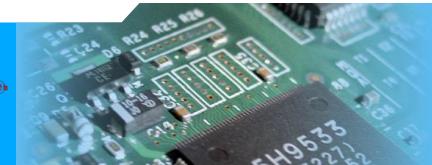
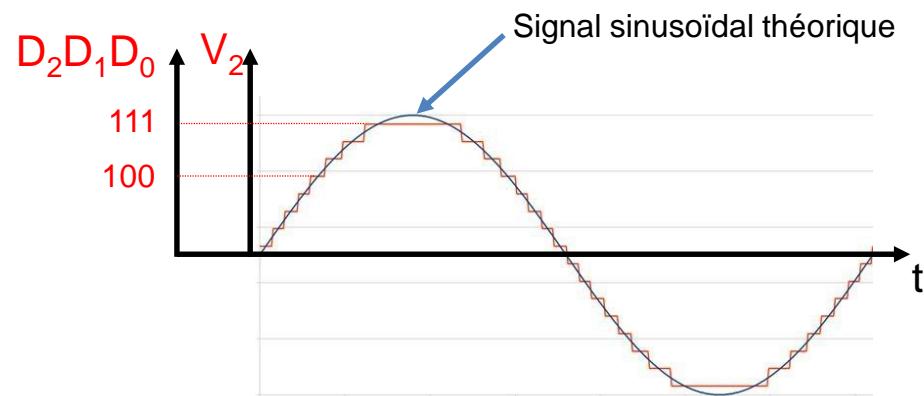
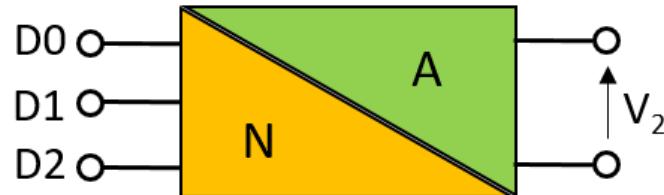
6.7. Fonction 7 : Conversion numérique analogique

Conversion numérique analogique : Fonction indispensable pour interagir avec l'environnement

- => Environnement : monde analogique
- => Informatique : monde numérique

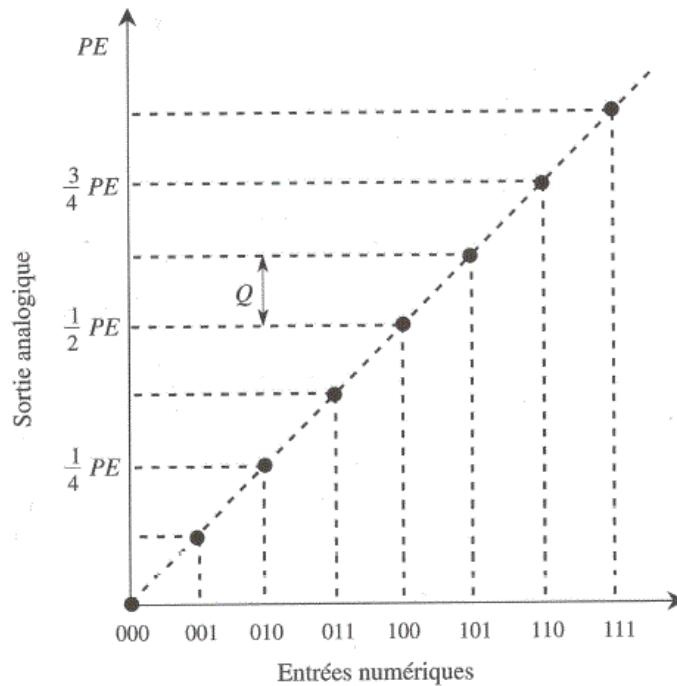
Objectif du convertisseur : Convertir un signal numérique (tension ou courant) en une donnée analogique

Convertisseur : Ensemble de composants analogiques et numériques



6.7. Fonction 7 : Conversion numérique analogique

Caractéristique de transfert d'un CNA



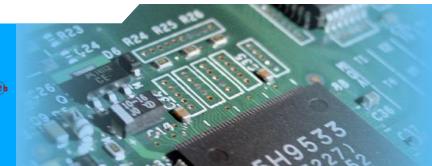
PE : Pleine échelle ou plage de conversion
 Code : Représentation de la donnée numérique
 Q : Quantum ou pas de quantification

Une combinaison d'entrée génère une valeur analogique correspondant à une fraction de la PE

Pas de perte d'information

Source : JP Troadec, Dunod

Caractérisation de transfert d'un CNA 3 bits

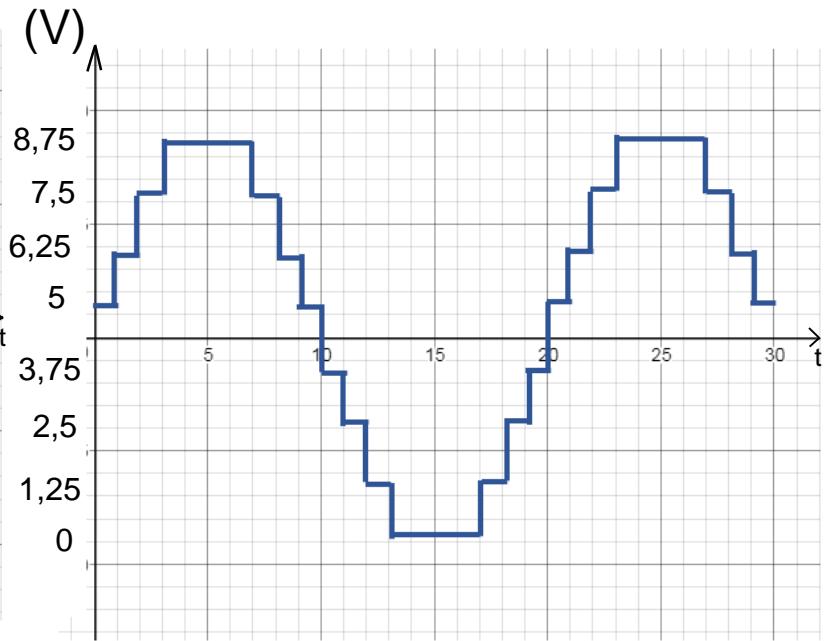
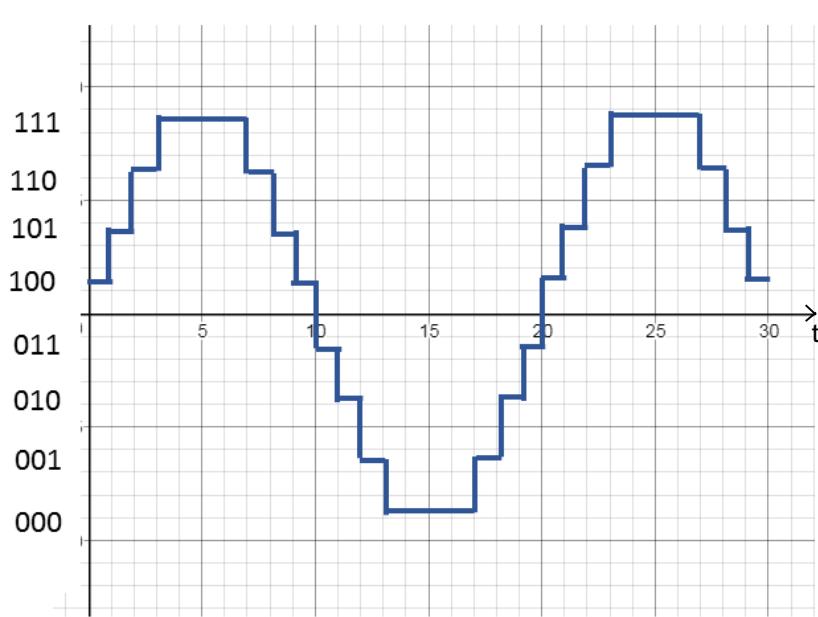


6.7. Fonction 7 : Conversion numérique analogique

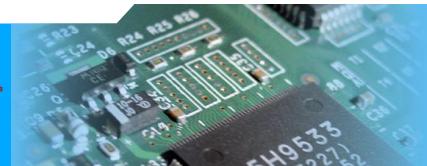
Entrée du CNA en binaire

Sortie du CNA en V

Numérisation 3 bits



Signal numérique = Signal analogique → Pas de Perte d'information



6.7. Fonction 7 : Conversion numérique analogique

Principe du codage

Numérique : utilisation de la base 2 pour représenter un signal numérique

Tout entier N positif peut être représenté en numération binaire :

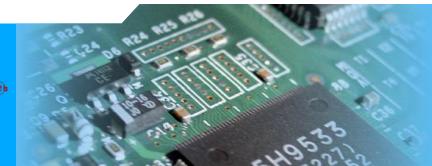
$$N = a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_1 2^1 + a_0 2^0 \quad \text{avec } (a_i = 0 \text{ ou } 1)$$

Exemple sur 3 bits :

$$N_{10} = 6 = 1*2^2 + 1*2^1 + 0*2^0$$

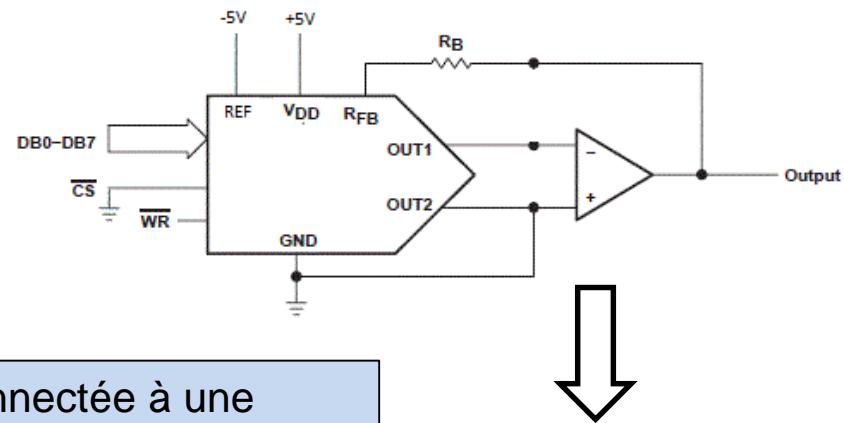
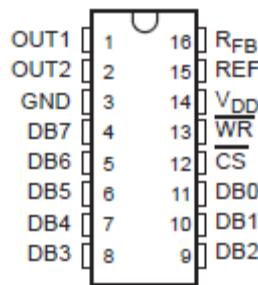
$$N_2 = 110$$

Source : JP Troadec, Dunod



6.7. Fonction 7 : Conversion numérique analogique

TP - Carte capteurs : utilisation d'un TLC7524



TP : chaque donnée numérique (entrée) est connectée à une sortie du microcontrôleur

TP : /CS : entrée de contrôle mise à 0 => Connectée au GND

/WR : entrée et signal généré par un composant externe (microcontrôleur) au CNA

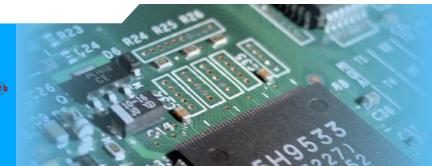
ordre de conversion (écriture)

si /WR passe de 1 à 0 => **début conversion**

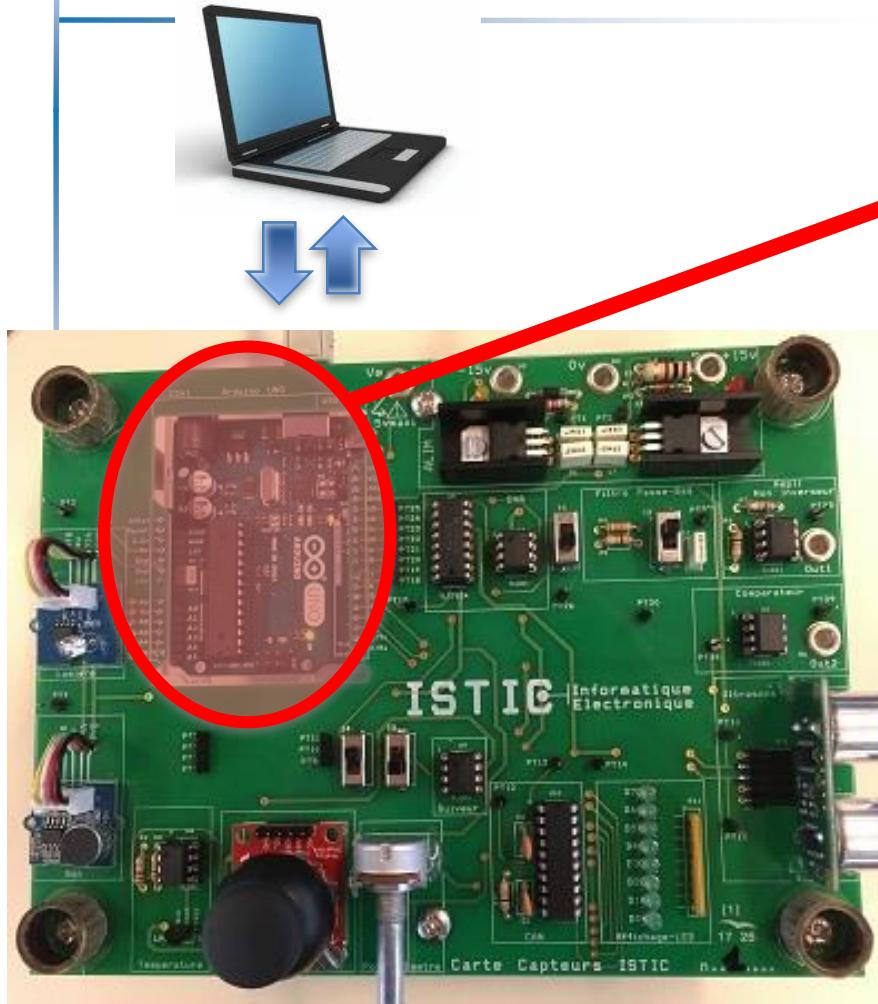
DIGITAL INPUT (see Note 3)		ANALOG OUTPUT
MSB	LSB	
1	1	$-V_{ref} (255/256)$
1	0	$-V_{ref} (129/256)$
1	0	$-V_{ref} (128/256) = -V_{ref}/2$
0	1	$-V_{ref} (127/256)$
0	0	$-V_{ref} (1/256)$
0	0	0

NOTE 3: LSB = 1/256 (V_{ref})

Source : www.ti.com



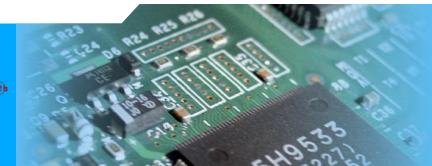
5. Le microcontrôleur (rappels - compléments)



- Collecter les données des capteurs
- Stocker les données des capteurs
- Interpréter les données des capteurs



- Assurer la communication entre l'extérieur (PC par exemple) et la carte capteurs
- Transmettre les données des capteurs



5. Le microcontrôleur (rappels - compléments)



Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Source : <http://arduino.cc/>

5. Le microcontrôleur (rappels - compléments)

Un microcontrôleur = 1 Ordinateur « comme à la maison ou à l'université » ?

OUI

Unité Centrale (Processeur)
Mémoire vive
Mémoire morte
Interfaces pour communiquer



Ca se programme donc ?... OUI

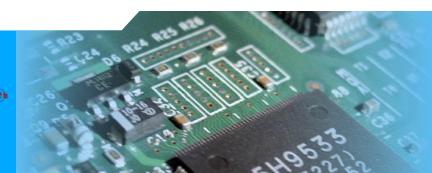
NON

Ressources très limitées



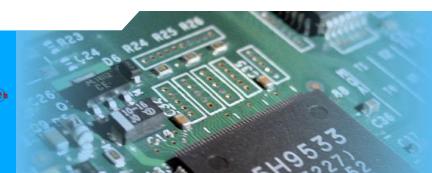
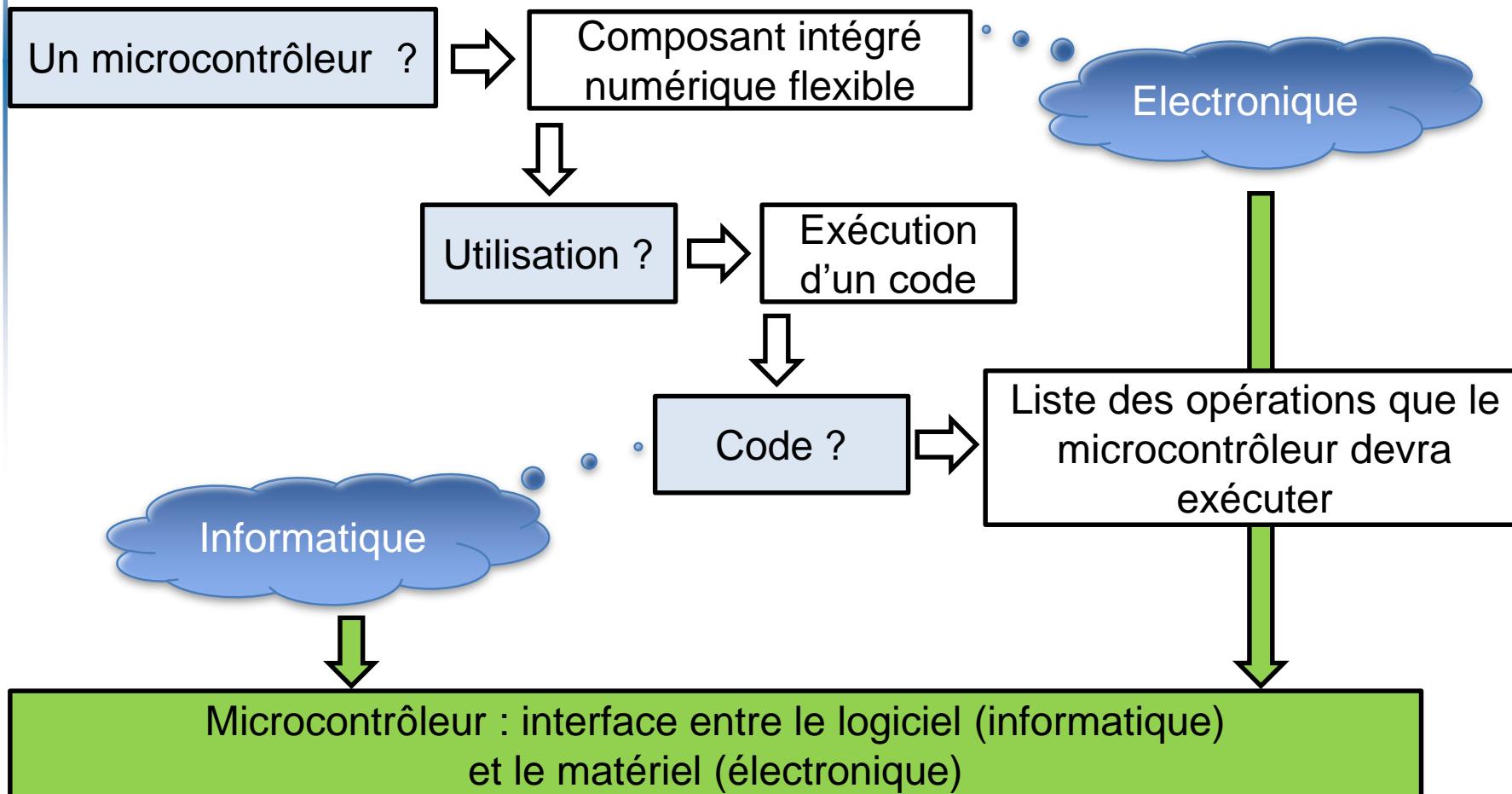
C'est peu utilisé ?... NON

Il y en a « presque » partout... dans une voiture : plus d'une centaine de microcontrôleurs



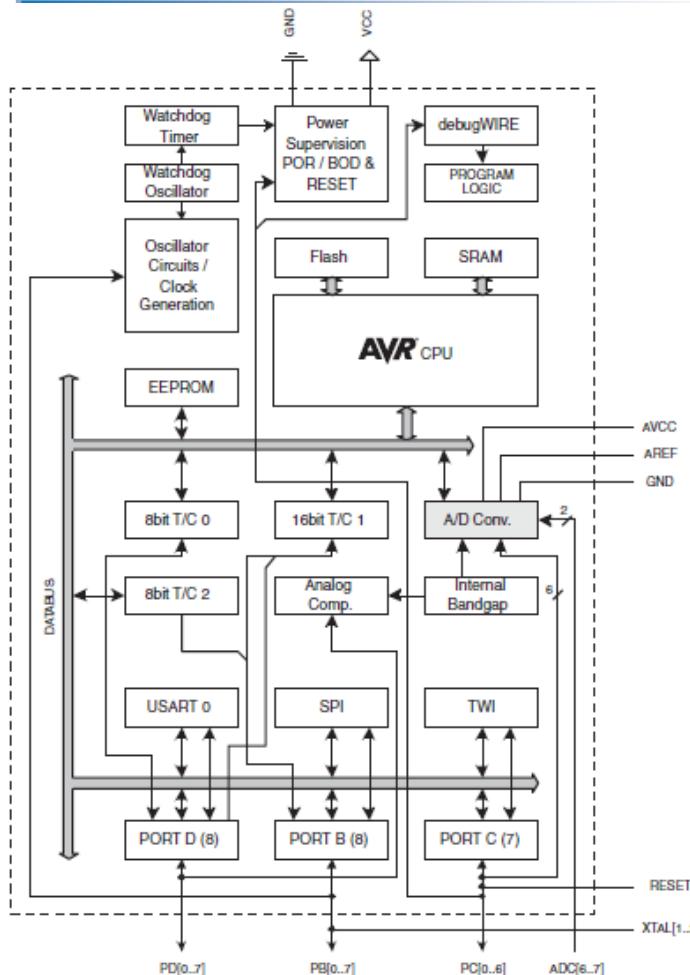
5. Le microcontrôleur

(rappels - compléments)



5. Le microcontrôleur

(rappels - compléments)



ATmega328 et non... Arduino



- Architecture RISC (Reduced instruction set computer)
- Low Power
- In System programmable system
- 20 MIPS (Mega Instructions Per Second)
- 3 types de mémoire (Flash, SRAM, EEPROM)
- 6 A/D Converters de 10 bits
- 6 PWM (*Pulse Width Modulation*)

5. Le microcontrôleur

(rappels - compléments)

Les mémoires internes

Flash memory

- Non volatile
- Rapide
- Stockage du Bootloader (programme initial)
- Stockage du programme application

SRAM : Static Random Access Memory

- Mémoire vive
- Volatile
- Stockage intermédiaire des données



EEPROM : Electrically Erasable Programmable Read Only Memory

- Non volatile
- Stockage de paramètres utiles à l'application

7. L'environnement logiciel Arduino

IDE : Integrated development environment

```

Arduino - 0011 Alpha
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */
int ledPin = 13;          // LED connected to digital pin 13
void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}
void loop()                // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);             // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  delay(1000);             // waits for a second
}

Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)
22

```

Fonctions de l'IDE

- Compilation du code programme
- Transfert du code compilé dans la mémoire flash du microcontrôleur

Voir www.arduino.cc



7. L'environnement logiciel Arduino

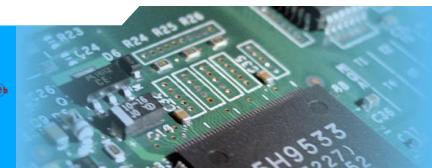
Un programme = un « sketch »

Un programme : 2 fonctions obligatoires

Setup : Instructions d'initialisation

Loop : Instructions répétées indéfiniment

```
// Déclarations des constants, variables, bibliothèques,  
...  
void setup()  
{  
    // Instructions exécutées une fois  
}  
void loop()  
{  
    // Instructions exécutées indéfiniment  
}
```



7. L'environnement logiciel Arduino

Quelques fonctions ...

Pause :

`delay(durée du délai)`

Exemple : Pause de 100 millisecondes

`delay(100);`

Définition des entrées/sorties numériques

`pinMode(nom de la broche, mode de la broche)`

Exemple : Définition d'une sortie

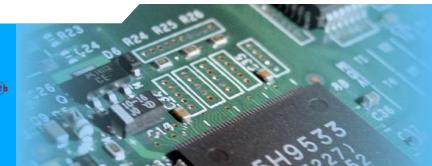
`int enable = 14;
pinMode(enable, output);`

Durée d'une impulsion

`durée = pulseIn(entrée, type, délaimax)`

Exemple :

`duration = pulseIn(in1, HIGH);`



7. L'environnement logiciel Arduino

Modification des sorties numériques

`digitalWrite(nom de la broche, valeur)`

Exemple : Mise à l'état haut d'une sortie enable

`digitalWrite(enable, HIGH);`

Lecture des entrées numériques

`Valeur = digitalRead(nom de la broche)`

Exemple : Lecture de l'entrée in1

`Valeur = digitalRead(in1);`

Lecture des entrées analogiques : présence de CAN

`Valeur = analogRead(nom de la broche)`

Exemple : Lecture de l'entrée ina1

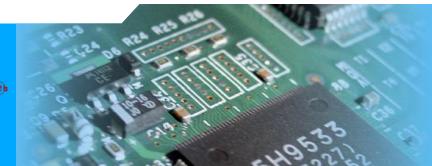
`Valeur = analogRead(ina1);`

Modification des sorties analogiques : Absence de CNA – utilisation de PWM

`analogWrite(nom de la broche, rapport cyclique)`

Exemple : Mise à VCC/2 d'une sortie outa1 (filtrage en aval nécessaire)

`analogWrite(outa1, 50);`



7. L'environnement logiciel Arduino

Communication avec le PC

Utilisation du port série

La carte Arduino UNO met à disposition un port série asynchrone

- Broche 0 « réception » : Rx
- Broche 1 « émission » : Tx

La carte Arduino UNO dispose d'une interface USB pour le chargement du soft

- Création d'un port COM virtuel
- Utilisation possible d'un `serialMonitor` dans l'IDE ou d'un hyperterminal

Initialisation du port Serial

`Serial.begin(vitesse)`

Fonction à prévoir dans le `setup`

`vitesse` = Vitesse standard de transmission

Exemple : initialisation du port série asynchrone à 9600 bauds

`Serial.begin(9600); // 9600 bits/sec`



7. L'environnement logiciel Arduino

Communication avec le PC

Désactivation du port Serial

```
Serial.end(); // broches 0 et 1 peuvent être utilisées
```

Lecture du port série

```
caractère = Serial.read();
```

Ecriture dans le port série

```
Serial.print(données, format);
```

La fonction assure les conversions

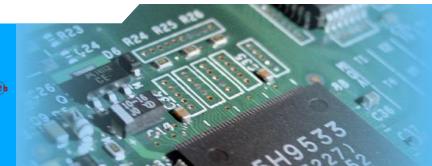
Exemples

```
Serial.print(17); //affichage de 17
```

```
Serial.print(17,BIN); //affichage de 10001
```

```
Serial.print(<< licence >>); // affichage de licence
```

```
Serial.println() // avec un retour à la ligne
```



8. Les capteurs

Méthode :

Présentation rapide en cours
Expérimentations en TP

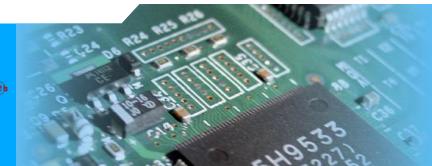
8.1 Joystick

8.2 Capteur ultrason

8.3 Capteur de température

8.4 Capteur de luminosité

8.5 Capteur de son



8. Les capteurs

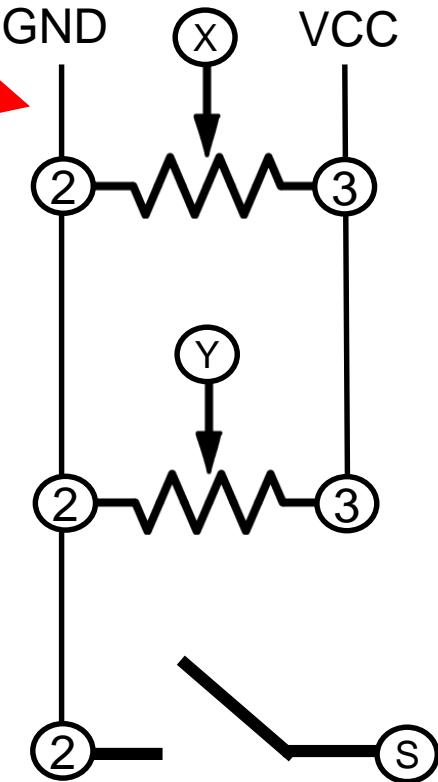
8.1. Joystick



Joystick

2 axes : X et Y
un interrupteur

- Les tensions aux points X et Y, V_X et V_Y sont proportionnelles à la position des points X et Y
- Si l'interrupteur est fermé : $V_S = \text{GND}$
- Si l'interrupteur est ouvert : $V_S = \text{VCC}$
(si on rajoute une résistance dite de pull-up)



Applications

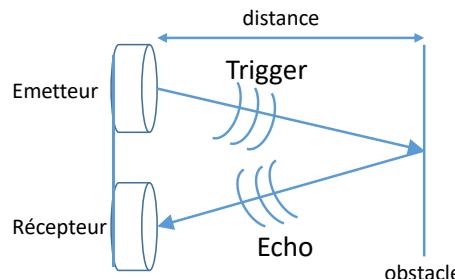
- Manette de jeux



8.2 Capteur ultrason

Objectif : Calcul de la distance entre le capteur et un obstacle

Principe : Calcul du temps entre l'instant d'émission d'un signal ultrason et l'instant de réception de ce signal après réflexion sur un obstacle



Source : www.adnauto.fr

Applications :

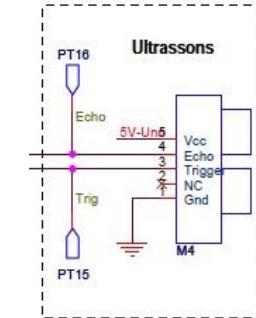
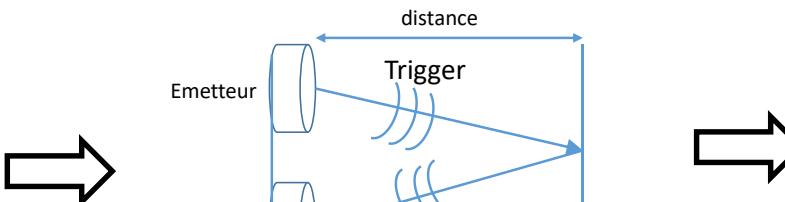
- Radar de recul pour des véhicules
- Robotique : capteur de positionnement



Source : www.robot-maker.com

8. Les capteurs

8.2 Capteur ultrason



Trigger : Entrée du capteur – ordre d'émission généré par le microcontrôleur
 Echo : Sortie du capteur – information récupérée par le microcontrôleur

Rôle du microcontrôleur



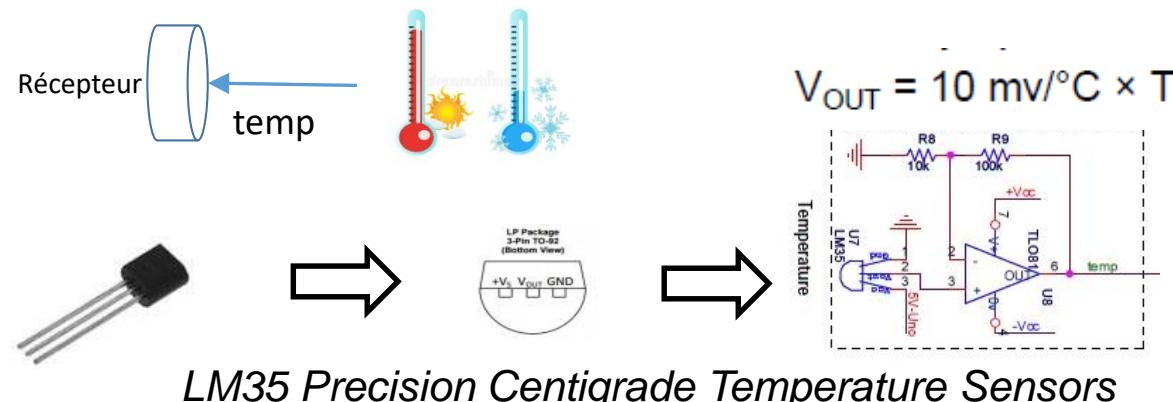
- Contrôler le capteur => Trigger
- Récupérer l'information => Echo
- Interpréter l'information
- Calculer la distance
- Transmettre la distance



8.3 Capteur de température

Objectif : Calcul de la température ambiante

Principe : Mesure d'une tension qui varie en fonction de la température ambiante



V_{out} : Tension analogique

Rôle du microcontrôleur

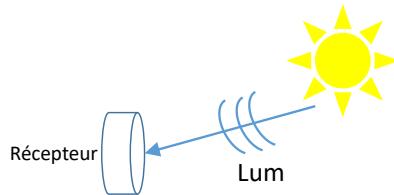
- Récupérer l'information => V_{out}
- Convertir la tension analogique en numérique
- Interpréter l'information
- Transmettre la température



8.4 Capteur de lumière

Objectif : mesurer le taux de luminosité ambiant (éclairage)

Principe : utilisation d'une cellule photoconductrice dont la résistance varie en fonction de la luminosité => Tension analogique



$$1 \text{ lux} = 1 \text{ lumen/m}^2$$

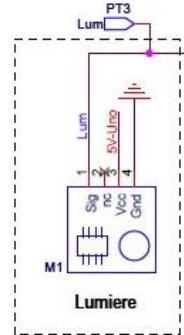
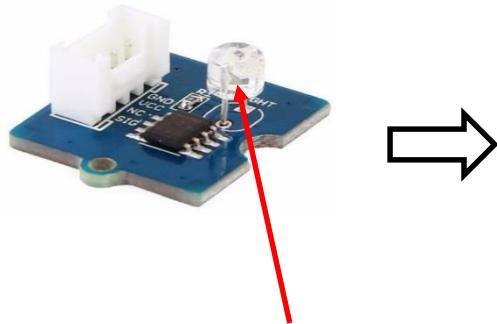
Activité ou lieu concerné	Éclairement moyen
Sensibilité d'une caméra	0,001 lux
Nuit de pleine lune	0,5 lux
Rue de nuit bien éclairée	20 à 70 lux
Local de vie	100 à 200 lux
Appartement bien éclairé	200 à 400 lux
Local de travail	200 à 3 000 lux

Source . wikipedia

Applications :

- Adaptation de l'éclairage à la luminosité ambiante
- Automobile : allumage de phare automatiquement

8.4 Capteur de lumière



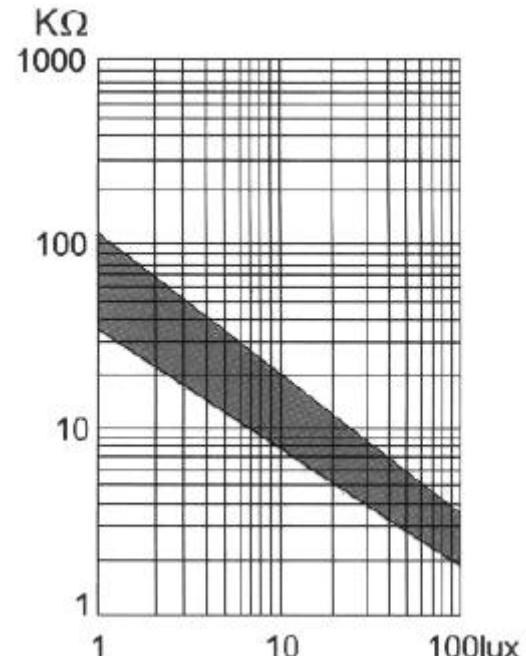
Cellule photoconductrice GL5528

Lum : Tension analogique

Rôle du microcontrôleur



- Récupérer l'information => Lum
- Convertir la tension analogique en numérique
- Interpréter l'information
- Transmettre la luminosité

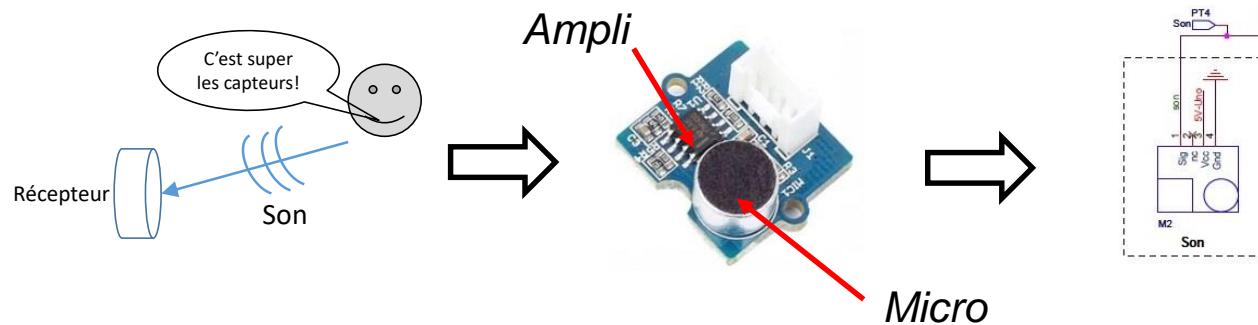


8. Les capteurs

8.5 Capteur de son

Objectif : mesurer l'ambiance sonore

Principe : utilisation d'un microamplifié => Tension analogique



Son : Tension analogique

Rôle du microcontrôleur

- Récupérer l'information => son
- Convertir la tension analogique en numérique
- Traitement de l'information => filtrage, moyennage, détection de max, ...
- Extraire et transmettre l'information



Fusion des données

Pourquoi ?

Transmission des données « capteurs » à une machine (PC)

Comment ?

Utilisation de la liaison série du microcontrôleur et USB du PC

Définition d'un protocole : langage commun

Ecriture du code

```
balisedebut  
joystick,x,514  
joystick,y,123  
joystick,bp,0  
temperature,19  
ultrason,25  
son,72  
lumiere,2  
balisefin
```

