



ARDUINO

Introduction (1)

Farouk MEDDAH



PLAN

➤ INTRODUCTION

- Microcontrôleur, Microprocesseur
- Circuits à base de Microcontrôleur
- Arduino, Raspberry, ...

➤ ARDUINO

- Caractéristiques
- Entrées/Sorties digitales
- Entrées/Sorties analogiques
- PWM
- Communication: SPI, I2C, ...



INTRODUCTION

MICROPROCESSEUR



Un peut d'histoire

- **Intel** initialement produit des mémoires.
- En **1969** **Busicom** (ex *Nippon Calculating Machine Corporation 1967*) fabricant de calculatrices japonais, a demandé d'Intel de réaliser 12 Circuits intégrés pour ses machines.
- L'ingénieur **Ted Hoff** d'Intel proposa de concevoir un seul circuit qui englobe le tout.
- 1970: la naissance du **4004** (Le premier microprocesseur à 4 bits).
- 1971: Intel achète les droits du 4004 pour 60000\$



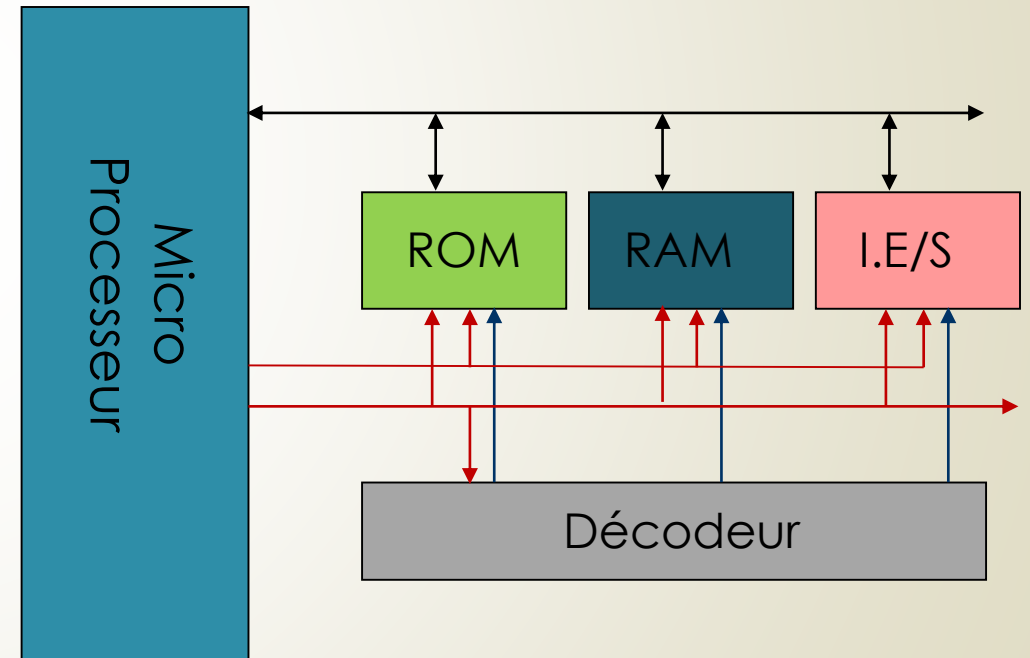
CIRCUITS A BASE DE MICROPROCESSEUR

Problème

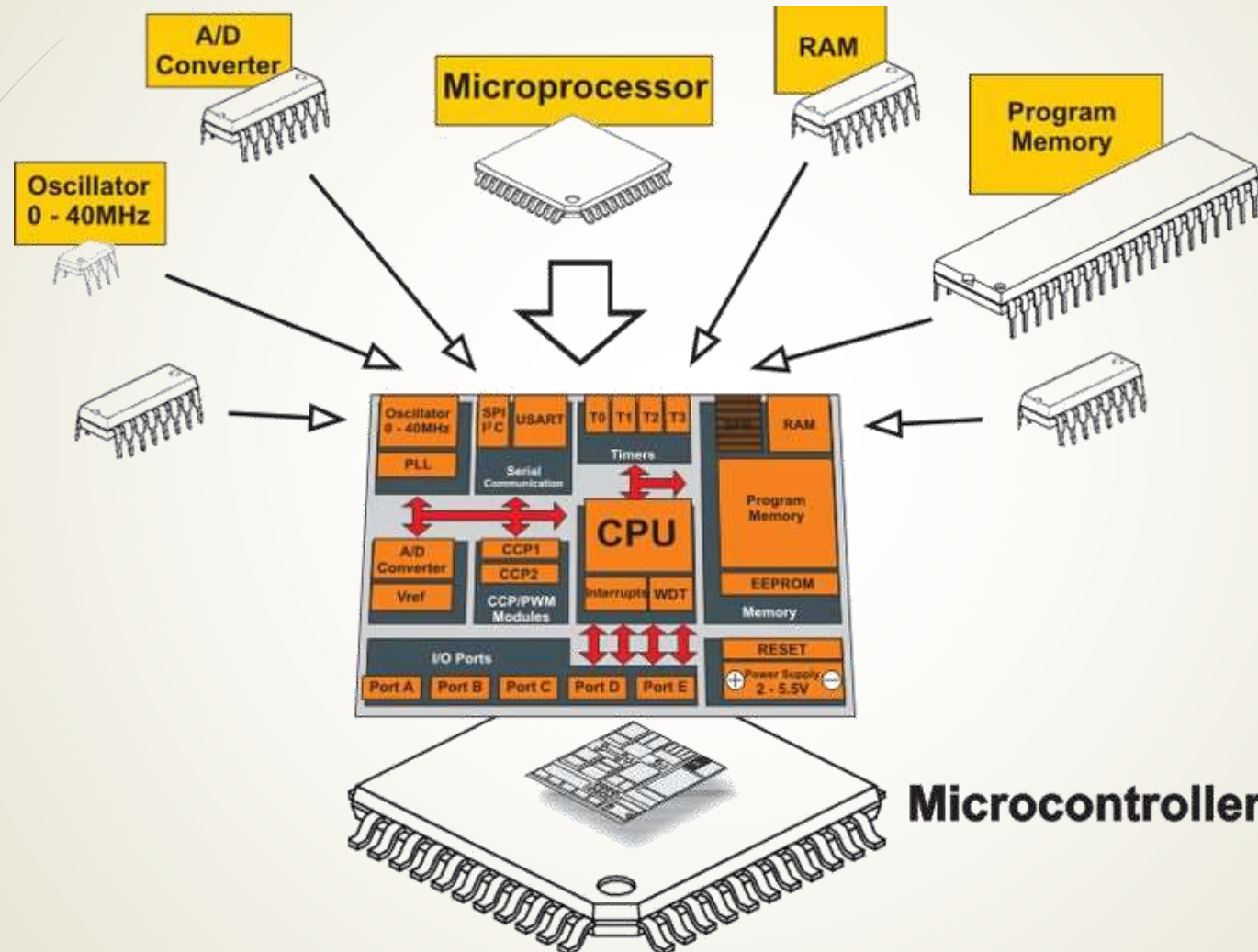
- Pour concevoir un système à base de microprocesseur il faut au moins 5 circuits:
 - RAM
 - ROM (ou Eeprom)
 - Décodeur
 - Interface (I/O)
 - Microprocesseur
 - + Autres...

Solution

- Microcontrôleur.



MICROCONTROLEUR



ROM(Eprom), RAM, ADC, DAC, Oscillateur(s), UART, Timer(s),
Décodeurs, ...

COMPARAISON

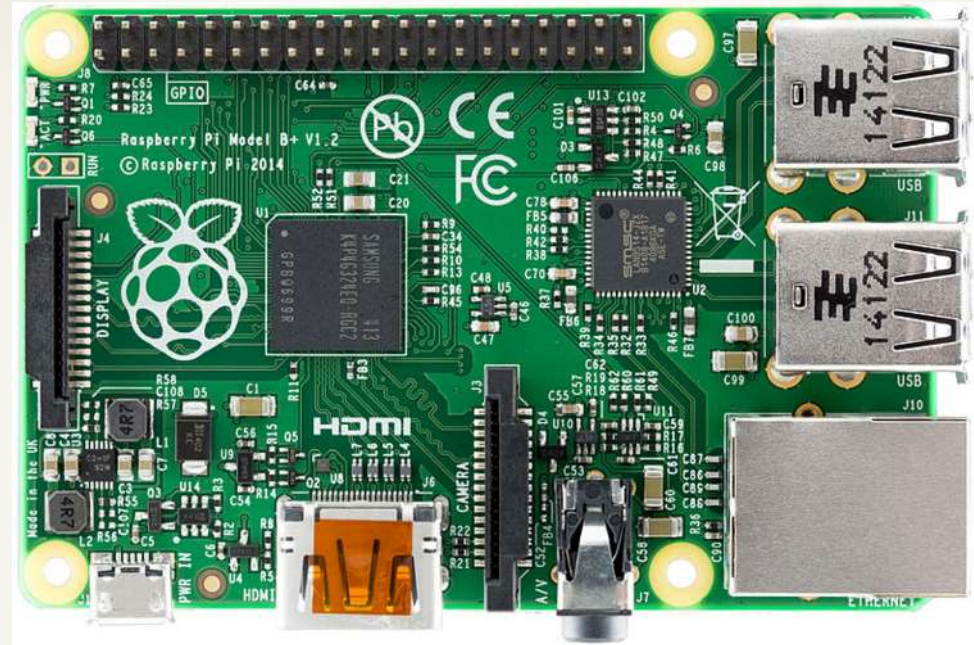
Microprocesseur

- Multi-utilisation
 - PC
 - MAC
- Mémoire externe
- Extra circuits à l'extérieur (interfaces)
- Généralement contient un système OS (Windows/Linux).

Microcontrôleur

- Utilisation unique
 - Four Micro-onde
 - Commande TV
- Mémoire à l'intérieur
- Périphériques à l'intérieur (vers l'externe)
- Généralement ne contient pas un système OS.

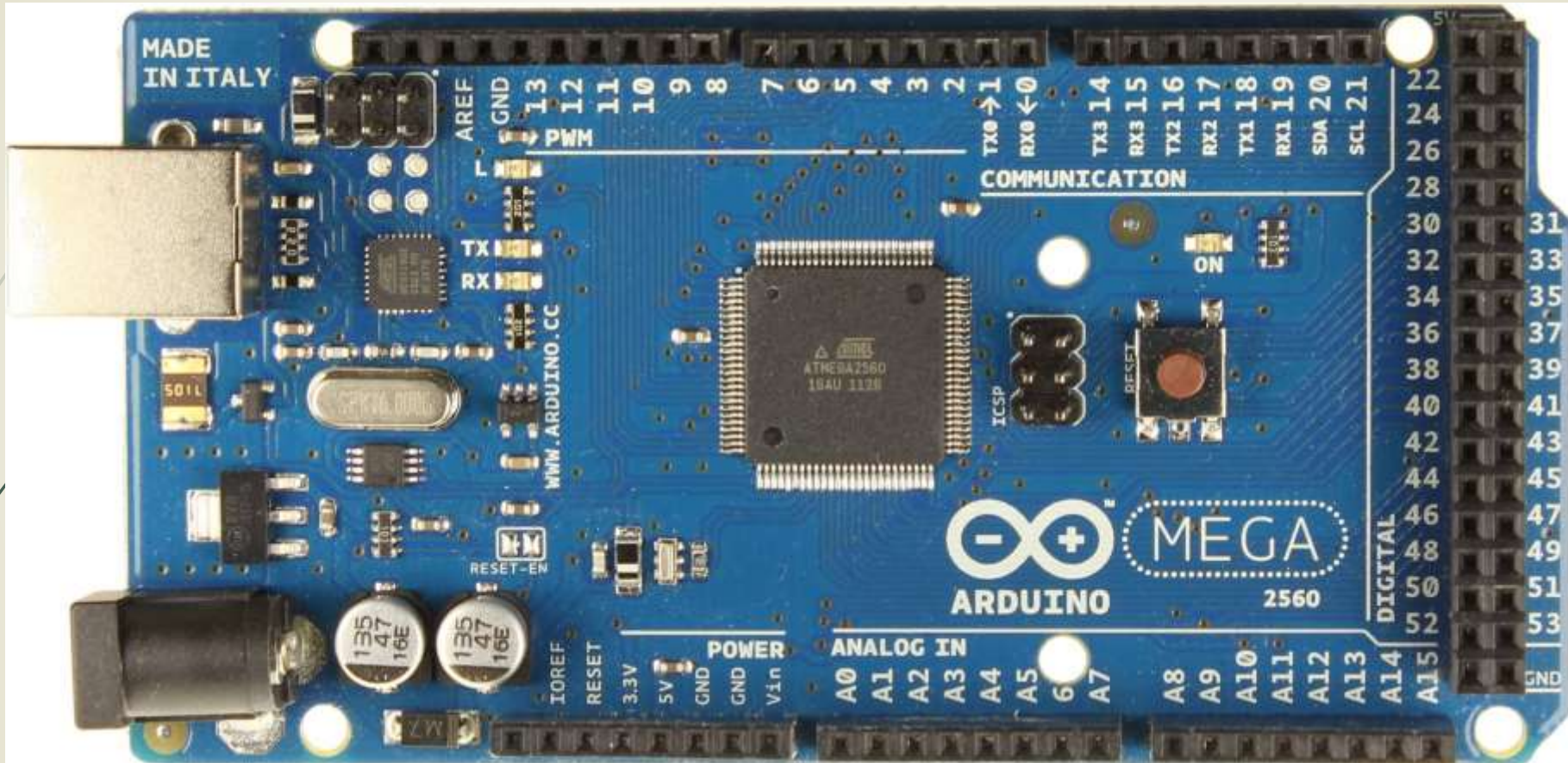
CARTES D'INTERFACES





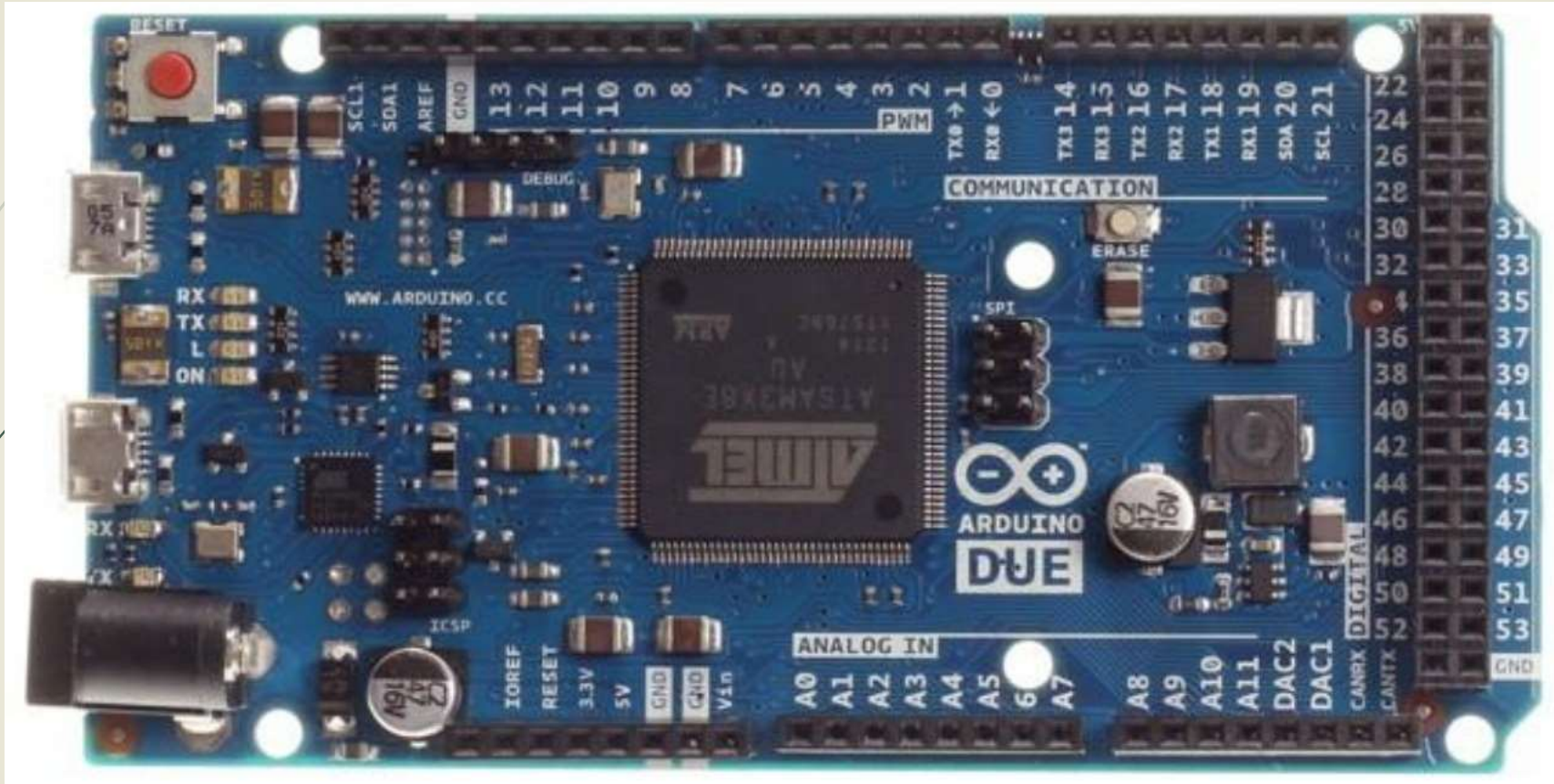
ARDUINO

ARDUINO MEGA 2560



atMega 2560, 16MHz, **4** UART
54 digitals (**15** PWM) + **16** analogs
256Kb + 8Kb + 4Kb

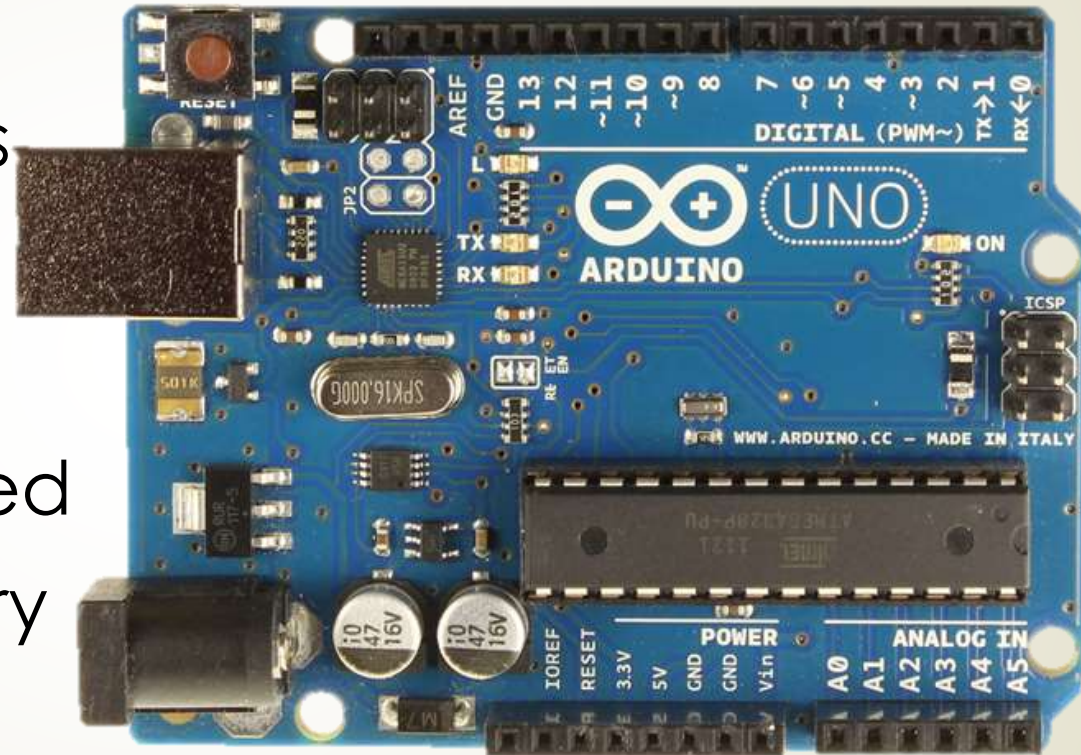
ARDUINO DUE



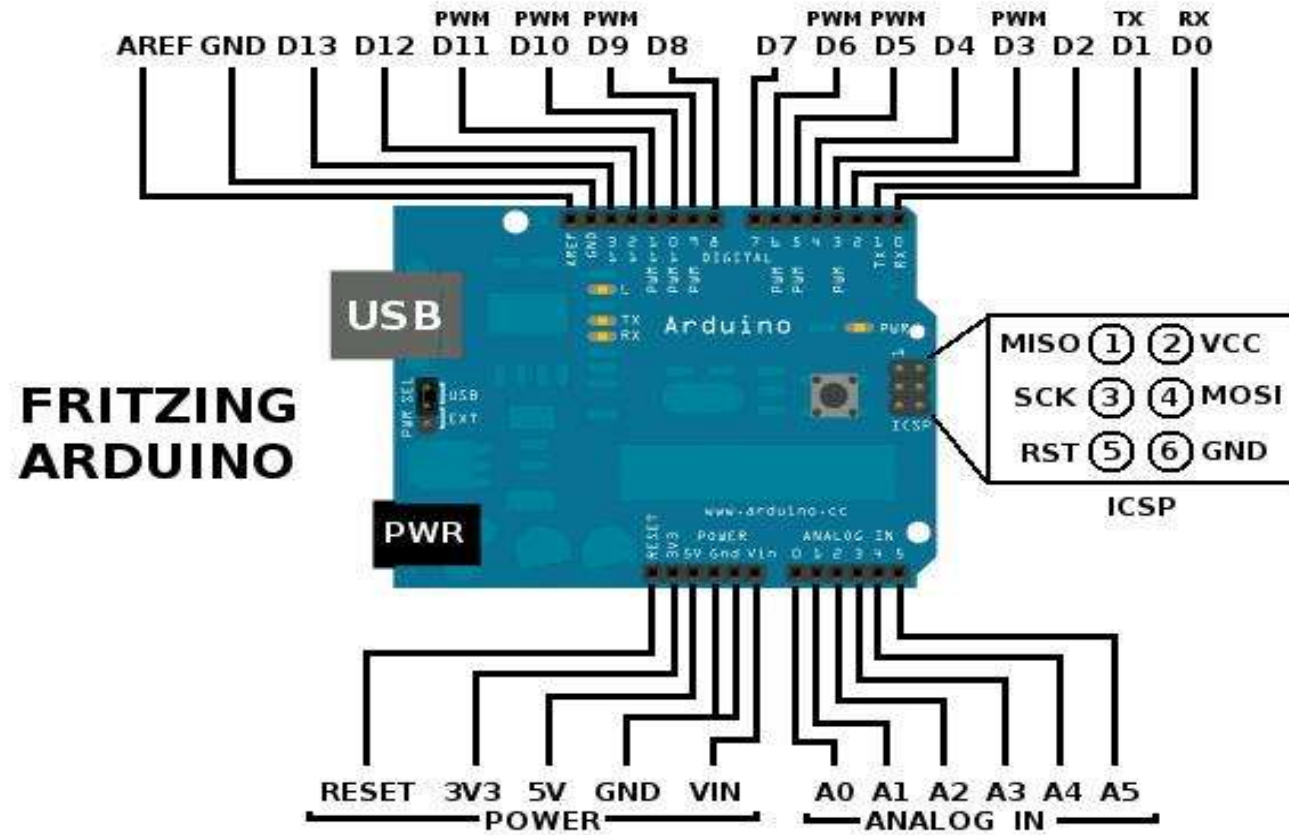
ARM SAM3X, 84MHz, DMA, **4** UART
54 digitals (**8** PWM) + **12** analogs + **2** DAC
FLASH: 512Kb + RAM: 96Kb

CARACTERISTIQUES (1)

- **14** Digital I/O pins
- **6** Analogue inputs
- **6** PWM pins
- USB serial
- **16MHz** Clock speed
- **32KB** Flash memory
- **2KB** SRAM
- **1KB** EEPROM



CARACTERISTIQUES (2)





ENTREES/SORTIES DIGITALES

REMARQUES

Il faut faire attention aux remarques suivantes:

- **Il faut** toujours indiquer le type des pins
Entrée ou Sortie avant de les utiliser.

[procédure `setup()`]

`pinMode(N°pin, mode);`

`pinMode(13, OUTPUT);`

- Mettre les pins non utilisés en entrée [Tous les pins sont mis en entrée lors du démarrage (pour des raisons de sécurité)].

ENTREES/SORTIES DIGITALES (1)

Un signal Digital ne peut avoir que 2 états

ON

- "1"
- True
- High
- 5V au pins des I/O



OFF

- "0"
- False
- Low
- 0V au pins des I/O



ENTREES/SORTIES DIGITALES (2)

Programmation des pins Digitaux

digitalWrite(Pin, Valeur);

➡ **digitalWrite(3, HIGH); //Set Output #3 High**

Emettre 5V depuis le pin 3.

➡ **digitalWrite(7, LOW); //Set Output #7 Low**

Le pin 7 sera mise à 0V.

ENTREES/SORTIES DIGITALES (3)

Programmation des pins Digitaux

Valeur = digitalRead(Pin);

- ➡ **Valeur = digitalRead(3); //Read Pin #3**
Lecture depuis le pin 3.

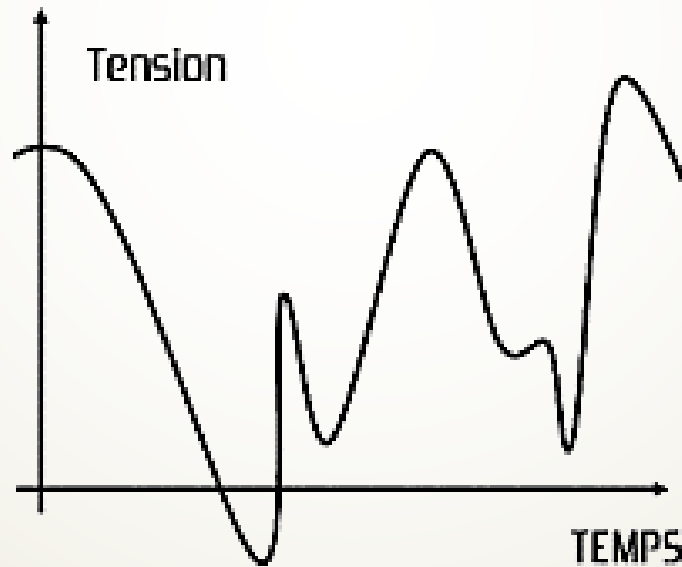


ENTREES/SORTIES ANALOGIQUES

ENTREES/SORTIES ANALOGIQUES (1)

Un signal Analogique varie dans des limites fixes.

► Peut avoir n'importe quelle valeur dans un intervalle. Ex $[0..5V]$



ENTREES/SORTIES ANALOGIQUES (2)

Programmation des pins Analogiques

Valeur = analogRead(Pin);

Le résultat: $0 \leq \text{Valeur} \leq 1023$

➡ **Valeur = analogRead(1); //Read Pin #1**

Lecture depuis le pin 1.

ENTREES/SORTIES ANALOGIQUES (3)

Programmation des pins Analogiques

`analogWrite(Pin, Valeur);`

$0 \leq \text{Valeur} \leq 255$

► `analogWrite(2, 230);`

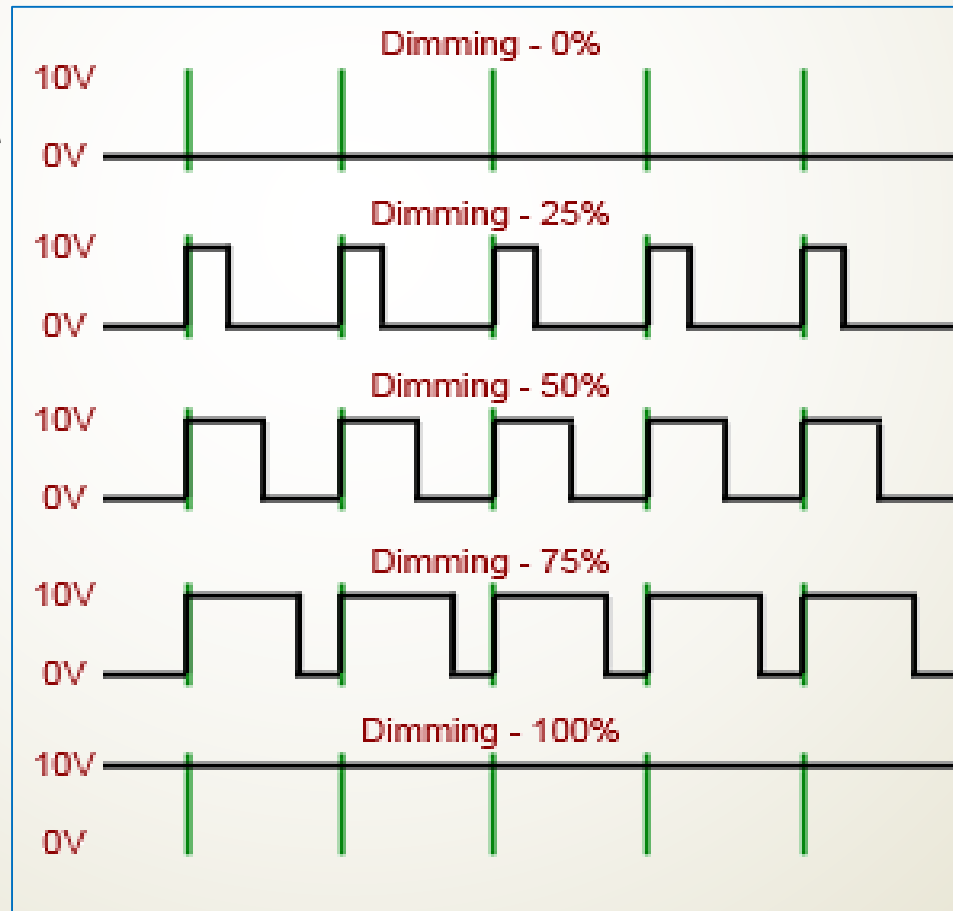
Emettre 4.5V ($5 \cdot 230 / 255$) depuis le pin 2.



PWM

PULSE WIDTH MODULATION

- La modulation de largeur d'impulsions
 - Vitesse d'un moteur DC
 - LED luminosité
 - Potentiomètre
 - ...

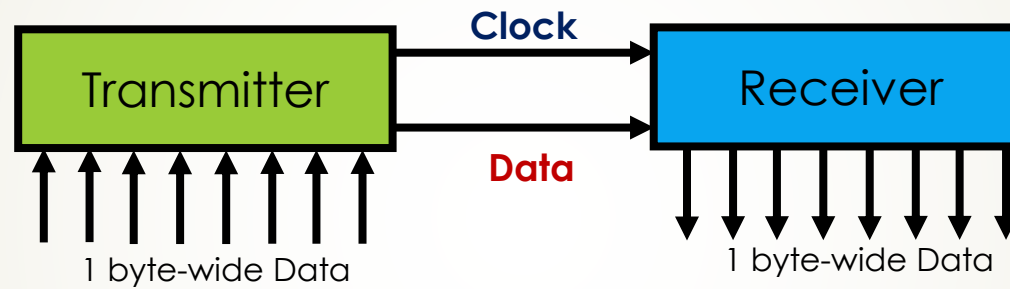




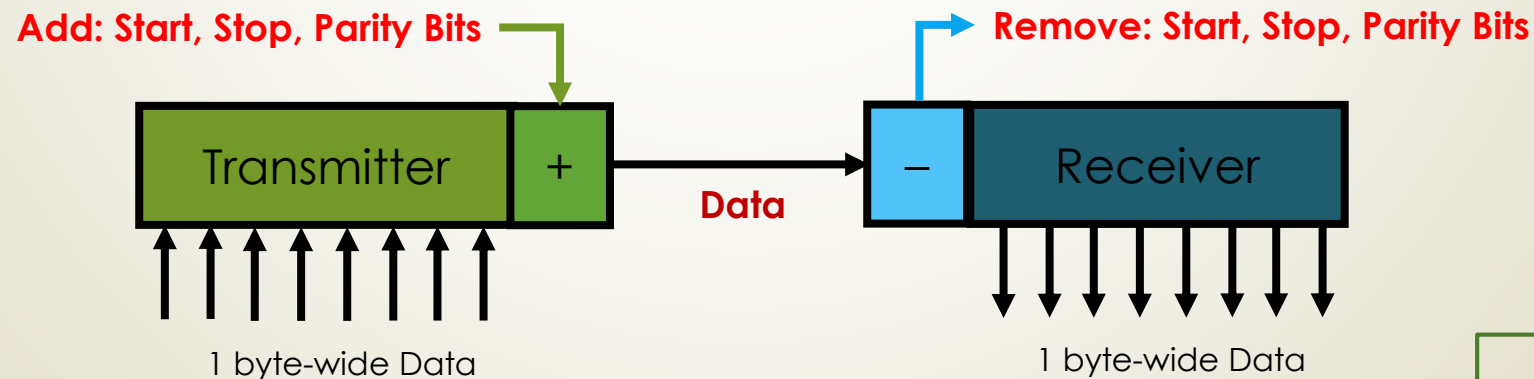
COMMUNICATION

COMMUNICATION (1)

- **Synchrone:** L'émetteur et le récepteur partagent une horloge commune.

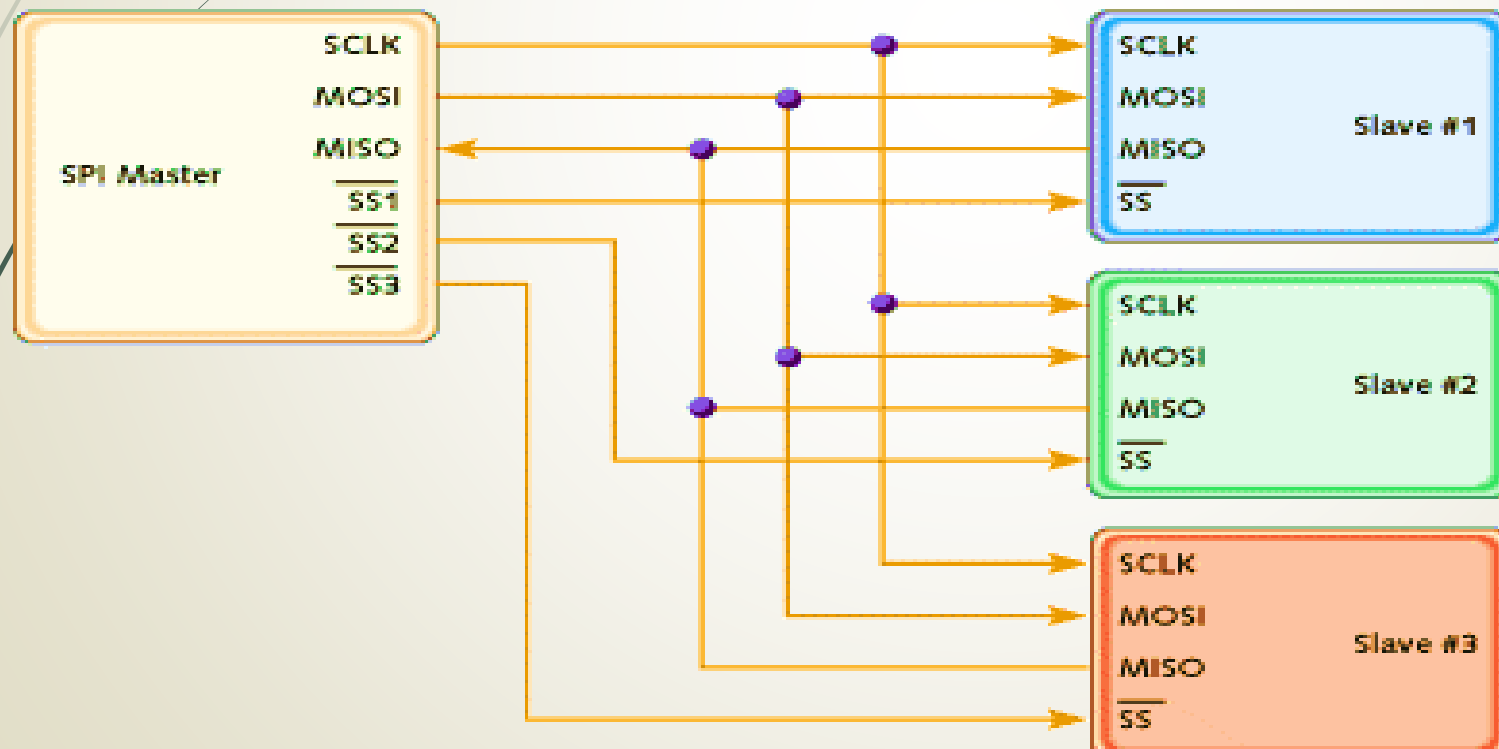


- **Asynchrone:** L'émetteur et le récepteur ne partagent aucune horloge commune.



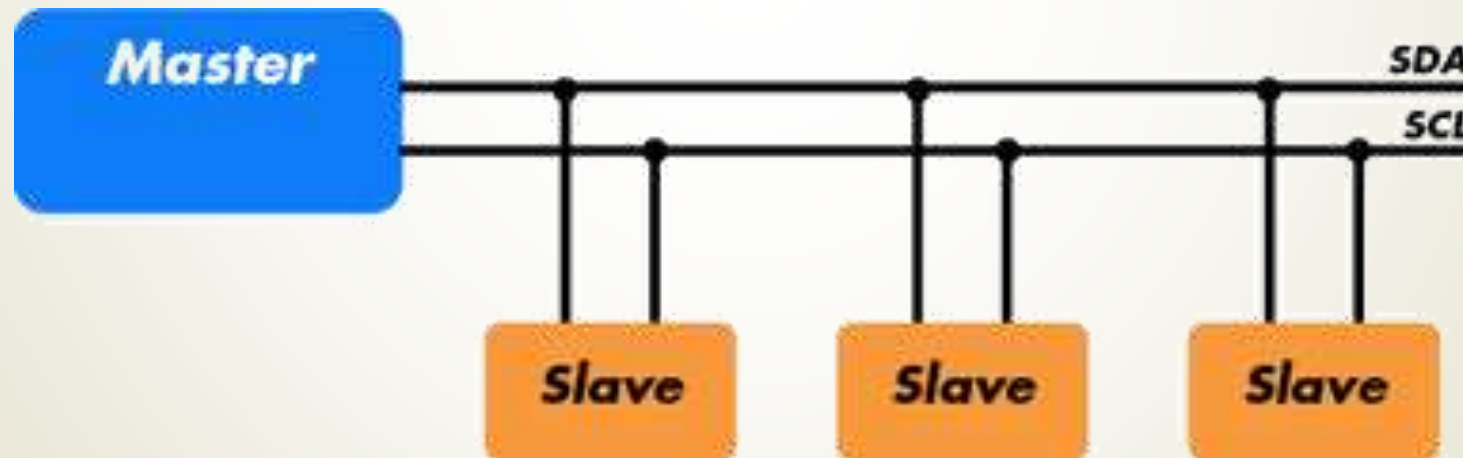
COMMUNICATION (3)

- **SPI:** (**S**erial **P**eripheral **I**nterface) Bus de données série synchrone [Motorola (Freescale)].
- Un maître et plusieurs esclaves.
- La communication est contrôlée par le maître.



COMMUNICATION (4)

- **I2C:** (Inter-Integrated Circuit) Nécessite seulement deux fils. [Philips 1982]. Appelé par fois **TWI** (Two Wire Interface).
- **SCL:** Serial Clock
- **SDA:** Serial Data





CAPTEURS ET SHIELDS

Il existe un
sous fo



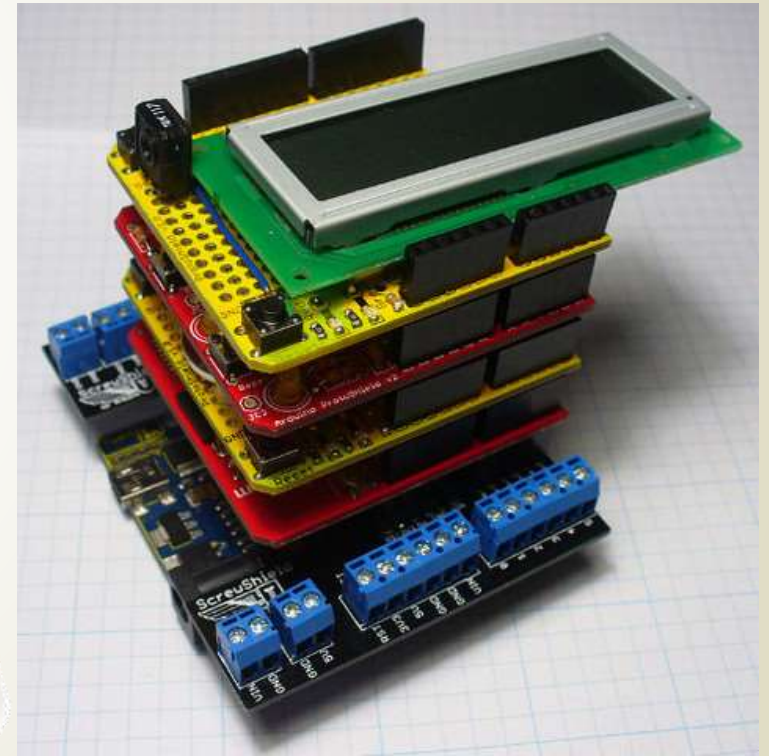
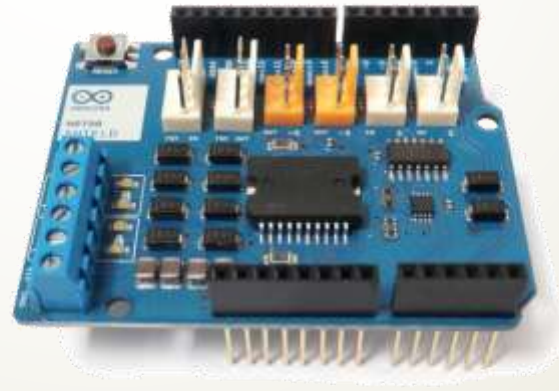
SHIELDS

Les (SHIELDS) sont des cartes d'extensions peuvent être placer au dessus de l'Arduino.

Exemples:

- Cartes réseaux
- GPS
- Écrans
- ...

<http://shieldlist.org/>





MERCI