



# Capteurs

Ou faire en sorte que l'on puisse faire la différence entre votre robot et Gilbert Montagné.

D'après Liam LOTTE

**AREM : CDR**



## ***Sommaire :***

### II) Capteurs

- 1) Kesako un capteur et une chaîne d'information ?
- 2) Types de capteurs et leurs sorties
- 3) Protocoles de communications classiques
- 4) Les capteurs du robot de l'année dernière

**AREM : CDR**



## ***1) Kesako un capteur et une chaîne d'information ?***

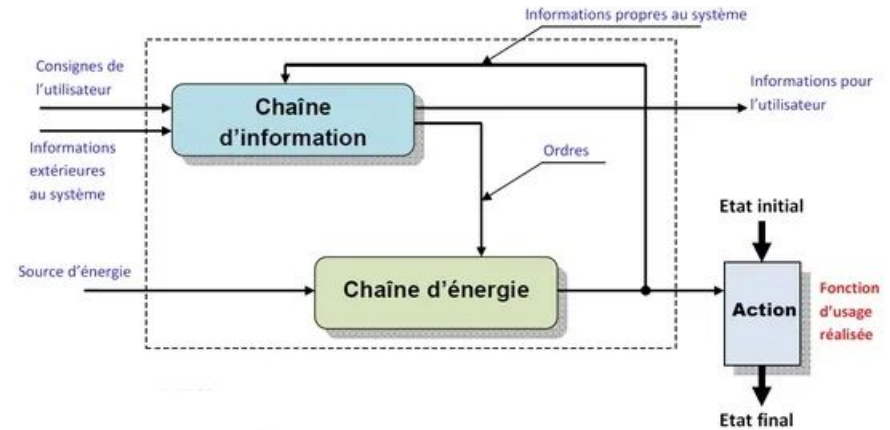
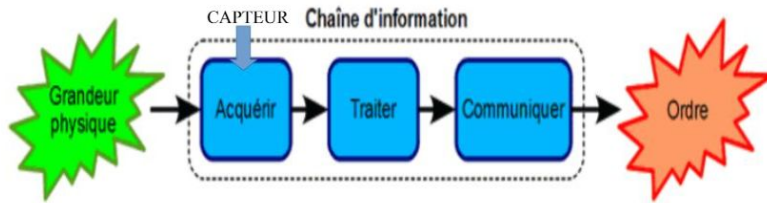
“Un **capteur** est un dispositif transformant l'état d'une **grandeur physique** observée en une grandeur utilisable, telle qu'une **tension électrique**, une **hauteur de mercure**, un **courant électrique** ou la déviation d'une aiguille.”

Source : Wikipédia (t'inquiète)

**AREM : CDR**

**AREM**

## *1) Kesako un capteur et une chaîne d'information ?*



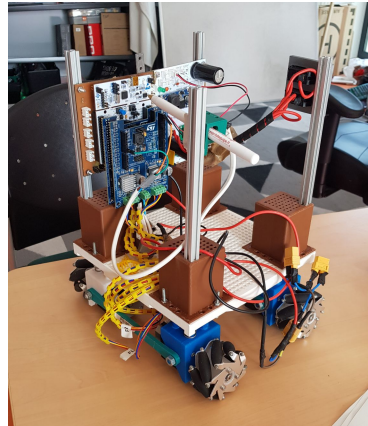
**AREM : CDR**



## ***1) Kesako un capteur et une chaîne d'information ?***

Cela nous sert à donner la perception du monde au robot, sans ça il est simplement aveugle...

Exemple:



**AREM : CDR**



## ***II) Types de capteurs et leurs sorties***

***Passif*** : Pas d'alimentation propre

*Exemple : Capteur Résistifs (résistance variant en fonction d'un paramètre donné).*

***Actif*** : Alimentation Propre

*Exemple : Capteurs Ultrasons*

**AREM : CDR**



## ***II) Types de capteurs et leurs sorties***

***Passif*** : Pas d'alimentation propre

*Exemple : Capteur Résistifs (résistance variant en fonction d'un paramètre donné).*

***Actif*** : Alimentation Propre <- Majorité de nos capteurs

*Exemple : Capteurs Ultrasons*

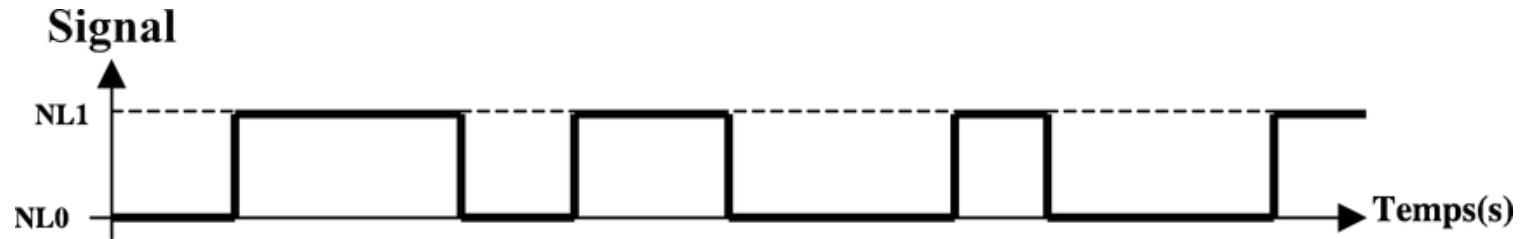
**AREM : CDR**

**AREM**



## ***II) Types de capteurs et leurs sorties***

Types de sorties : ***TOR***



**AREM : CDR**





## ***II) Types de capteurs et leurs sorties***

Types de sorties : ***TOR***

How to deal with ? : Entrée de base du microcontrôleur avec la fonction digitalRead() d'Arduino

**AREM : CDR**



## ***II) Types de capteurs et leurs sorties***

Types de sorties : ***Temps de réponse***

How to deal with ? : Vous attendez avec des delay()

(processus bloquant !)

Il existe des méthodes plus malignes en utilisant les interruptions du microcontrôleur (méthode plus complexe).

**AREM : CDR**

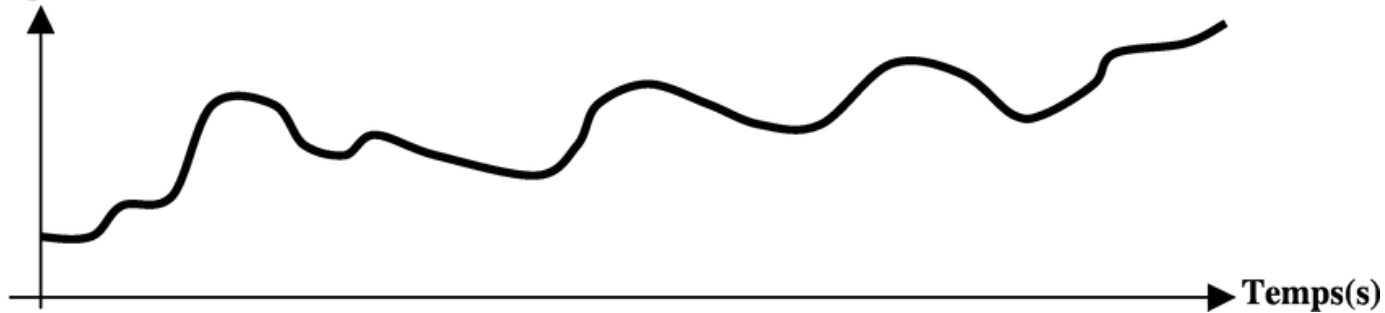
**AREM**



## ***II) Types de capteurs et leurs sorties***

Types de sorties : ***Analogique***

**Signal**



**AREM : CDR**



## ***II) Types de capteurs et leurs sorties***

Types de sorties : ***Analogique***

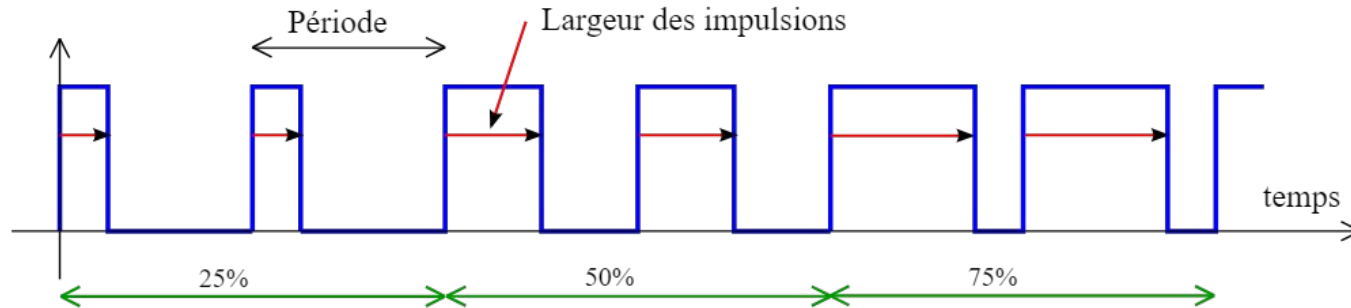
How to deal with ? : Convertisseur Analogique Numérique du  $\mu$ C grâce à la fonction Arduino “analogRead()”

**Warning** : Votre PIN d’entrée doit être connecté à un ADC (Convertisseur Analogique Numérique)

**AREM : CDR**

## ***II) Types de capteurs et leurs sorties***

Types de sorties : **PWM** (Pulse Width Modulation)



**AREM : CDR**



## ***II) Types de capteurs et leurs sorties***

Types de sorties : **PWM** (Pulse Width Modulation)

How to deal with ? : Convertisseur Analogique Numérique du  $\mu$ C grâce à la fonction Arduino “analogRead()”... again

**Warning** : Votre PIN d'entrée doit être connecté à un ADC (Convertisseur Analogique Numérique)... encore et encore

**AREM : CDR**



## ***II) Types de capteurs et leurs sorties***

Types de sorties : Plein de protocoles de communications...

How to deal with ? : En général trouver une bibliothèque... mais on va donner des exemples !

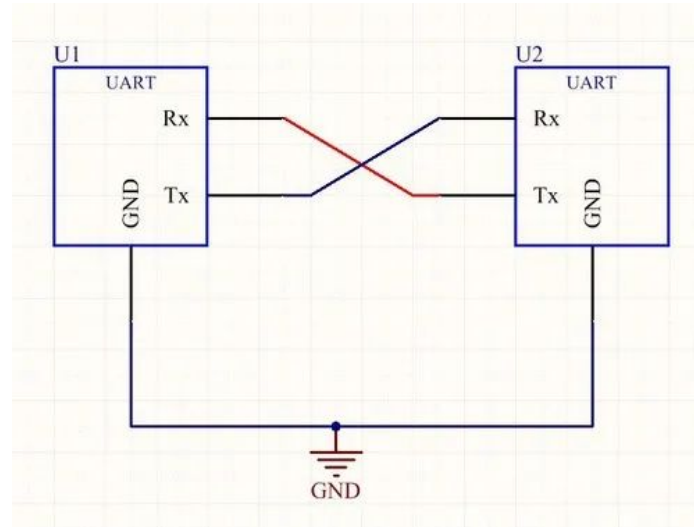
**AREM : CDR**



## III) Protocoles de communications classiques

### UART :

Différence  
entre full-duplex  
et half-duplex





**AREM**



### ***III) Protocoles de communications classiques***

***UART :***

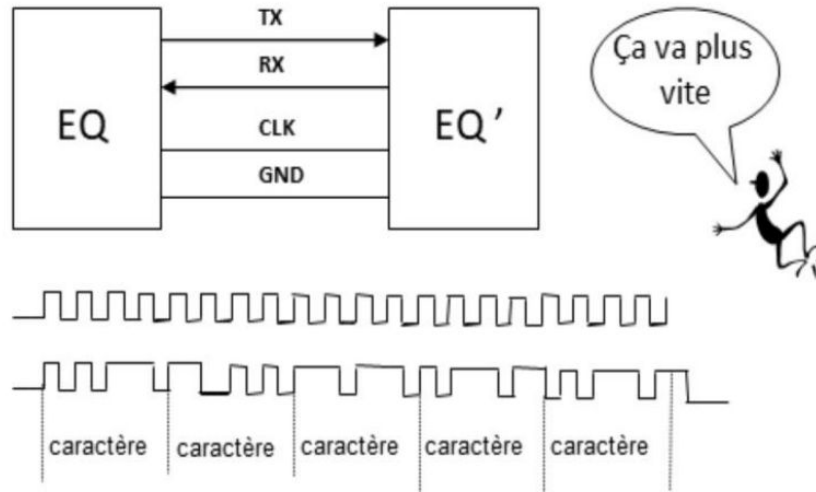


**AREM : CDR**

**AREM**

### *III) Protocoles de communications classiques*

*USART :*



merci Marques

**AREM : CDR**



### ***III) Protocoles de communications classiques***

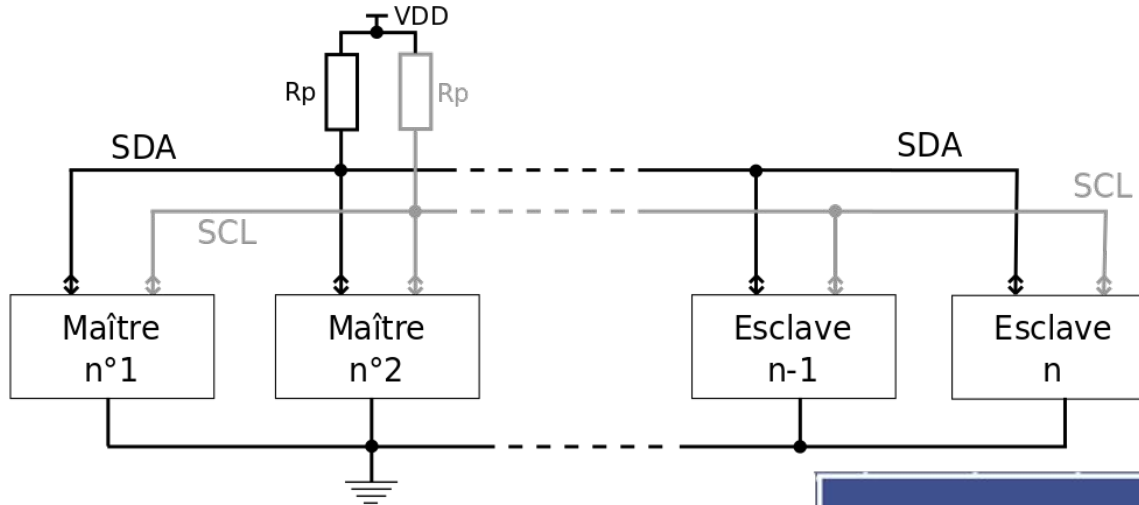
***UART et USART :***

*Comment on fait mon capitaine ?* : Bibliothèque <Serial.h> native sur Arduino.

**AREM : CDR**

### III) Protocoles de communications classiques

**I<sup>2</sup>C :**





### ***III) Protocoles de communications classiques***

***I<sup>2</sup>C :***

Attention ! : Protocole esclave maître

- Un esclave ne peut communiquer avec un autre esclave !
- Idem pour les maîtres !

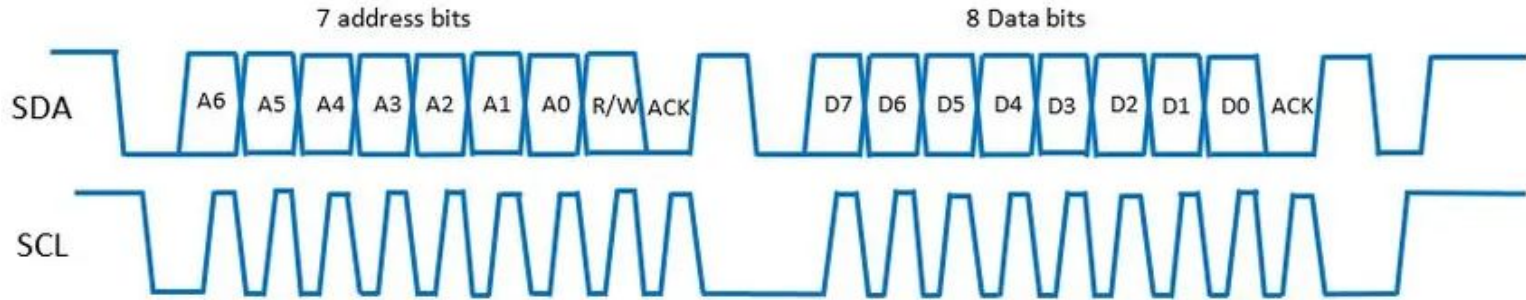
**AREM : CDR**

**AREM**



### ***III) Protocoles de communications classiques***

***I<sup>2</sup>C :***



**AREM : CDR**



### ***III) Protocoles de communications classiques***

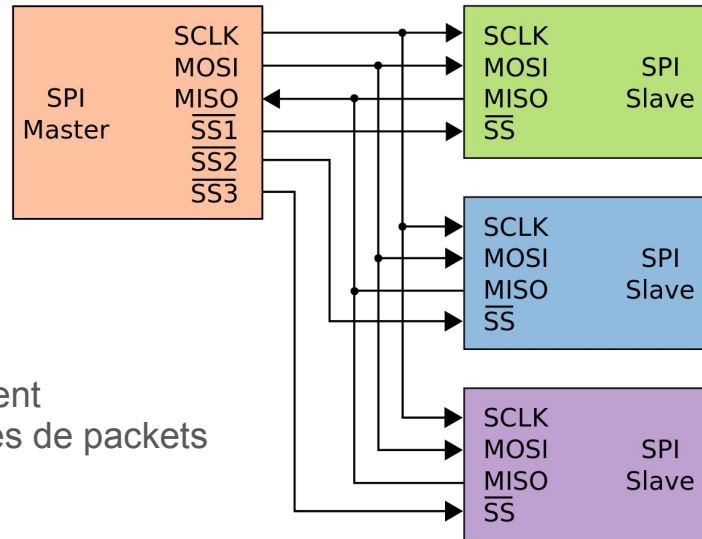
***I<sup>2</sup>C :***

*Comment on fait mon capitaine ? : Bibliothèque <Wire.h> native sur Arduino.*

**AREM : CDR**

## III) Protocoles de communications classiques

**SPI :**



- Maître/Esclave
- Full Duplex
- Pas Acquiescement
- Choix des tailles de packets





### ***III) Protocoles de communications classiques***

***SPI*** : en gros l'I<sup>2</sup>C mais en rapide



**AREM : CDR**



### ***III) Protocoles de communications classiques***

***SPI :***

*Comment on fait mon capitaine ?* : Bibliothèque <SPI.h> native sur Arduino.

**AREM : CDR**



### ***III) Protocoles de communications classiques***

Protocole :	U(S)ART	I <sup>2</sup> C	SPI
Avantage :	<ul style="list-style-type: none"><li>- Rapide</li><li>- Simple</li><li>- Bi Directionnel</li></ul>	<ul style="list-style-type: none"><li>- Simple d'utilisation</li><li>- Commun</li></ul>	<ul style="list-style-type: none"><li>- Rapide</li><li>- Modulaire</li></ul>
Inconvénient :	<ul style="list-style-type: none"><li>- Uniquement point à point</li></ul>	<ul style="list-style-type: none"><li>- Lent</li></ul>	<ul style="list-style-type: none"><li>- Plus de câbles que l'I<sup>2</sup>C</li></ul>

Point à Point / Série

**AREM : CDR**



## ***IV) Les capteurs du robot de l'année dernière***

### ***Robot :***

- Capteurs fin de course -> *Fonction digitalRead()*
- Capteurs Ultrasons -> *Bibliothèque <Ultrasonic.h>*
- Capteurs à effet halls -> *Fonction analogRead()*

### ***Le panier :***

- Capteur de force -> *Fonction analogRead()*

**AREM : CDR**



## ***IV) Les capteurs du robot de l'année dernière***

### ***Robot :***

- Capteurs fin de course -> *Fonction digitalRead()*
- Capteurs Ultrasons -> ***Bibliothèque <Ultrasonic.h> <- Librairie spéciale !***
- Capteurs à effet halls -> *Fonction analogRead()*

### ***Le panier :***

- Capteur de force -> *Fonction analogRead()*

**AREM : CDR**



## *La règle maîtresse*

Pour chaque chose que vous avez à faire, faites les recherches de la *datasheet* de votre capteur et *cherchez si des bibliothèques existent déjà* !

*Vous gagnerez du temps*



**AREM : CDR**



**AREM**

## ***La règle maîtresse***

### ***la suite***

Si vous utilisez des capteurs, **documentez les** ! Pour stocker votre documentation, utilisez le saint GitHub !

Chad mec qui documente son travail



Virgin codeur fou



**AREM : CDR**



[\*https://github.com/AREM-Proiets/documentation-capteurs\*](https://github.com/AREM-Proiets/documentation-capteurs)



**AREM : CDR**





KA-CHOW!

Go faire des robots les zamis