

Arduino ou la démocratisation des microcontrôleurs

Julien VILLEMEJANE | Professeur Agrégé



Institut d'Optique Graduate School / Palaiseau
Laboratoire d'Enseignement Expérimental
Co-responsable des Travaux Pratiques
Electronique / Informatique

Arduino ou la démocratisation des microcontrôleurs

- Présentation de l'Institut d'Optique
- **Informatique et systèmes embarqués**
- **Introduction à Arduino**
- **Atelier pratique**



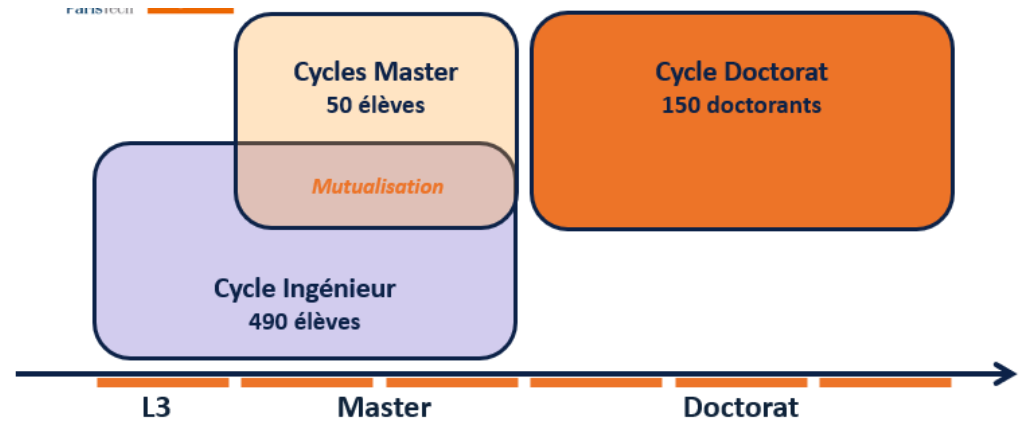
Julien VILLEMEJANE | Professeur Agrégé

Institut d'Optique Graduate School / Palaiseau
Laboratoire d'Enseignement Expérimental
Co-responsable des Travaux Pratiques
Electronique / Informatique

L'Institut d'Optique Graduate School



Graduate School en photonique



Plus de 1000 personnes pour développer la photonique

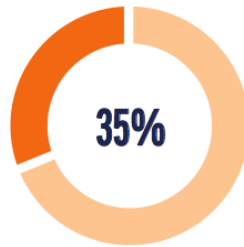
- 490 élèves en cycle Ingénieur + 25 en Master
- 150 doctorants
- 130 Chercheurs et Enseignants-chercheurs
- 100 Ingénieurs, techniciens et administratifs
- 210 personnes en innovation et entrepreneurs

Sur 3 piliers : Formation, Recherche, Innovation-Entrepreneuriat

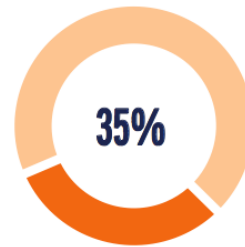
Sur 3 sites : Paris Saclay, Bordeaux, St Etienne



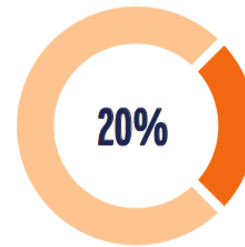
Graduate School en photonique



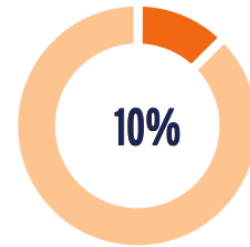
Chercheurs



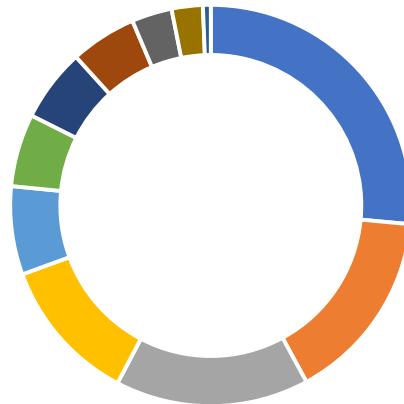
Ingénieurs



Managers



Audit, conseil, marketing



- Aérospatial - Défense
- R&D scientifique
- Conception bureaux d'étude
- Autres industries
- Techno de l'information
- Banques assurances
- Production
- Automobile
- Energie
- Chime, pharmacie, santé
- BTP

La photonique

Sciences et technologies de la lumière



La photonique

1997-2018 : prix Nobel liés à la photonique :

10 prix Nobel de physique (dont 3 français)

2 prix Nobel de chimie

1997 : Refroidissement d'atomes avec de la lumière

1999 : Lasers femtosecondes pour la chimie

2000 : Semi-conducteurs pour l'opto-électronique

2001 : Condensat de Bose-Einstein

2005 : Spectroscopie laser et peigne de fréquence

2009 : Communications par fibres optiques et capteurs CCD

2012 : Mesure et manipulation de systèmes quantiques

2014 : LED bleues

2014 : Microscopie de fluorescence à haute résolution

2017 : Détection des ondes gravitationnelles

2018 : Génération d'impulsions optiques ultra-courtes et de haute intensité

Pincés optiques et leur application aux systèmes biologiques



INTERNATIONAL
YEAR OF LIGHT
2015

La photonique

Fonctions	Aquérir de l'information	Transmettre de l'information	Délivrer de l'information	Eclairer	Apporter de l'énergie	Produire
Types de systèmes	Capteurs et imageurs	systèmes de communication	Ecrans, projecteurs...	LED, éclairage	Photo-voltaïque	Lasers
Exemples						

La photonique / Exemples



Phares intelligents



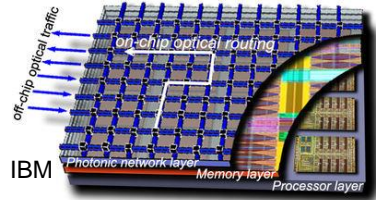
Réalité augmentée

Informatique et Systèmes embarqués

Autour de vous ! Elec / Info / Embarqué



AUDI



IBM



Studyrama - 6979

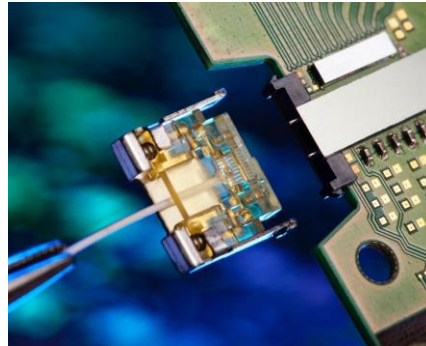


<http://www.leseclairagistesassocies.com>

“ OUVREZ LES YEUX ”



Moto 360 watch Android Wear



Drone AirSelfie



IPSR

Les prémices de l'informatique



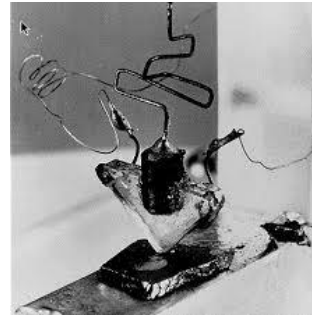
1907
Lee De Forest

ELECTRON

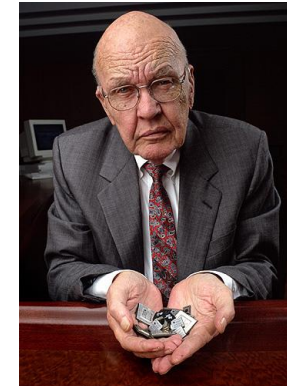


1923
Iconoscope

ELECTRONIQUE



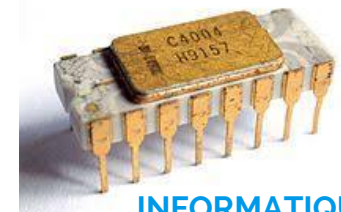
1948
Transistor



Courtesy Texas Instruments

1959
Circuits intégrés

**MICRO-
ELECTRONIQUE**



1972
Intel 4004

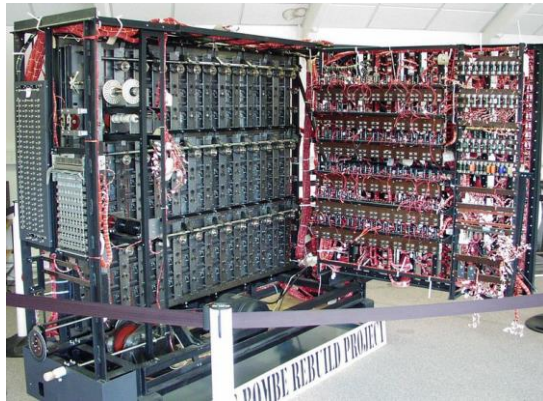
INFORMATIQUE

Informatique / Pour quoi faire ?

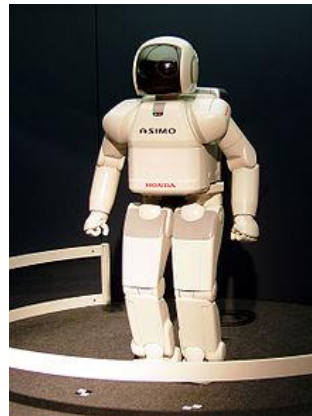


1652
La PASCALINE

- Aide aux calculs
- Traitement de l'information
- Automatisation de calculs / de tâches
- Télécommunications
- Stockage de l'information



1940
La BOMBE / Turing



2000
ASIMO



2014
GOOGLE / Datacenter



2016
GOOGLE / Datacenter

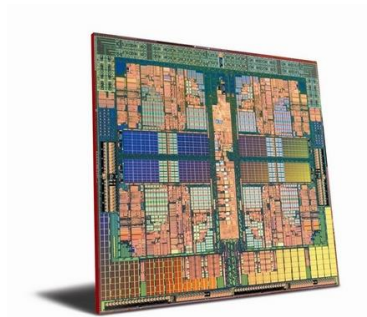
Informatique / Industrielle / Embarquée



Informatique / Comment ?

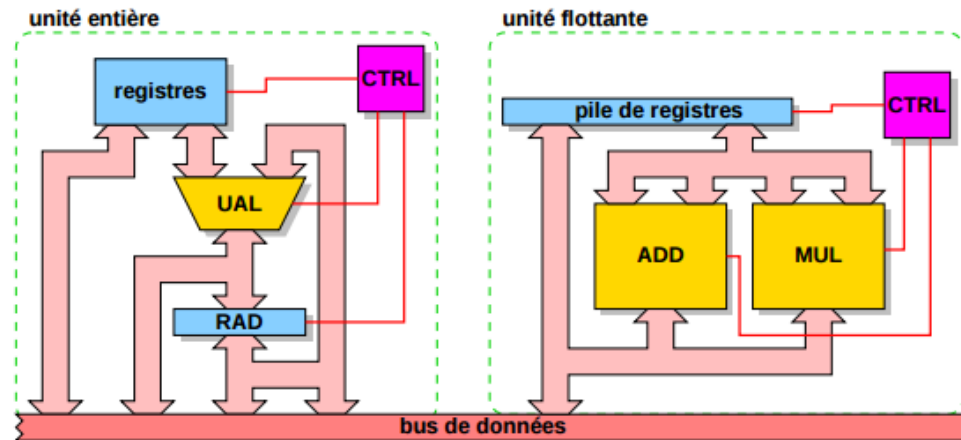
- Calculateur / Machine

PROCESSEUR



AMD Phenom – 4 coeurs – 3.4 GHz

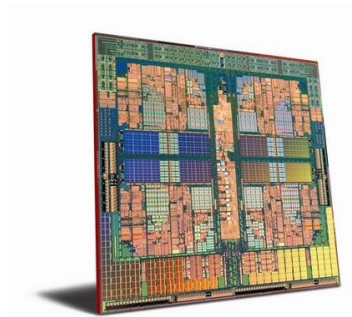
- Exécution séquentielle d'instructions
- Unité(s) de calculs précablée(s)



Informatique / Comment ?

- Calculateur / Machine

PROCESSEUR



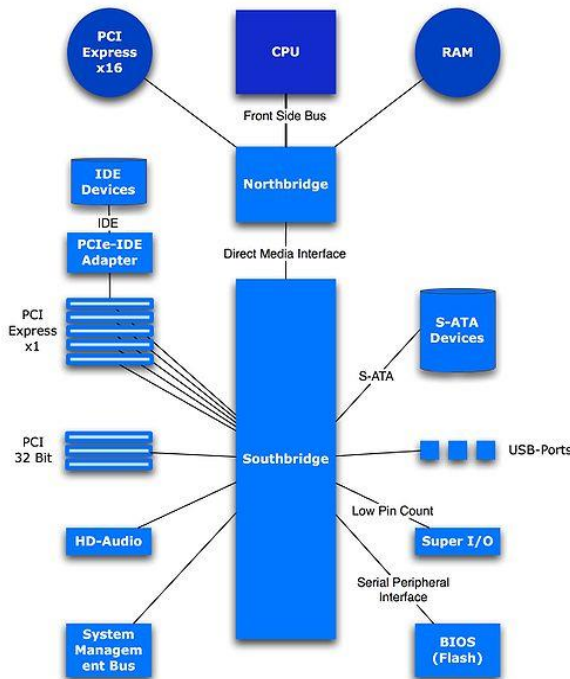
AMD Phenom – 4 coeurs – 3.4 GHz

- Exécution séquentielle d'instructions
- Unité(s) de calculs précablée(s)

MÉMOIRES



- Stockage temporaire mais rapide
- Stockage à long terme mais "lent"

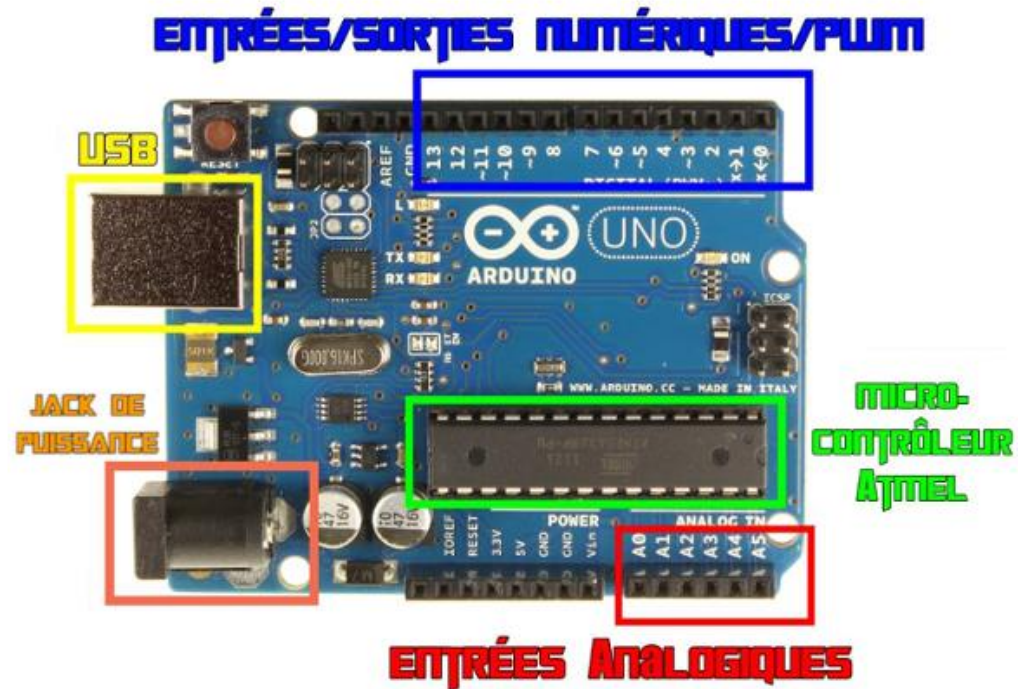


ARDUINO

Bases de programmation

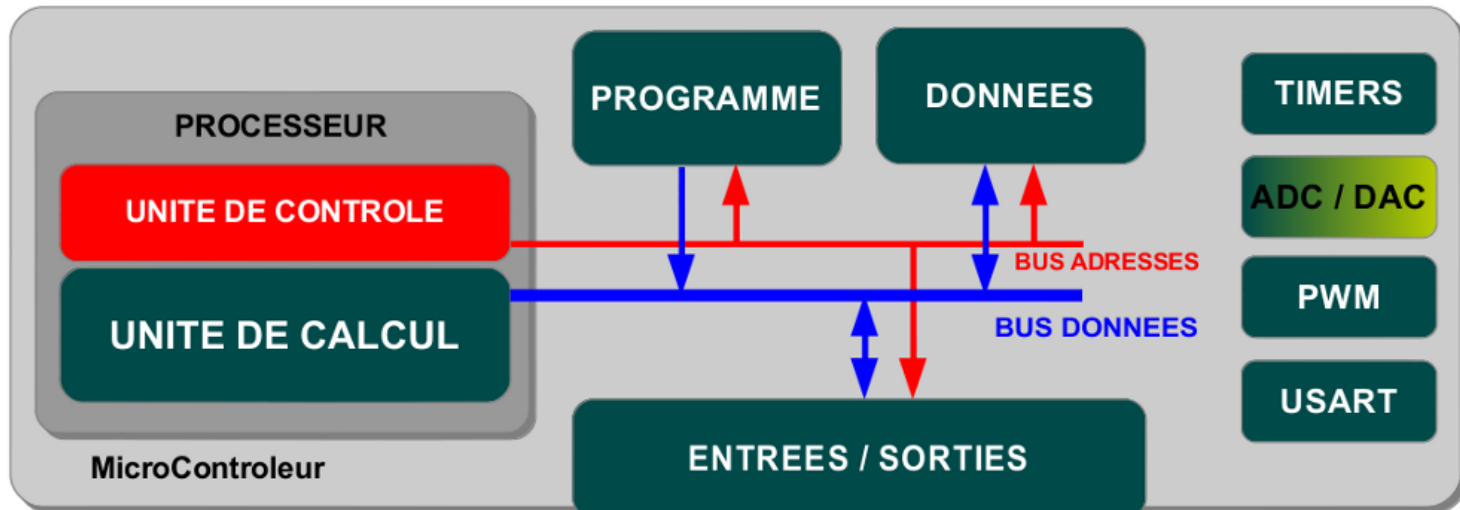
Qu'est-ce qu'une carte Arduino ?

- Microcontrôleur
- Alimentation
- Téléversement
- Entrées / Sorties



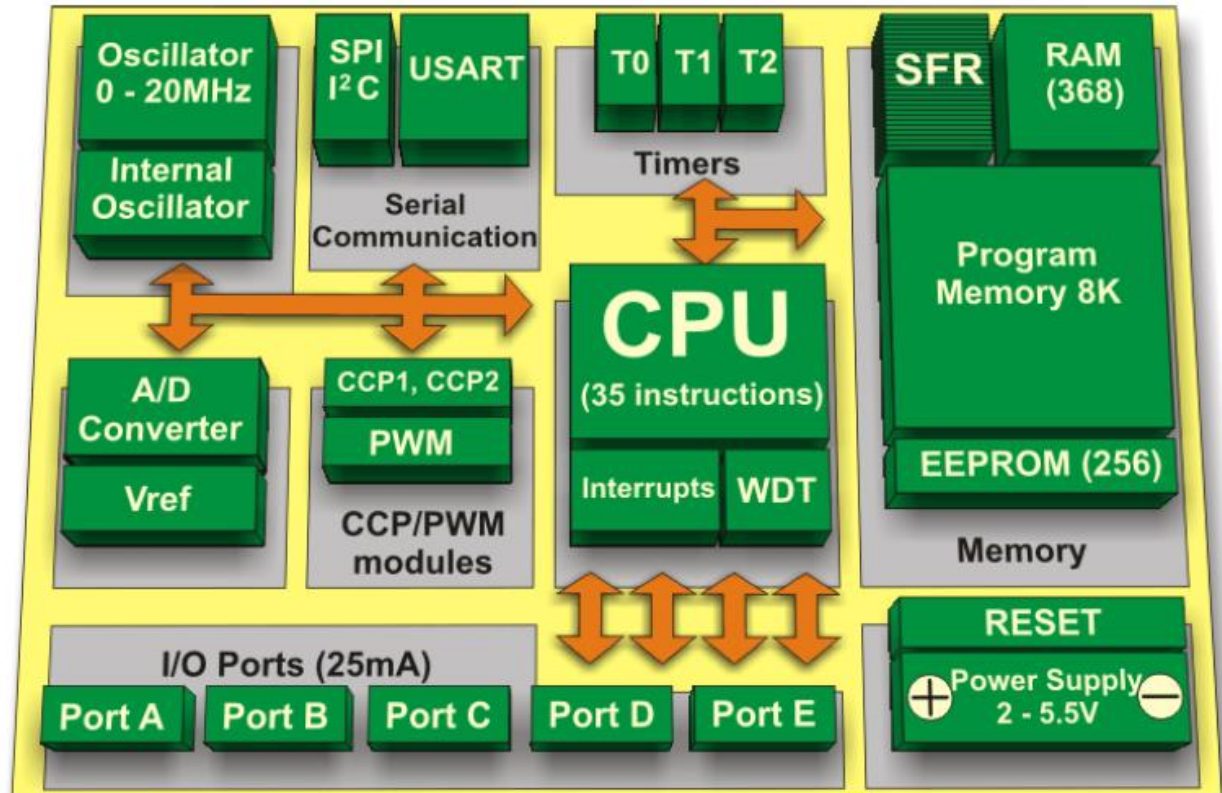
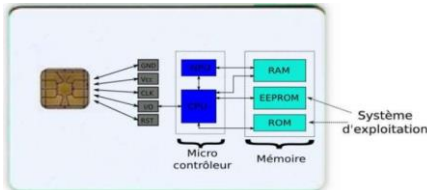
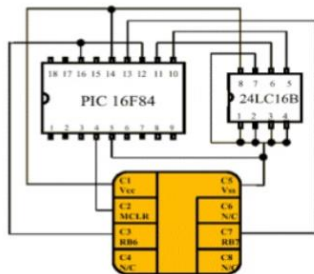
Qu'est-ce qu'un microcontrôleur ?

- Processeur spécialisé
- Entrées / Sorties réactives



Qu'est-ce qu'un microcontrôleur ?

- Processeur spécialisé
- Entrées / Sorties réactives

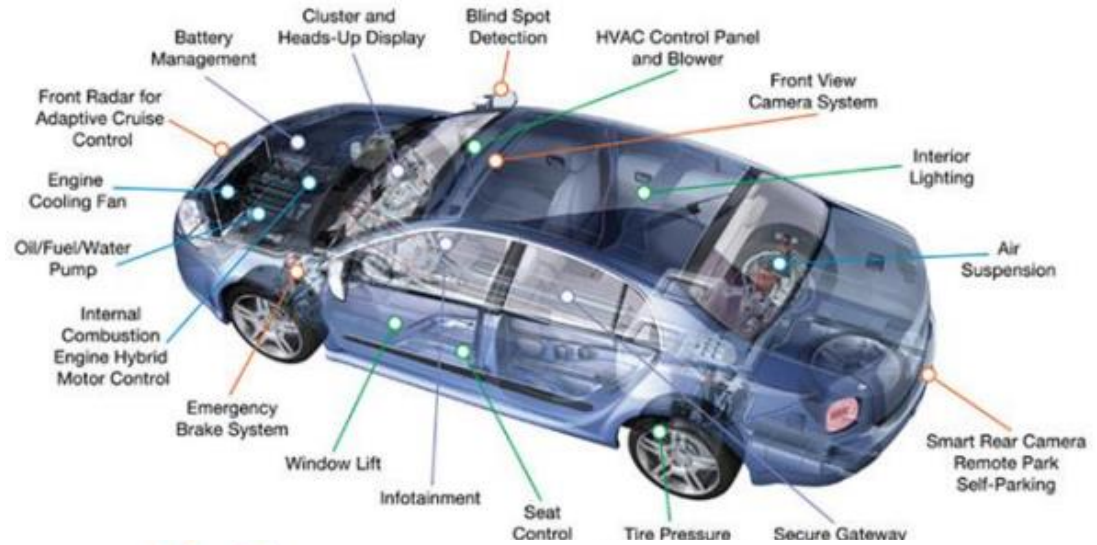


Mouna Yanguï

<http://learn.mikroe.com/> - PIC16F887

Où sont-ils utilisés ?

- Système embarqué
- Capteurs intelligents



Et par rapport à un PC ?

- Processeurs généralistes vs Processeurs embarqués

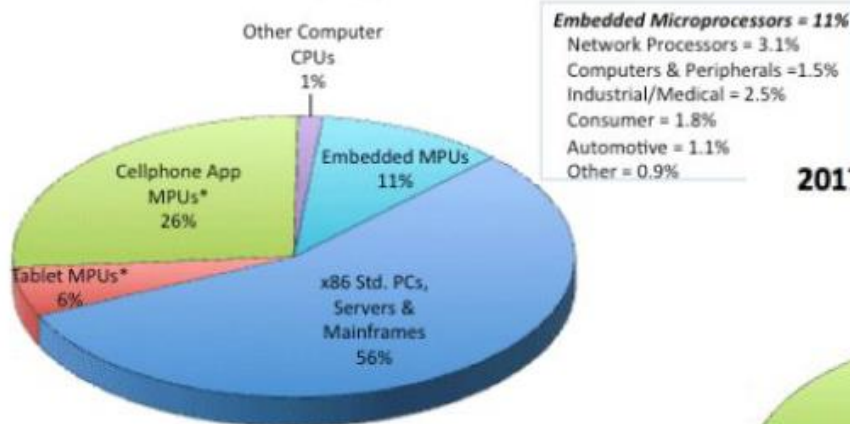
	PC standard	ARDUINO Uno
Fréquence	2 GHz	16 MHz
Core / Architecture	4 / 64 bits	1 / 8 bits
Consommation	100 à 500 W	< 1 W
Entrées/Sorties	/	6 Analog / 13 Digital
Ports extension	USB, PCI...	SPI, I2C, RS232
RAM	4 Go	2 ko



Et par rapport à un PC ?

- Processeurs généralistes vs Processeurs embarqués

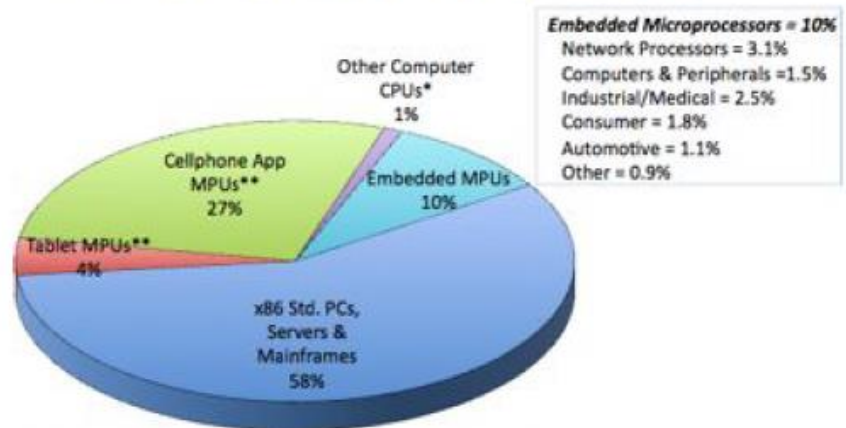
2013 MPU Sales by Applications (Fcst, \$61.0B)



*Includes ARM-based and x86 processors.

Source: IC Insights

2017 MPU Sales by Applications (Fcst, \$66.6B)



*Covers ARM and other RISC MPUs in servers and workstations.

**Includes ARM and x86 mobile application processors.

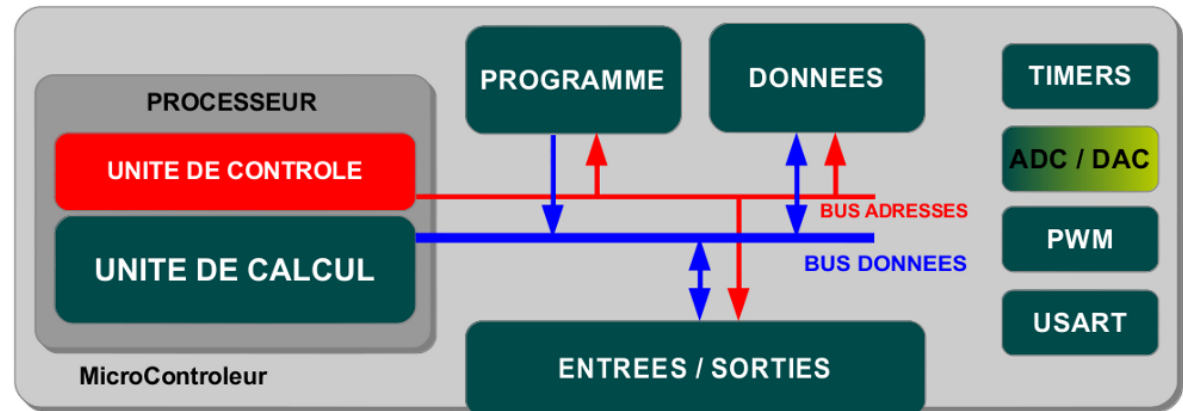
Source: IC Insights

Revenons à Arduino

- Accès à des entrées/sorties réactives



ARDUINO Uno
16 MHz
1 core / 8 bits
< 1 W
6 Analog / 13 Digital
SPI, I2C, RS232
2 ko



- Grande communauté
- Shields (extensions)

Les outils pour programmer

- Programme
- IDE = Interface de Développement
- Téléversement



```

Blink | Arduino 1.0
File Edit Sketch Tools Help

Blink
Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.
*/

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
    
```

```

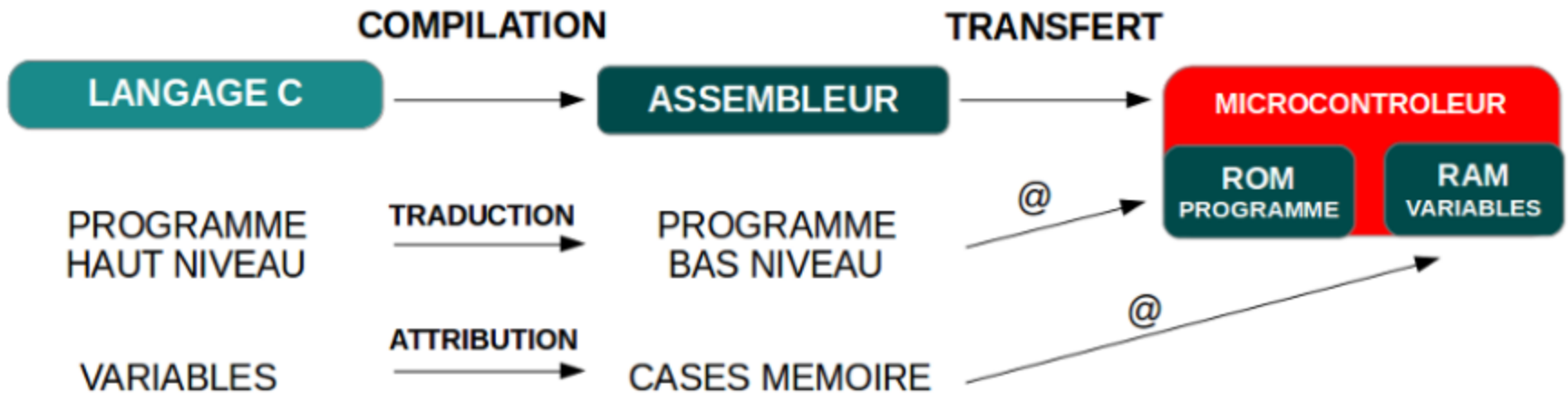
PROGRAMINO IDE FOR ARDUINO & GENUINO 1.1.0.1
File Edit Options View Sketch Hardware Web Tools Help

Project Explorer
C:\Program Files (x86)\PROGRAM
Terminal.ino

Terminal.ino new_sketch.ino
16 long cnt = 0;
17
18
19 void setup()
20 {
21   // Set baudrate to 19200 baud
22   Serial.begin(19200);
23 }
24
25
26 void loop()
27 {
28
29   // Read the analog input A0
30   Serial.print(analogRead(A0));
31   Serial.print(" ");
32   Serial.println(cnt);
33   delay(200);
34   cnt++;
35
36   // Save data after 50 measurements into the Logfile
37   if(cnt==50)
38   {
39     Serial.println("#SAVE#Logfile.txt");
40   }
41 }
    
```

Les étapes pour programmer

- Programme
- IDE = Interface de Développement
- Téléversement



Mon premier programme

- Prise en main interface
- Connexion USB
- Programme **Blink**



1

2

3

Pilotage d'une sortie externe / LED

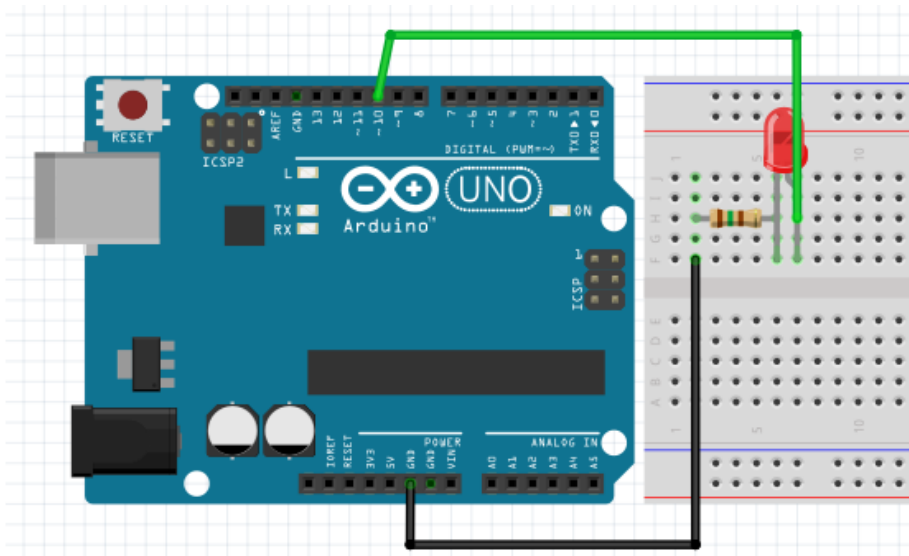
- Cablage d'une LED sur D10

$$R_{LED} = \frac{V_{ARDUINO} - V_{SEUIL}}{I_{MAX}}$$

- Programme **Blink** modifié

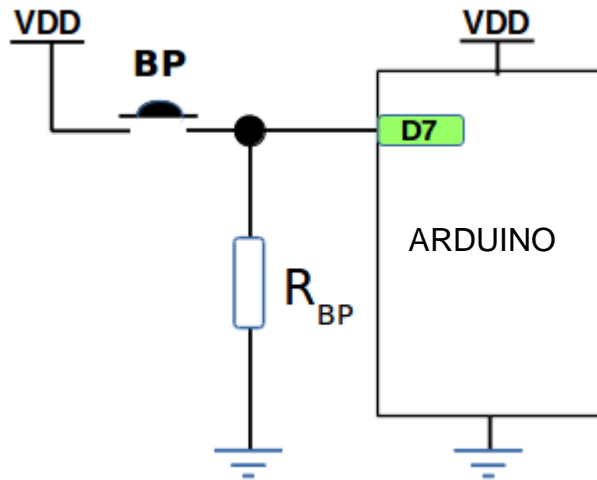
Blink §

```
void setup() {
  pinMode(10, OUTPUT);
}
void loop() {
  digitalWrite(10, HIGH);
  delay(1000);
  digitalWrite(10, LOW);
  delay(1000);
}
```



Récupération d'une donnée numérique

- RBP pour protéger l'alimentation



Blink §

```
int val = 0;

void setup() {
  pinMode(13, OUTPUT);
  pinMode(7, OUTPUT);
}

void loop() {
  val = digitalRead(7);
  digitalWrite(13, val);
}
```

Récupération d'une donnée analogique

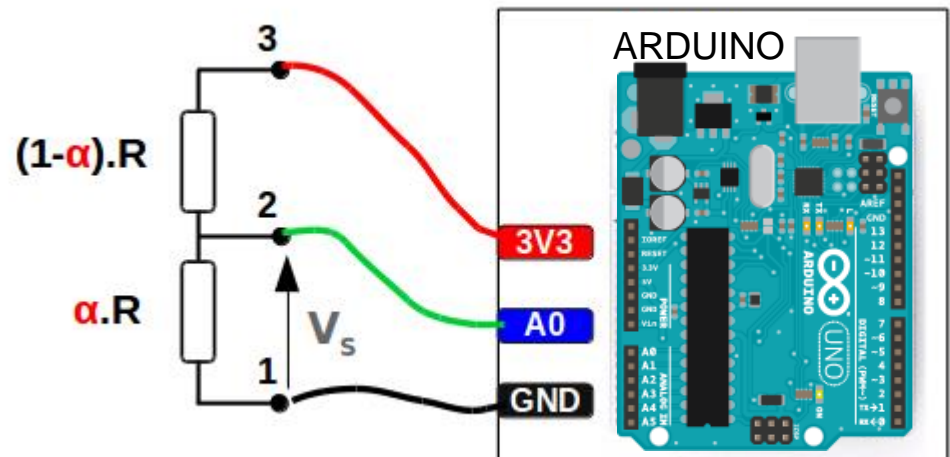
- Câblage potentiomètre

AnalogReadSerial §

```
void setup() {
}

void loop() {
  int sensorValue = analogRead(A0);

  delay(1);
}
```



- Comment vérifier que la donnée est bien convertie ?

- CAN intégré – 10 bits

Récupération d'une donnée analogique

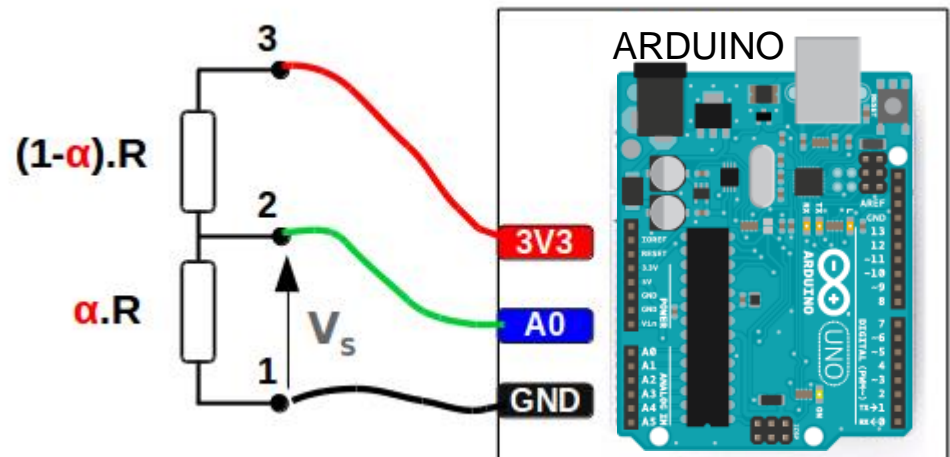
- Câblage potentiomètre
- Affichage console + Traceur / Exemple : [Basics/AnalogReadSerial](#)

AnalogReadSerial §

```

void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(1);
}
    
```



- CAN intégré – 10 bits

Récupération d'une donnée analogique

- Câblage potentiomètre
- Affichage console + Traceur / Exemple : [Basics/AnalogReadSerial](#)

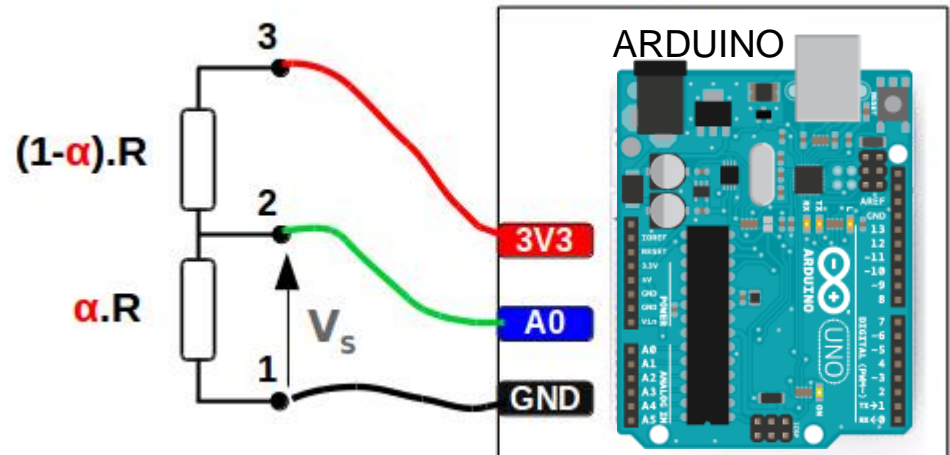
AnalogReadSerial §

```

void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(1);
}

```



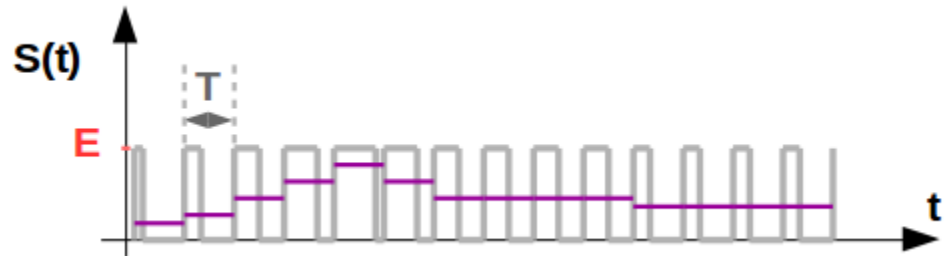
- CAN intégré – 10 bits

Contrôler la luminosité d'une LED

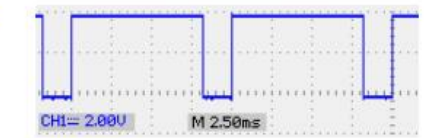
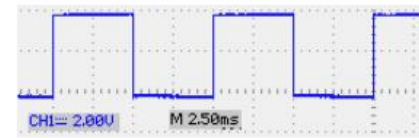
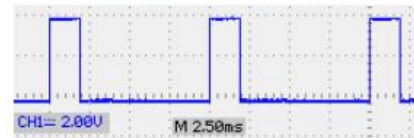
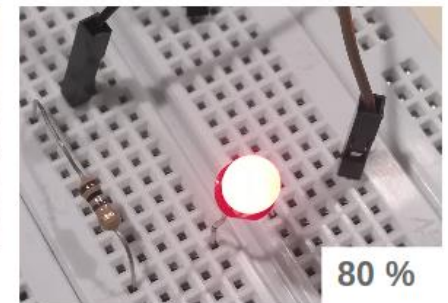
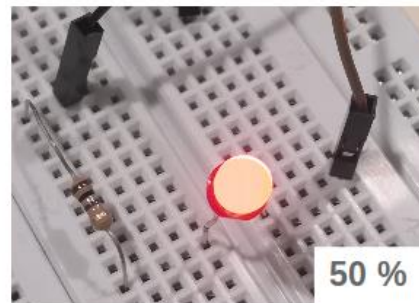
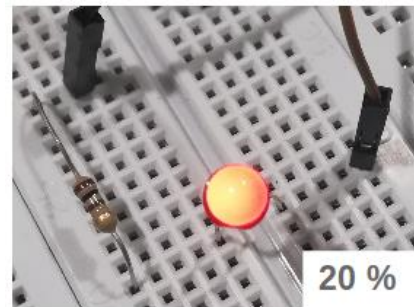
- Modulation de largeur d'impulsions
- PWM (Pulse Width Modulation)

- Signal à période fixée
- Variation du temps haut

- RC = rapport cyclique

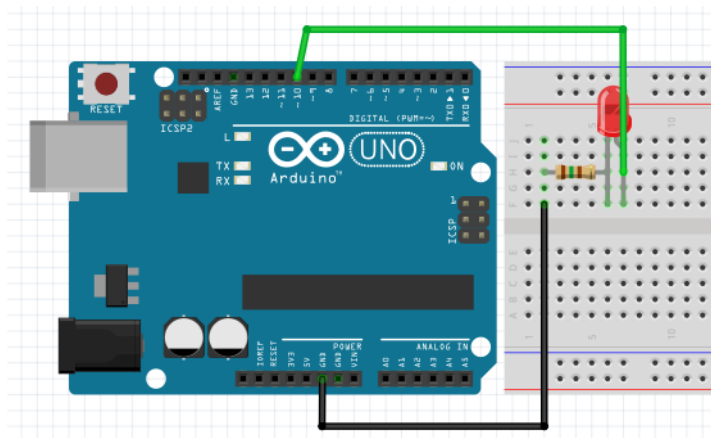


$$s(t) = \int_0^T s(t) dt = \frac{t_h}{T} \cdot E = RC \cdot E$$



Contrôler la luminosité d'une LED

- Modulation de largeur d'impulsions
- Exemple : Basics/Fade



- Quel est le signal appliqué sur la LED ?

Fade §

```
int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;   // how bright the LED is
int fadeAmount = 5;   // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

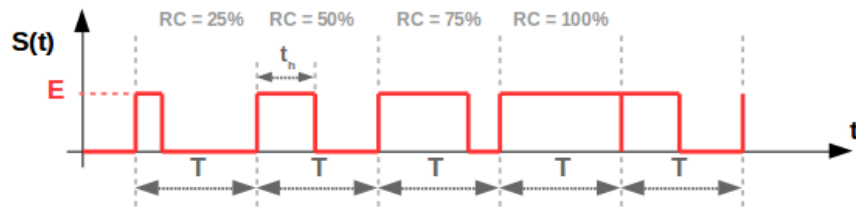
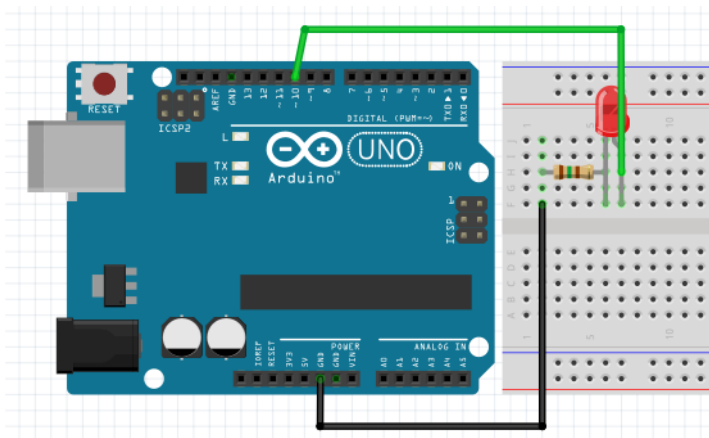
// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

Contrôler la luminosité d'une LED

- Modulation de largeur d'impulsions
- Exemple : Basics/Fade



Fade §

```
int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;   // how bright the LED is
int fadeAmount = 5;   // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

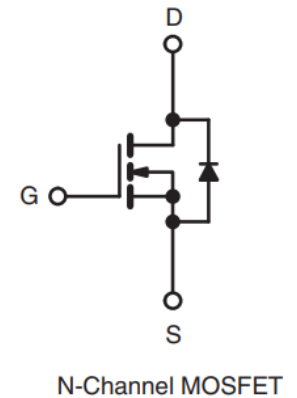
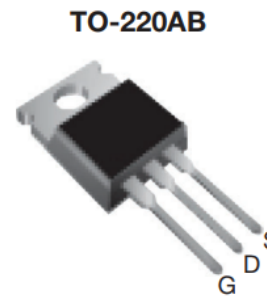
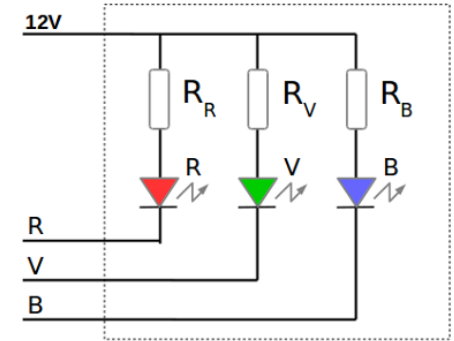
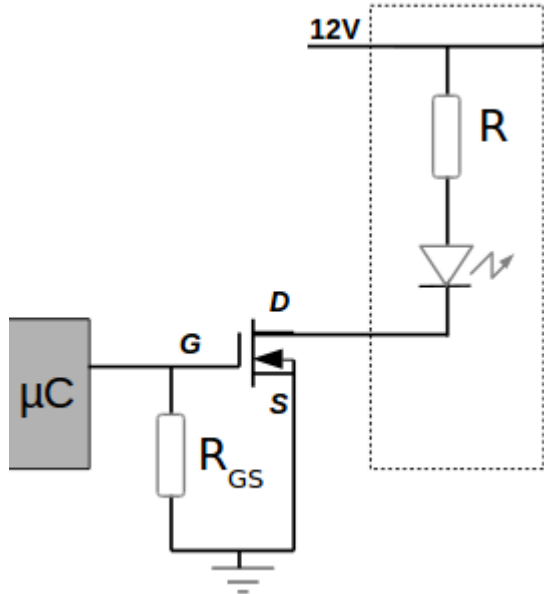
// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

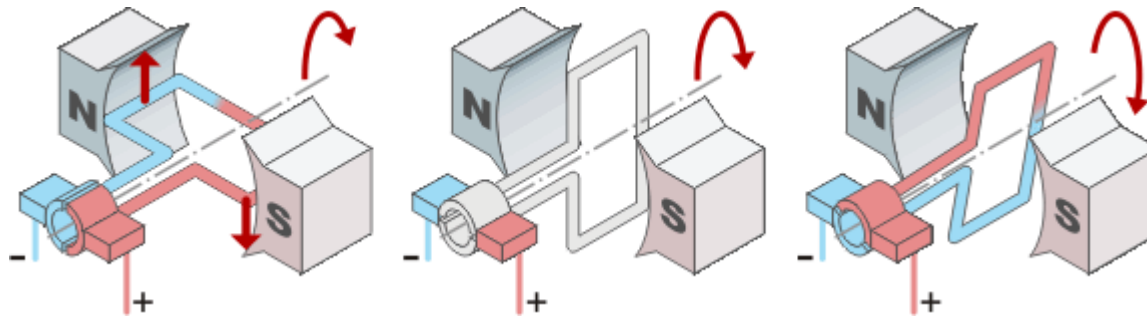
Piloter un composant de puissance

- Transistor IRL540 ou IRF540 ou BS170
 - Courant max ID
 - Tension max VDS
 - Tension min de commande VGS



Piloter un moteur à courant continu

- **Moteur** = élément de **conversion** d'une puissance **électrique** vers une puissance **mécanique**



- Aspect électrique : $P_{em} = E \cdot I$
- Aspect mécanique : $P_{em} = T_{em} \cdot \Omega$

où :

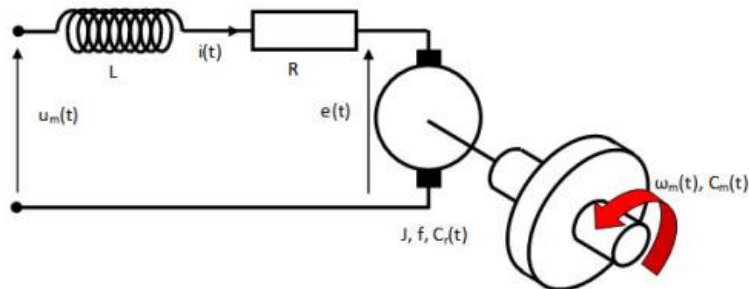
- E : la force contre-électromotrice [volt]
- I : le courant dans l'induit [ampère]
- T_{em} : le couple du moteur [N.m]
- Ω : vitesse angulaire de l'induit [rad/s]

- Couple souvent lié au courant
- **Éléments de puissance indispensables**

- **Transistor / Pont en H / Drivers**

Piloter un moteur à courant continu

- Vitesse proportionnelle à fem
- Quasiment proportionnelle à U



Les équations qui modélisent le comportement du moteur sont les suivantes :

Loi d'Ohm dans le circuit d'induit :	$u_m(t) = e(t) + R \cdot i(t) + L \cdot \frac{di(t)}{dt}$	(1)
Équations de l'électromagnétisme dans le moteur :	$e(t) = K_e \cdot \omega(t)$	(2)
	$C_m(t) = K_c \cdot i(t)$	(3)
Équation de la dynamique de l'arbre moteur :	$C_m(t) - C_r(t) - f \cdot \omega(t) = J \cdot \frac{d\omega(t)}{dt}$	(4)

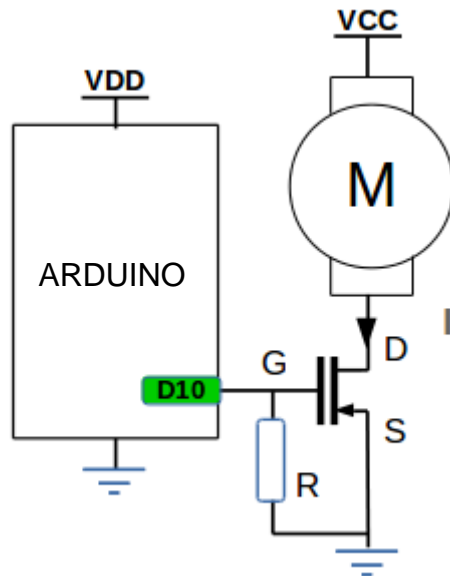
Avec :

- $u_m(t)$ est la tension d'alimentation (V)
- $i(t)$ est le courant consommé (A)
- $e(t)$ est la tension contre-électromotrice (V)
- R est la valeur de la résistance (Ω)
- L est la valeur de l'inductance (H)
- K_e est le coefficient de fem (V/(rad/s))
- $\omega(t)$ est la vitesse de rotation de l'arbre moteur (rad/s)
- f est le paramètre de frottement "fluide" total (N.m/(rad/s))
- J est l'inertie totale ramenée sur l'axe moteur (kg.m²)
- K_c est la constante de couple (N.m/A)
- $C_m(t)$ est le couple moteur. (N.m)
C'est « l'effort tournant » qu'est capable de fournir le moteur. Plus ce couple est important et plus le moteur aura la capacité à faire tourner une lourde charge.
- $C_r(t)$ est le couple résistant sur l'axe moteur. (N.m)
C'est un effort qui s'oppose au mouvement de rotation du moteur et qui a tendance à le freiner. Par exemple, des herbes hautes et épaisses vont générer un couple résistant au niveau des lames fixées sur l'axe de sortie du moteur électrique que l'on trouve dans une tondeuse.

- Facile à mettre en œuvre
- Peu de couple
- Asservissement de position nécessitant un encodeur externe
- Usure mécanique (balais)

Piloter un moteur à courant continu

- Une seule direction / Transistor



Fade §

```
int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;   // how bright the LED is
int fadeAmount = 5;   // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

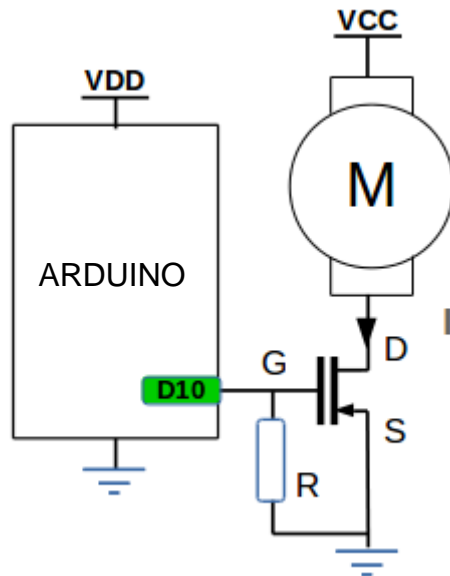
  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

- Et pour changer de sens de rotation ?

Piloter un moteur à courant continu

- Une seule direction / Transistor



Fade§

```
int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;   // how bright the LED is
int fadeAmount = 5;   // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

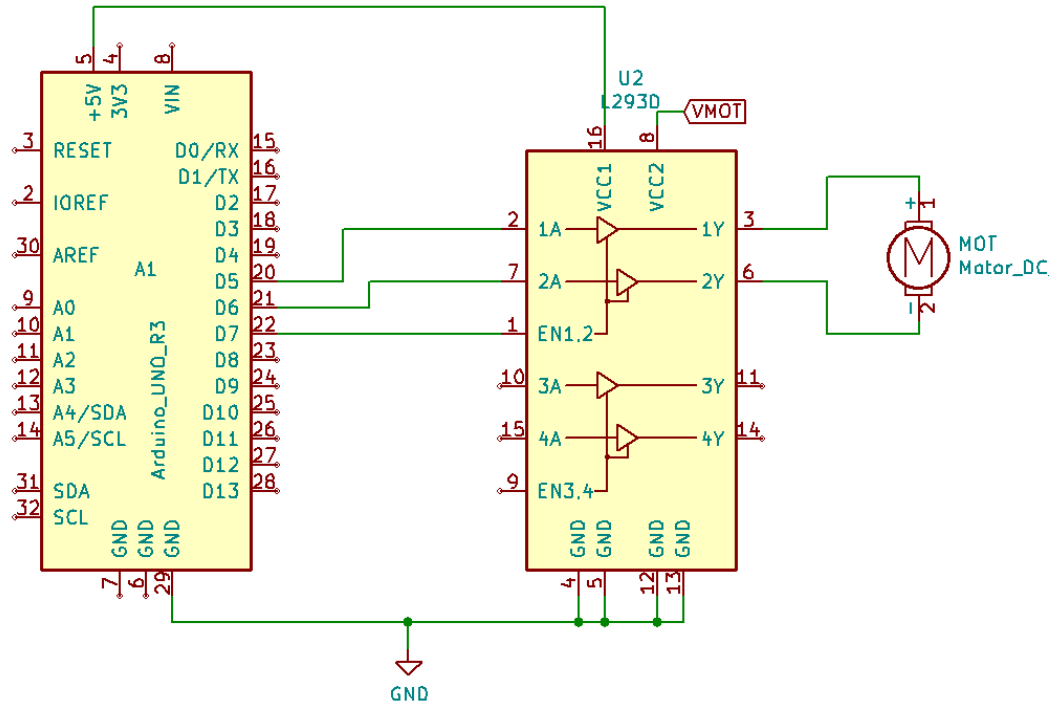
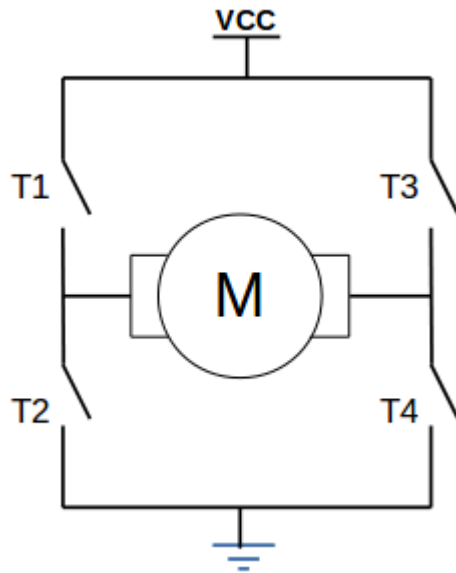
  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

- Et pour changer de sens de rotation ?

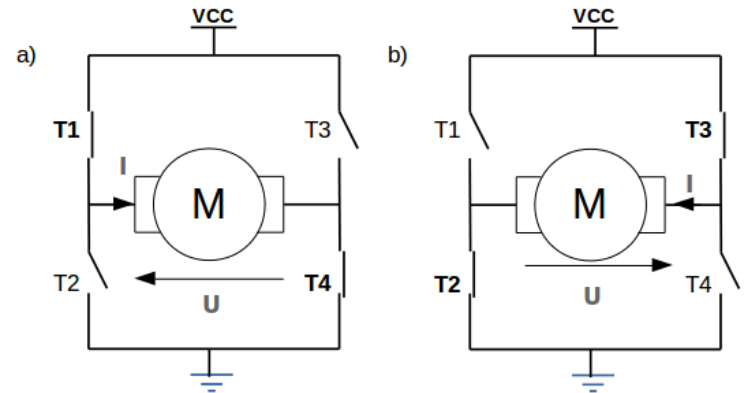
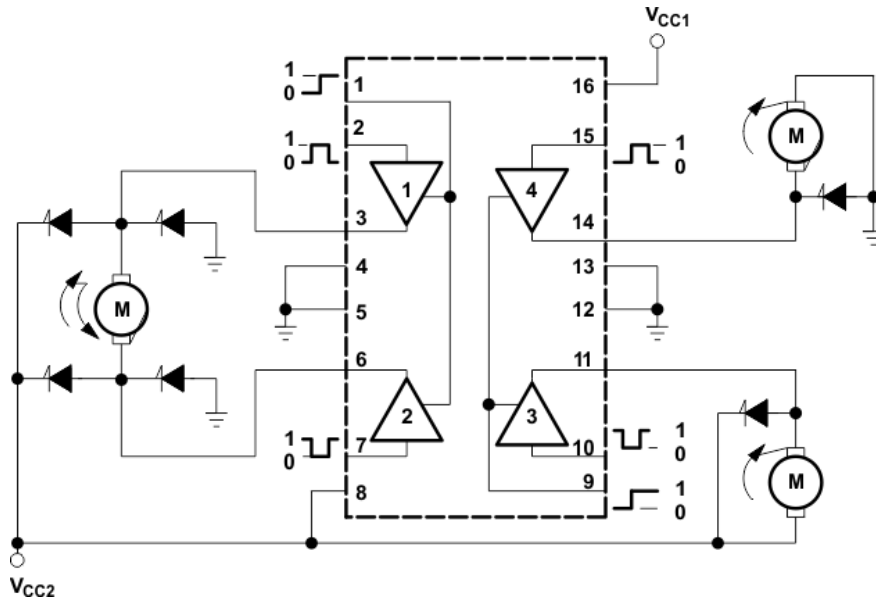
Piloter un moteur à courant continu

- Deux directions / Pont en H / L293-D

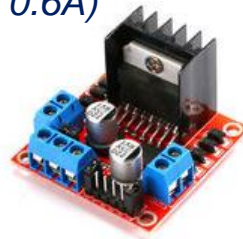


Piloter un moteur à courant continu

- Courant élevé



**Pont en H – L293D (avec diode roue libre – 0.6A)
ou L298 (sans diode roue libre – 1.5A)**



- **Facile à mettre en œuvre**
- *Peu de couple*
- *Asservissement de position nécessitant un encodeur externe*
- *Usure mécanique (balais)*

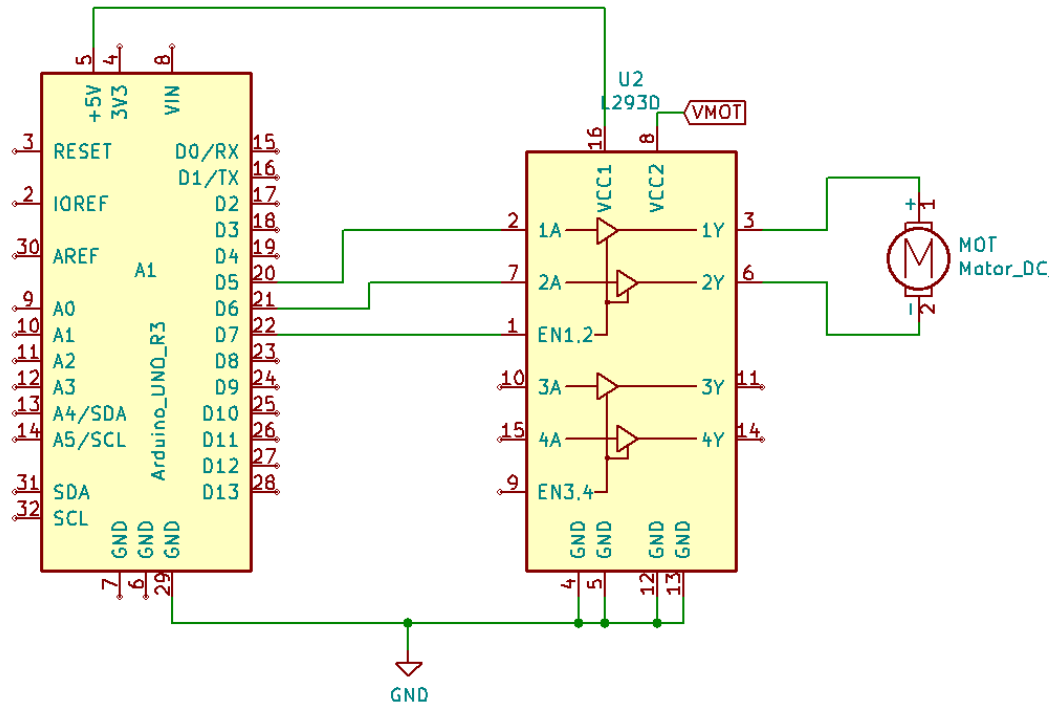
Piloter un moteur à courant continu

- Deux directions / Pont en H / L293-D

MCC8PWM

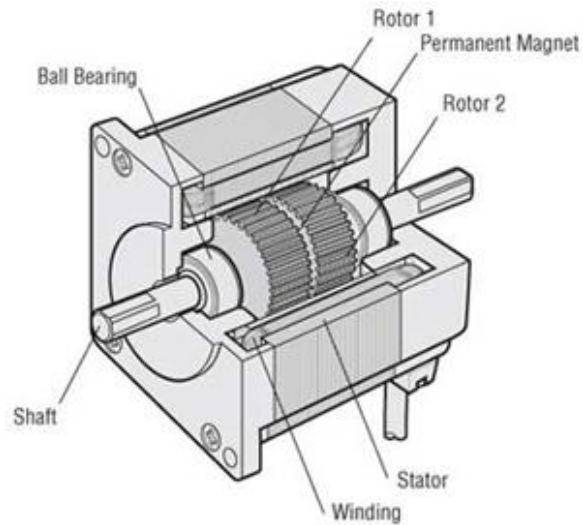
```

if(etat_activation == 1){
  val_pot = analogRead(A0);
  rapport_cyclique = map(val_pot, 0, 1023, -255, 255);
  if(rapport_cyclique < 0){
    digitalWrite(enableAB, HIGH);
    analogWrite(moteurA, -rapport_cyclique);
    analogWrite(moteurB, 0);
  }
  else{
    digitalWrite(enableAB, HIGH);
    analogWrite(moteurA, 0);
    analogWrite(moteurB, rapport_cyclique);
  }
}
else{
  digitalWrite(enableAB, LOW);
  analogWrite(moteurA, 0);
  analogWrite(moteurB, 0);
}
}
    
```



Piloter un moteur pas à pas

- Avancement pas par pas



Motor Structural Diagram: Cross-Section Parallel to Shaft

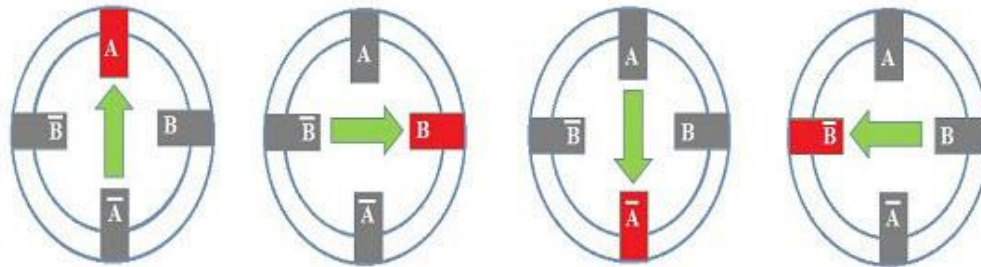


- **Asservissement de position « inclus »**
- **Couple intéressant**
- *Pilotage à maîtriser*
- *Vitesse réduite*

Piloter un moteur pas à pas

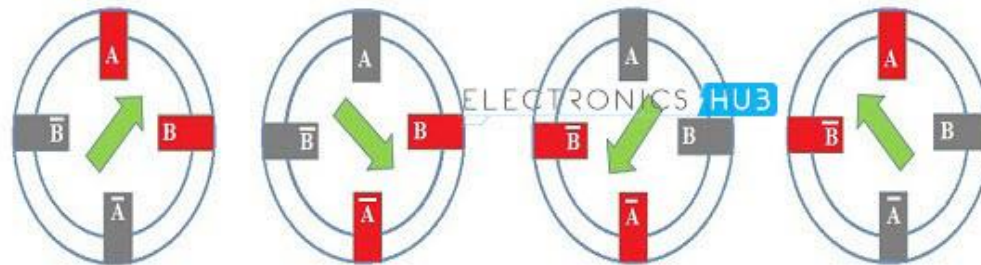
- Avancement pas par pas

Full Step - One Phase ON

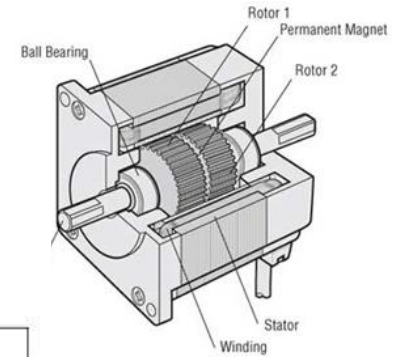


Step	Phase			
	A	B	\bar{A}	\bar{B}
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Full Step - Two Phase ON



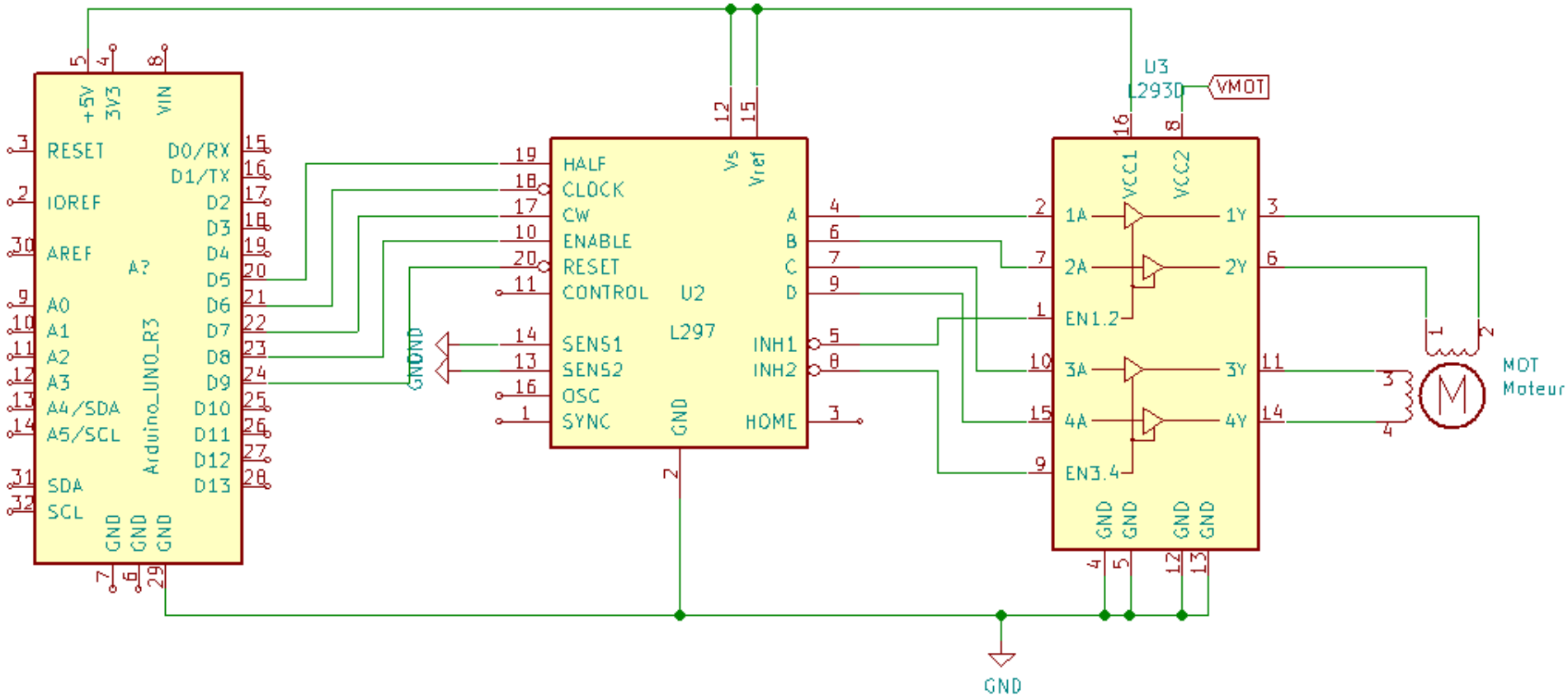
Step	Phase			
	A	B	\bar{A}	\bar{B}
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1



Structural Diagram: Cross-Section Parallel to Shaft

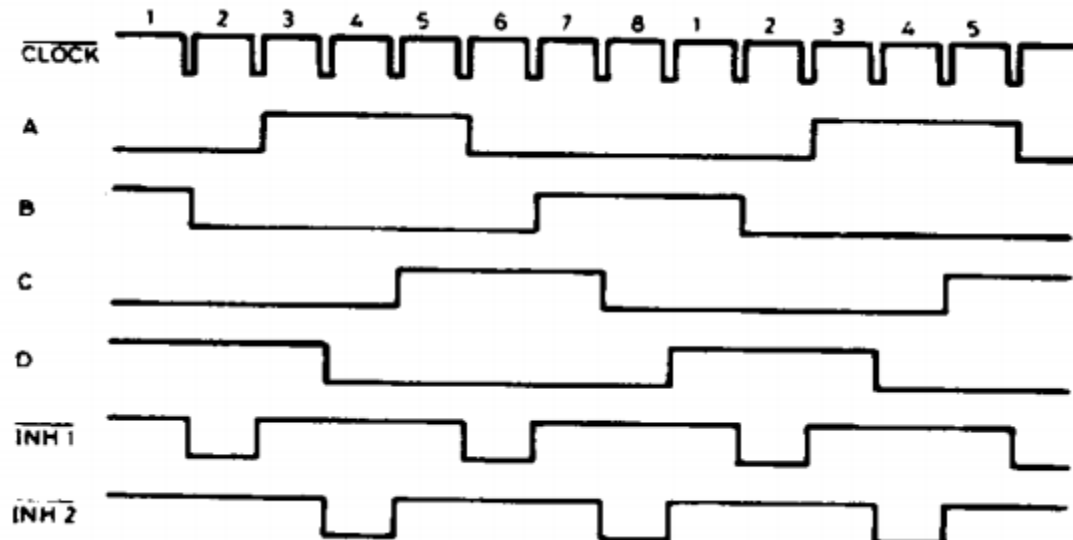
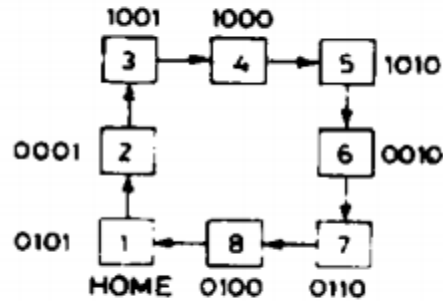
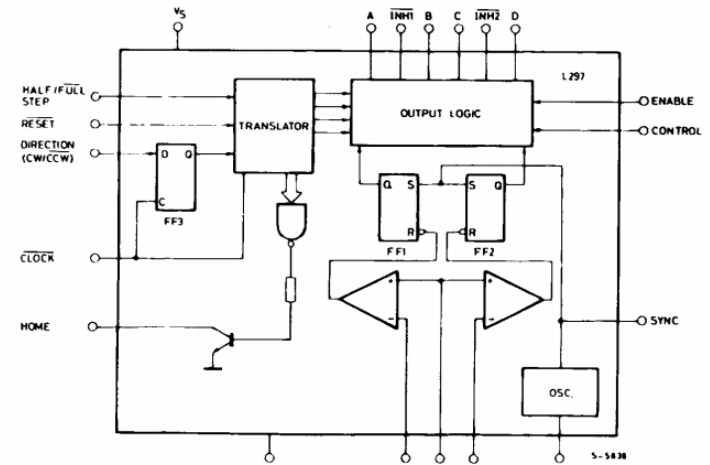
Piloter un moteur pas à pas

- Pont en H / L293D
- Driver / L297



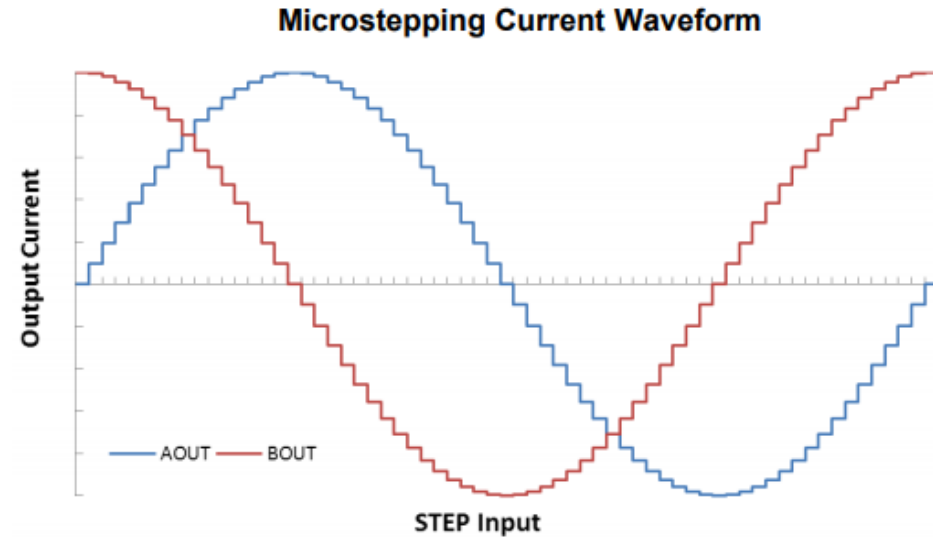
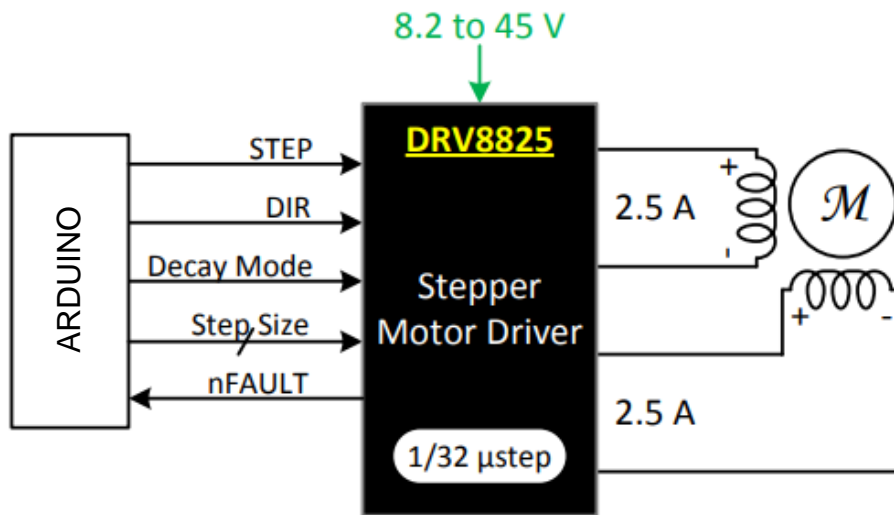
Piloter un moteur pas à pas

- Pont en H / L293D
- Driver / L297
- Commande en Demi-pas



Piloter un moteur pas à pas plus subtilement

- Utilisation d'un pont en H
- Commande avec **DRV8825**



Piloter un moteur *brushless*

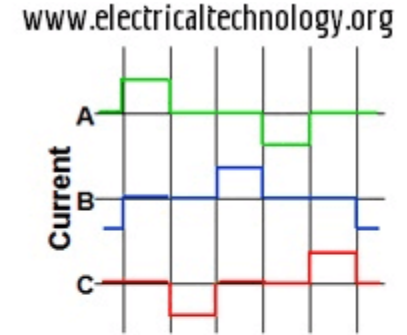
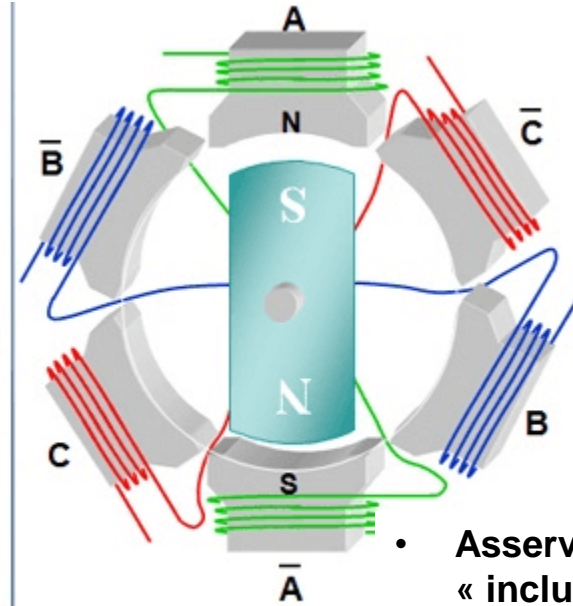
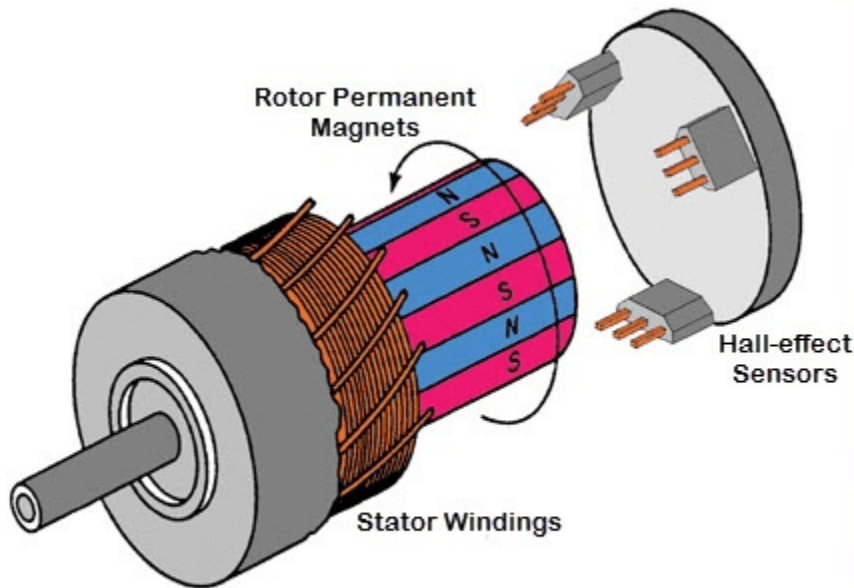
- BLDC : BrushLess Direct Current
machine synchrone auto-pilotée à aimants permanents
- Avancement pas par pas
- Sans contact entre rotor et stator



- **Asservissement de position**
« inclus »
- **Couple intéressant**
- *Pilotage à maîtriser*

Piloter un moteur *brushless*

- BLDC : BrushLess Direct Current
machine synchrone auto-pilotée à aimants permanents

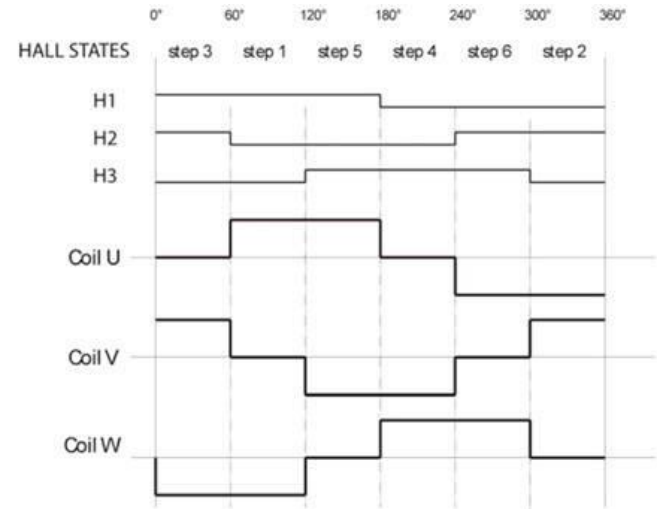
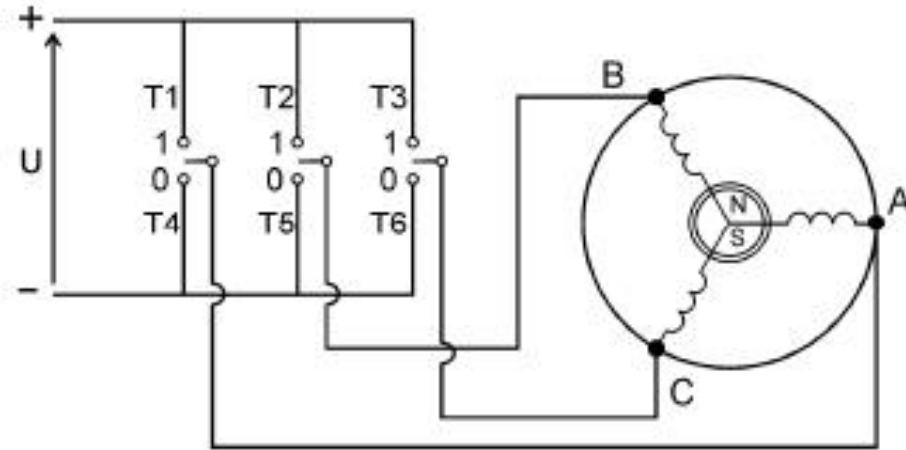


Construction, Working Principle & Operation of BLDC Motor

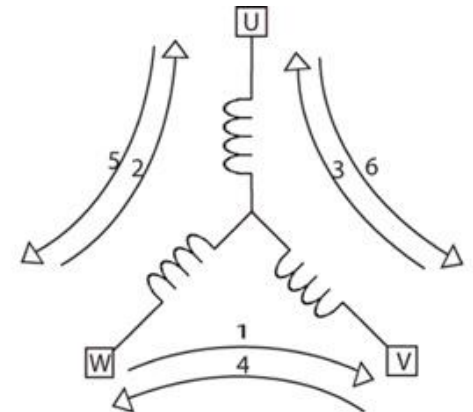
- Asservissement de position « inclus »
- Couple intéressant
- Pilotage à maîtriser

Piloter un moteur *brushless*

- Transistors

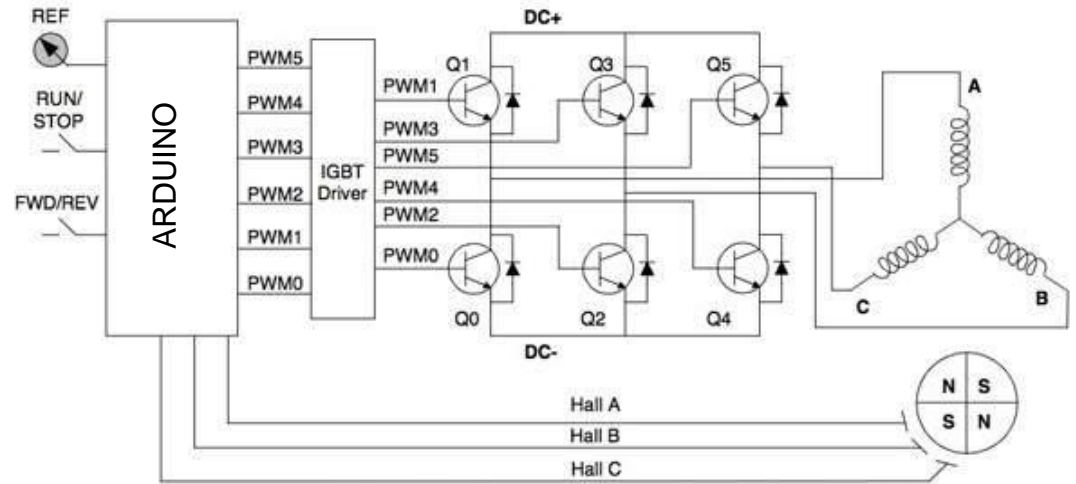
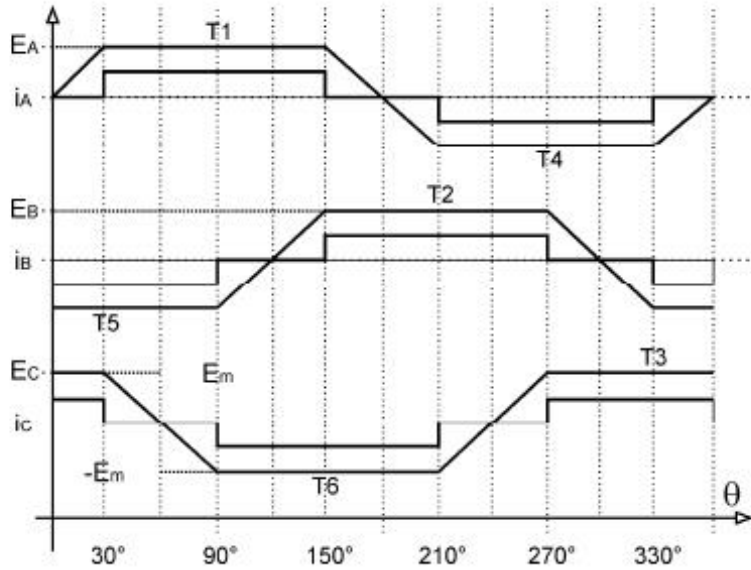
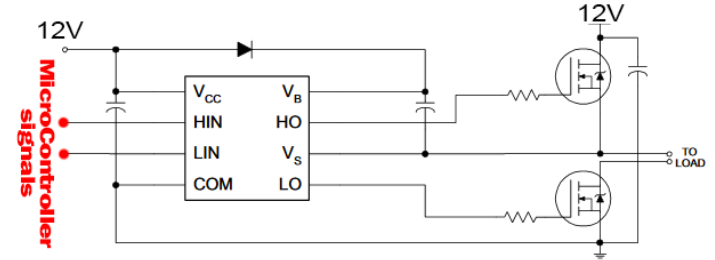


<http://www.energoelektronika.pl/do/ShowNews?id=1599>



Piloter un moteur *brushless*

- Avec drivers de MOS



<http://www.energoelektronika.pl/do/ShowNews?id=1599>

<https://www.digikey.com/en/articles/techzone/2013/mar/an-introduction-to-brushless-dc-motor-control>

Piloter un moteur *brushless*

- Avec drivers de MOS

Commande BLDC

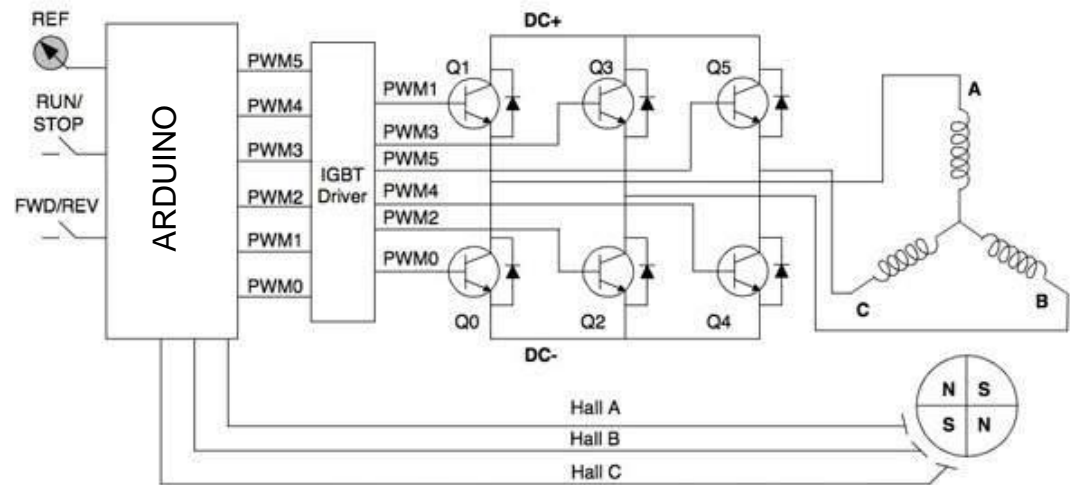
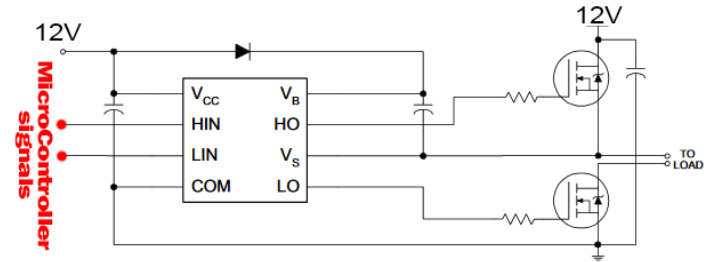
case 1:

```
digitalWrite(AA1, LOW);
digitalWrite(AA2, LOW);
digitalWrite(BB1, LOW);
digitalWrite(CC2, LOW);
digitalWrite(BB2, HIGH);
digitalWrite(CC1, HIGH);
delta = emA-sum;
break;
```

//Phase2 A-B

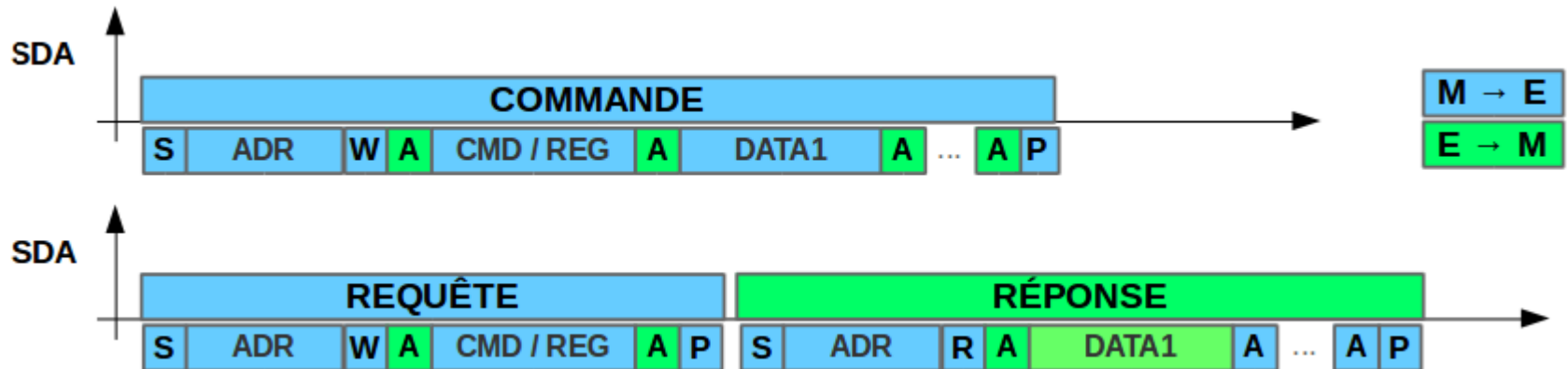
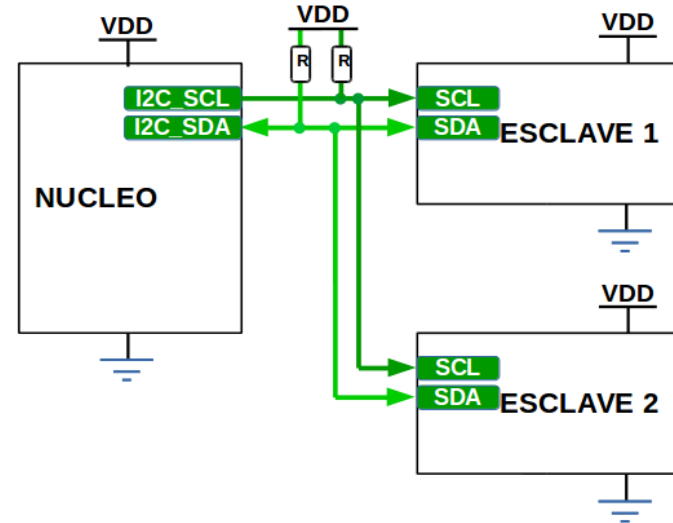
case 2:

```
digitalWrite(AA2, LOW);
digitalWrite(BB1, LOW);
digitalWrite(CC1, LOW);
digitalWrite(CC2, LOW);
digitalWrite(AA1, HIGH);
digitalWrite(BB2, HIGH);
delta = emC-sum;
```



Communiquer en I2C

- Réseau de terrain sur 2 fils
- Relation maître/esclaves
- Adressage des esclaves



Communiquer en I2C

- Exemple du **TC1321** / DAC sur carte *Microchip I2C*

- Bibliothèque **WIRE**

TABLE 4-2: TC1321 COMMAND SET (READ_BYTE AND WRITE_BYTE)

Command Byte Description		
Command	Code	Function
RWD	00h	Read/Write Data (DATA)
RWCR	01h	Read/Write Configuration (CONFIG)

TABLE 4-3: CONFIGURATION REGISTER (CONFIG), 8-BIT, READ/WRITE

Configuration Register (CONFIG)									
Bit Name	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]	
Bit Function	Reserved (Note 1)							SHDN (Note 2)	

Note 1: Always returns '0' when reading
 Note 2: 1 = Standby (Shut down) mode
 0 = Normal mode

TABLE 4-4: DATA REGISTER (DATA), 10-BIT, READ/WRITE

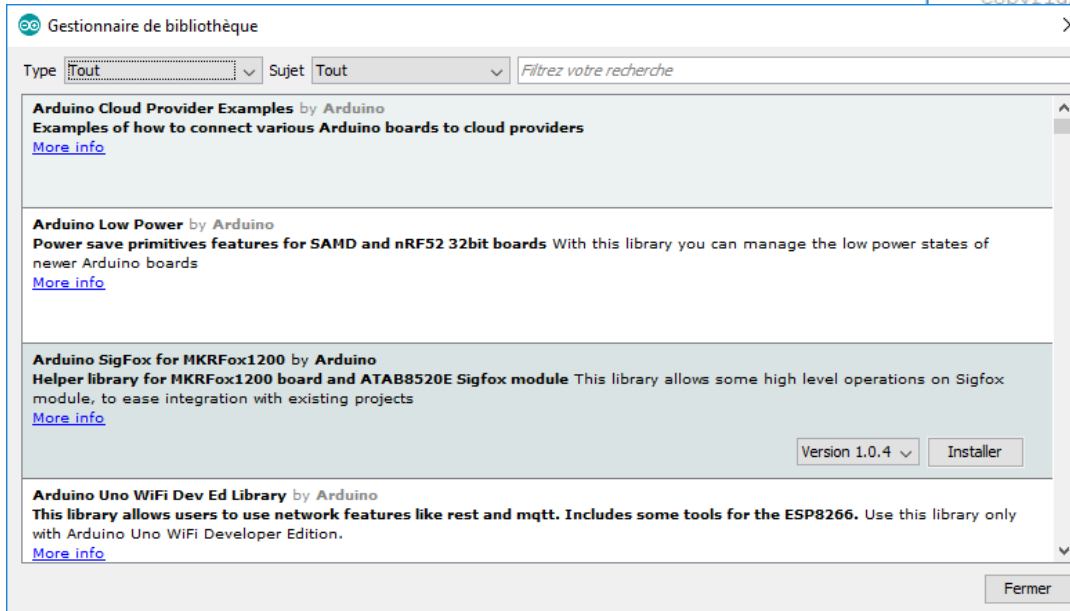
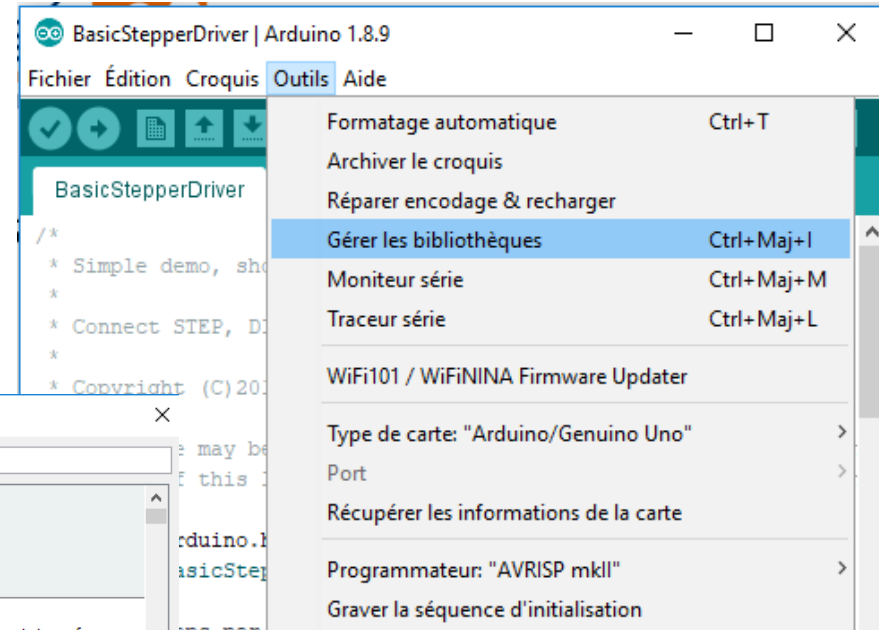
Data Register (DATA) for 1st Byte								Data Register (DATA) for 2nd Byte								
D[9]	D[8]	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]	X	X	X	X	X	X	
MSB	X	X	X	X	X	X	X	X	X	LSB	X	X	X	X	X	X

Functions

- `begin()`
- `requestFrom()`
- `beginTransaction()`
- `endTransmission()`
- `write()`
- `available()`
- `read()`
- `SetClock()`
- `onReceive()`
- `onRequest()`

Des tonnes de bibliothèques

- Brushless Servo
- Stepper Driver / DRV8825



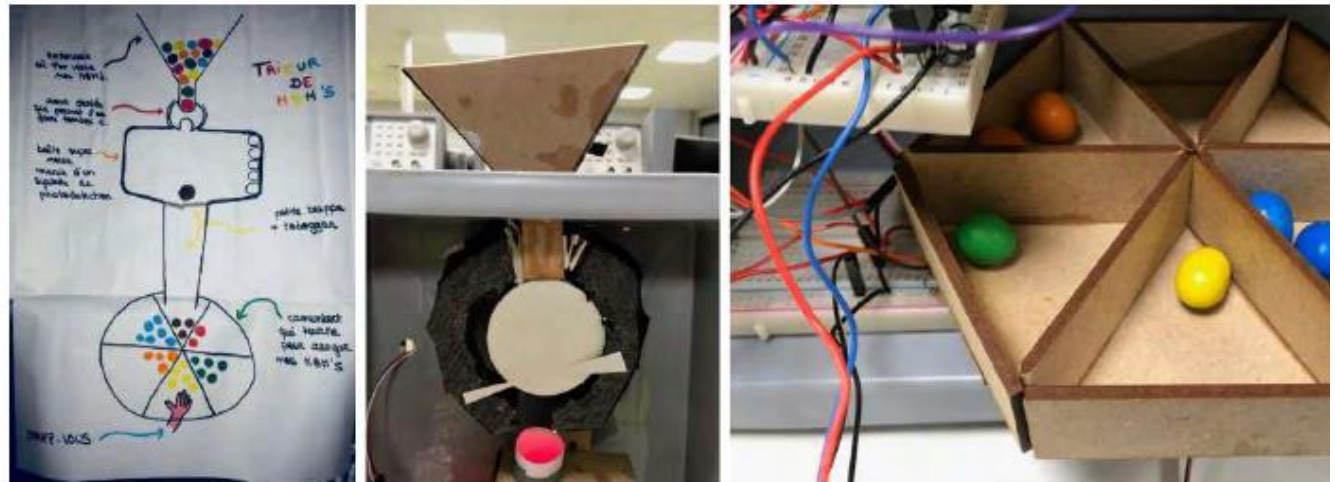
Systemes embarqués

Quelques exemples

Quelques exemples

<http://lense.institutoptique.fr/>

realisations
nucleo



Quelques exemples

<http://lense.institutoptique.fr/>

realisations
nucleo

