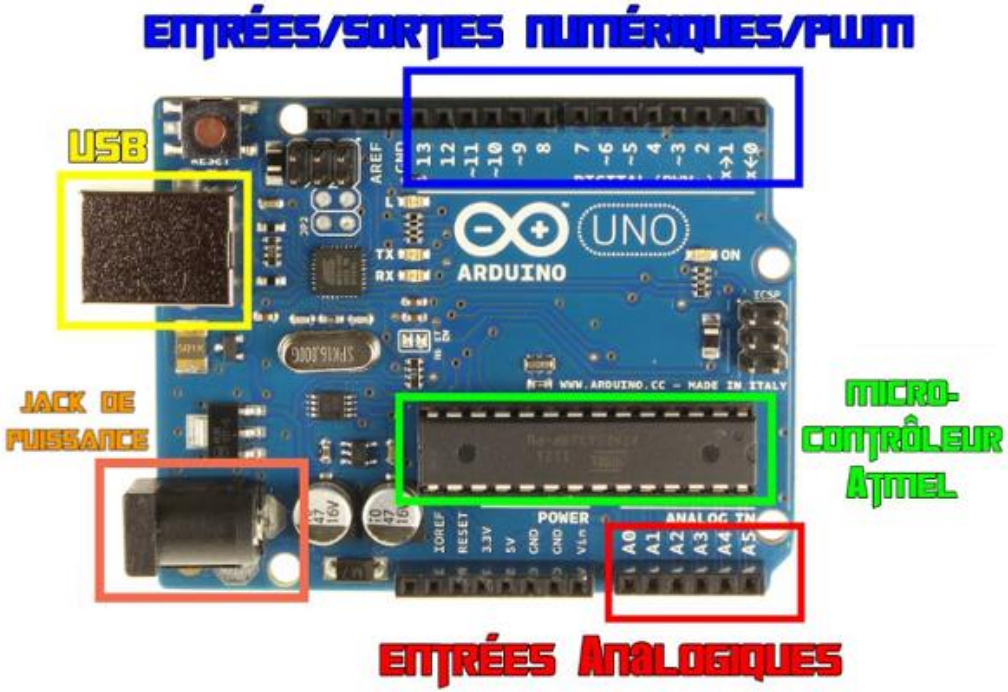
ARDUINO
C’est quoi donc ?

Julien VILLEMEJANE

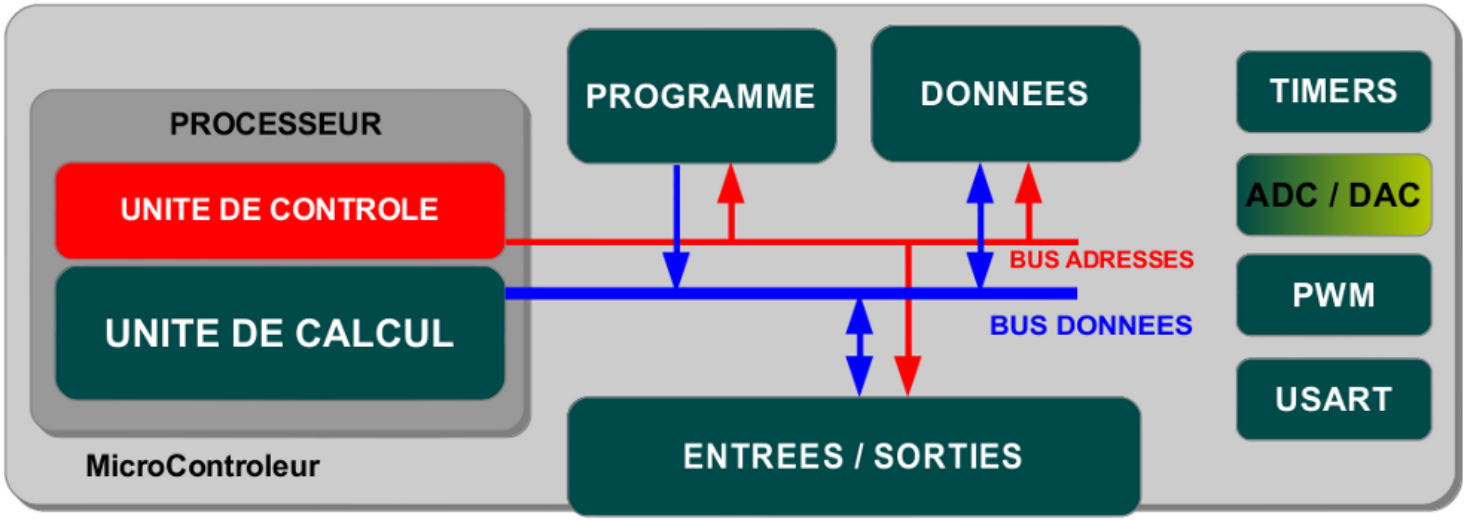
Qu’est-ce qu’une carte Arduino ?

- Microcontrôleur
- Alimentation
- Téléversement
- Entrées / Sorties



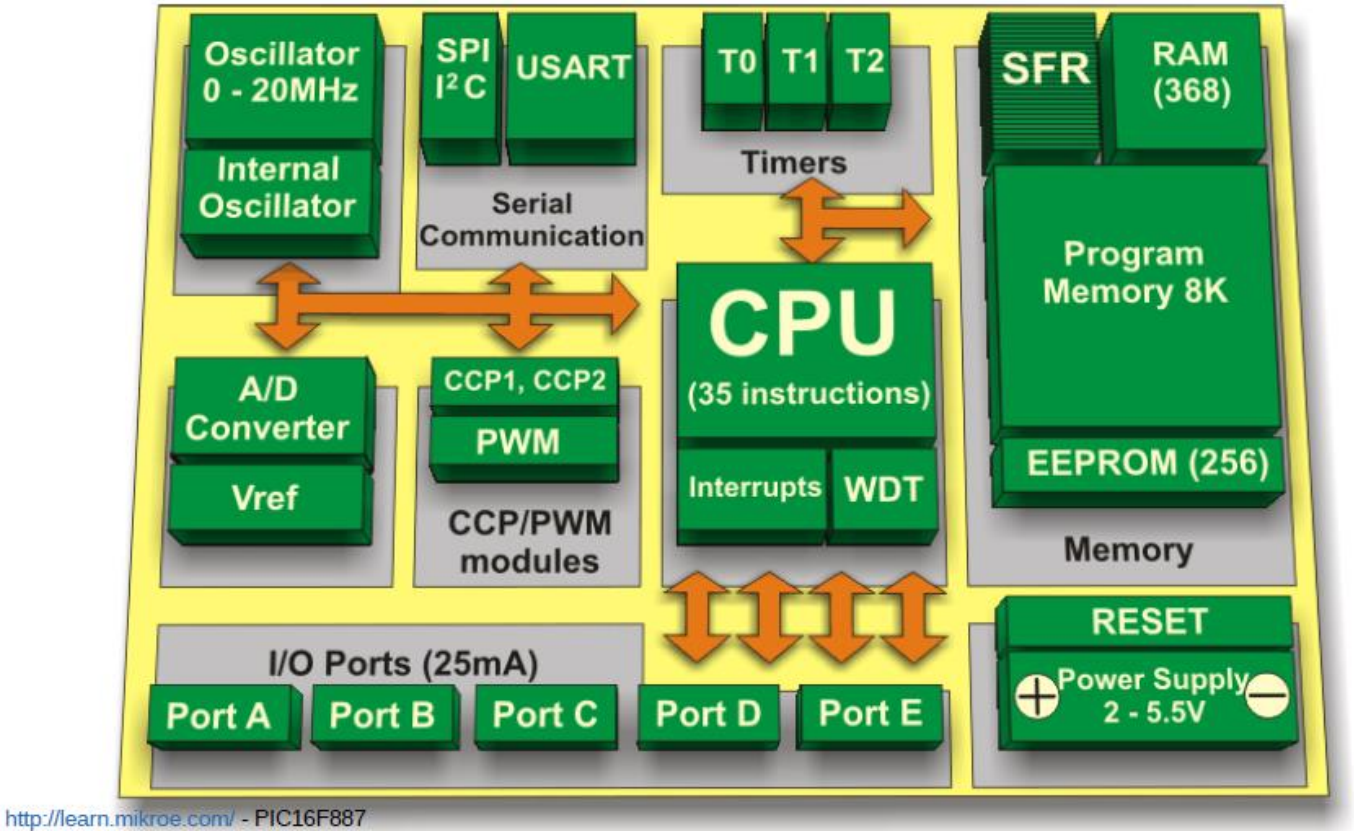
Qu’est-ce qu’un microcontrôleur ?

- Processeur spécialisé
- Entrées / Sorties réactives



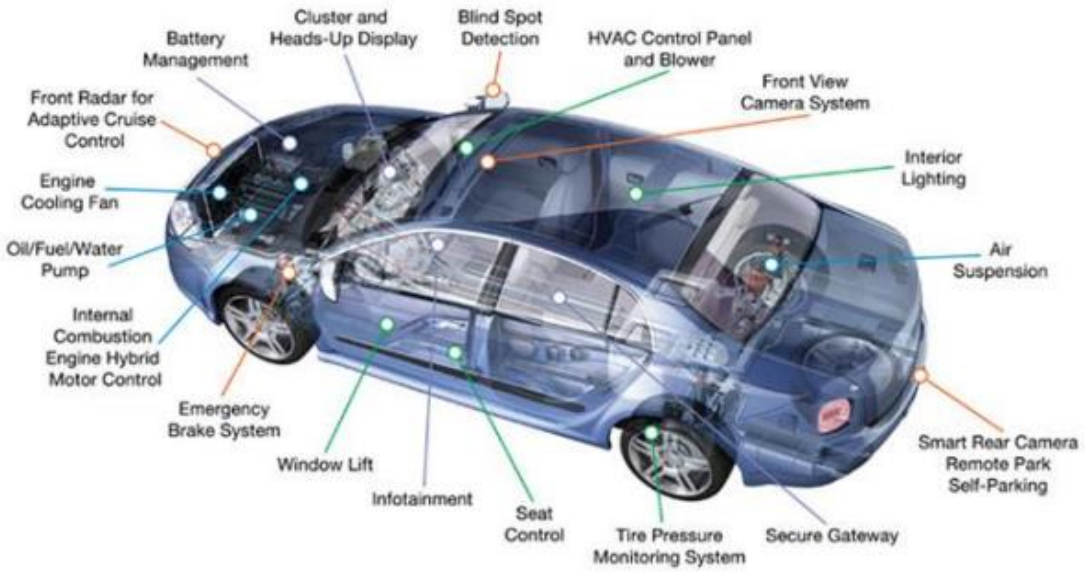
Qu'est-ce qu'un microcontrôleur ?

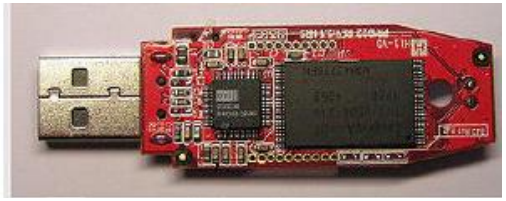
- Processeur spécialisé
- Entrées / Sorties réactives



Où sont-ils utilisés ?

- Système embarqué
- Capteurs intelligents





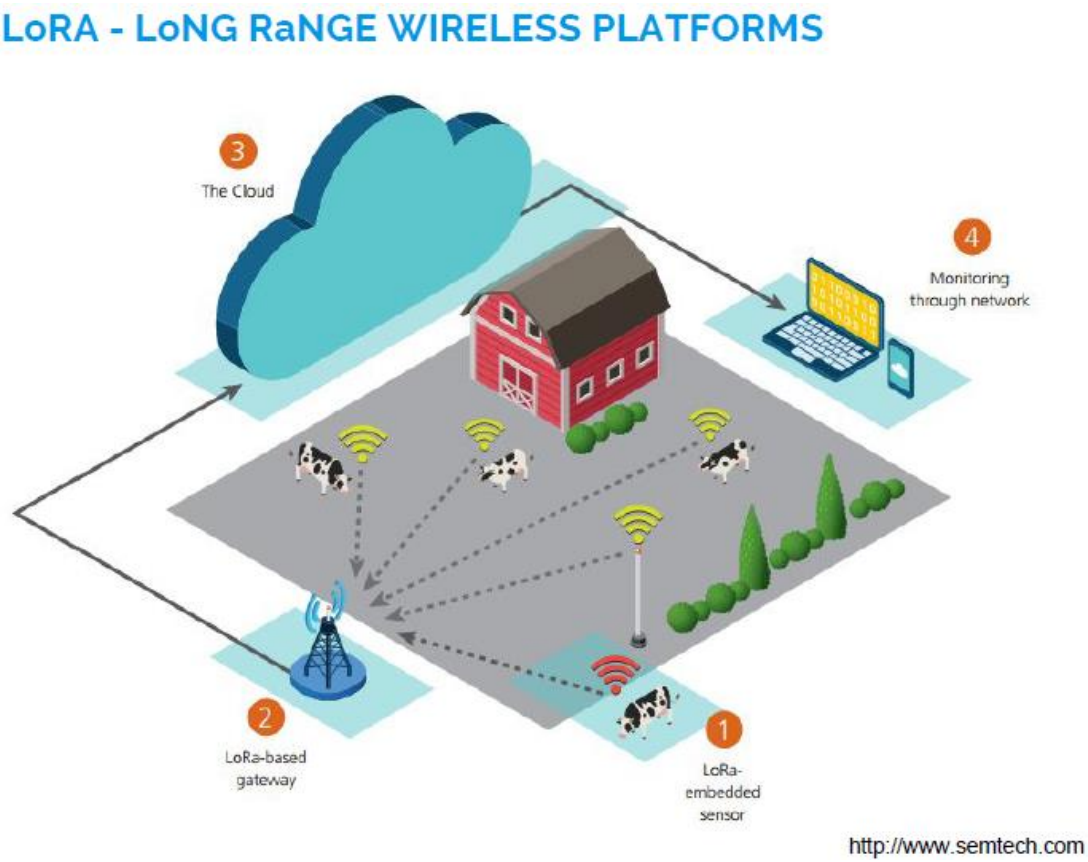
Paris-Saclay

Saint-Étienne

Bordeaux

Où sont-ils utilisés ?

- Système embarqué
- Capteurs intelligents



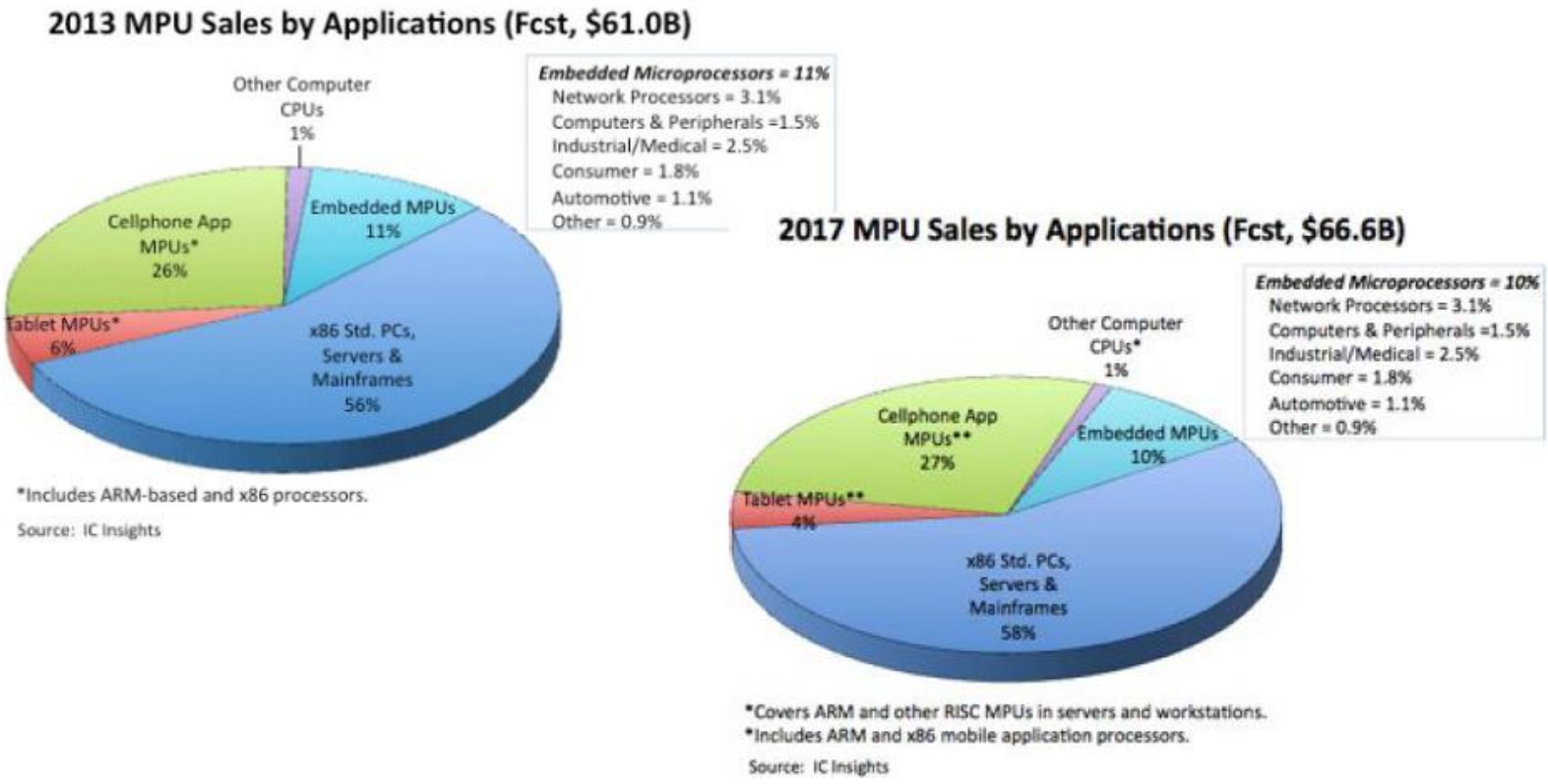
Paris-Saclay

Saint-Étienne

Bordeaux

Et par rapport à un PC ?

- Processeurs généralistes vs Processeurs embarqués



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

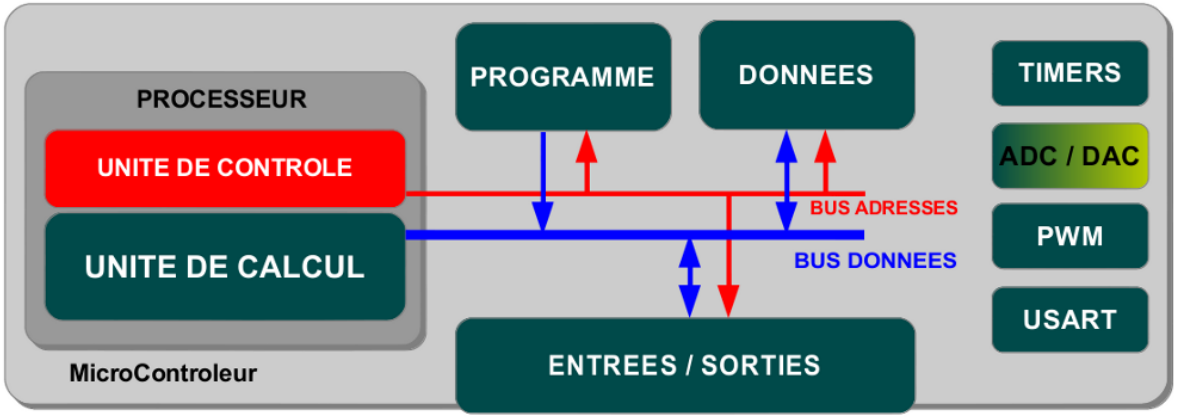
	PC standard	ARDUINO Uno
Fréquence	2 GHz	16 MHz
Core / Architecture	4 / 64 bits	1 / 8 bits
Consommation	100 à 500 W	< 1 W
Entrées/Sorties	/	6 Analog / 13 Digital
Ports extension	USB, PCI…	SPI, I2C, RS232
RAM	4 Go	2 ko

- Et combien de temps met une application à se lancer sur un PC ?
- Est-ce une durée fixe ?

Revenons à Arduino

- Accès à des entrées/sorties réactives

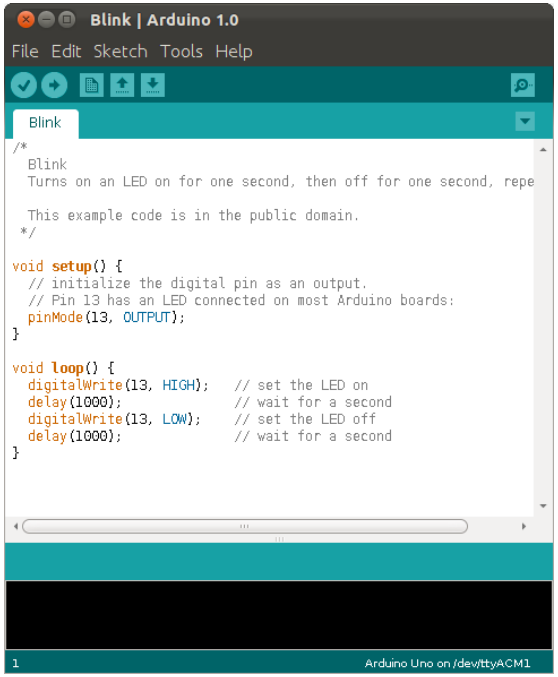
ARDUINO Uno
16 MHz
1 core / 8 bits
< 1 W
6 Analog / 13 Digital
SPI, I2C, RS232
2 ko

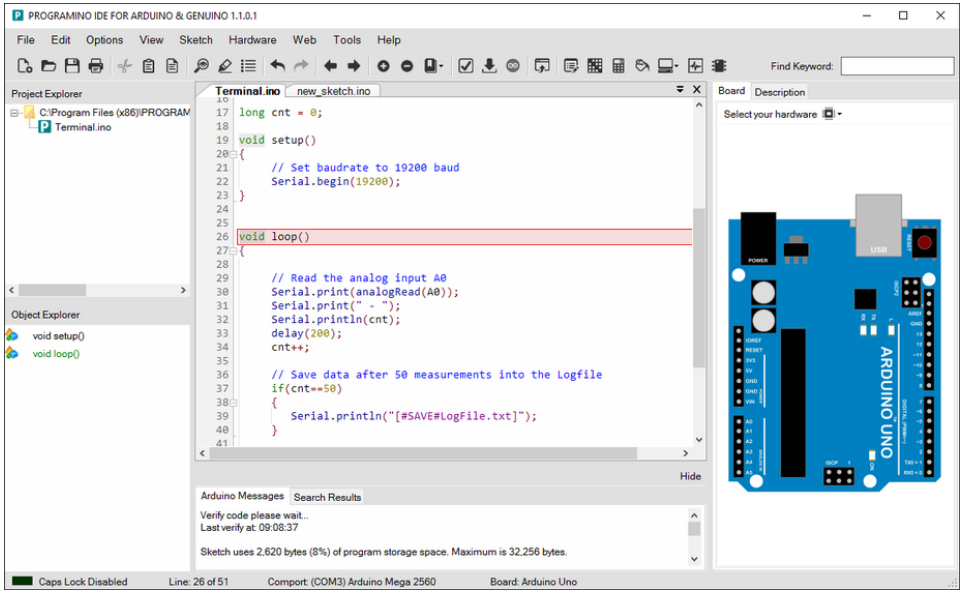


- Grande communauté
- Shields (extensions)

Comment ça marche une carte Arduino ?

- Programme
- IDE = Interface de Développement
- Téléversement





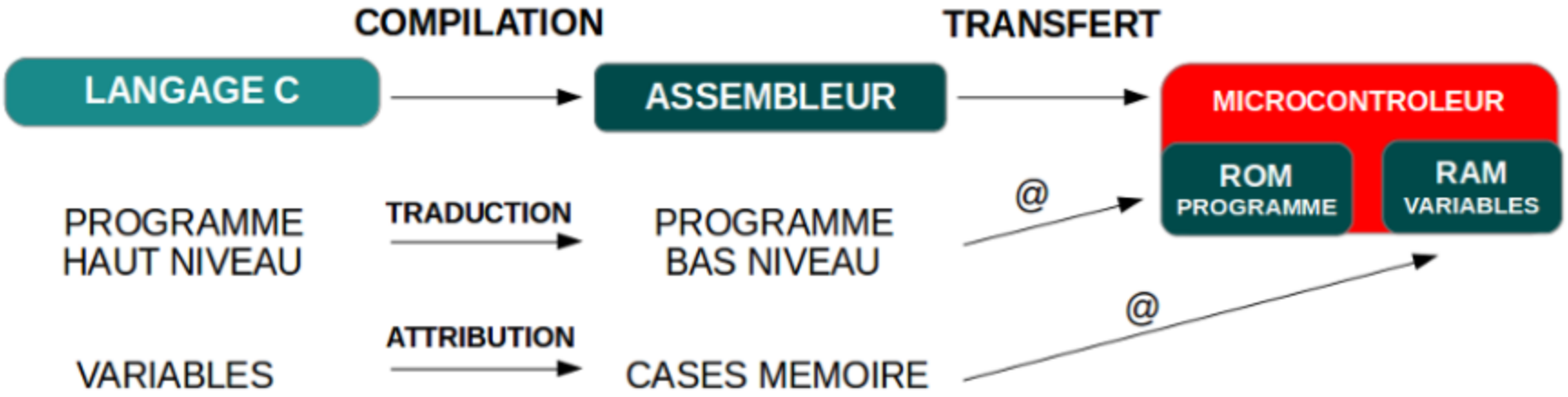
Paris-Saclay

Saint-Étienne

Bordeaux

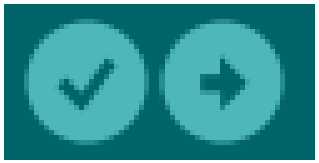
Comment ça marche une carte Arduino ?

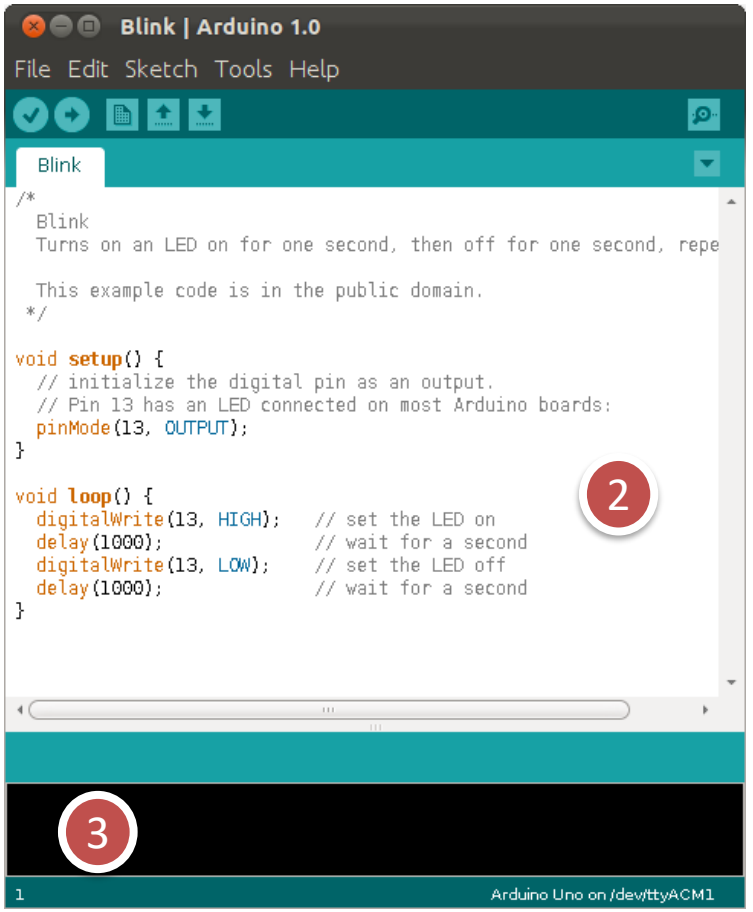
- Programme
- IDE = Interface de Développement
- Téléversement



Mon premier programme Arduino

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



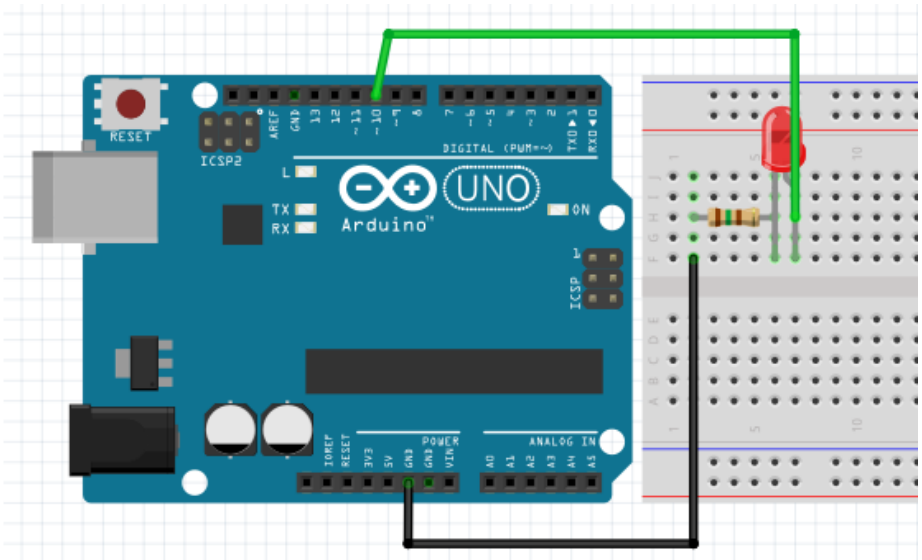


Piloter une sortie externe

- Cablage d'une LED sur D10

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

- Programme ***Blink*** modifié



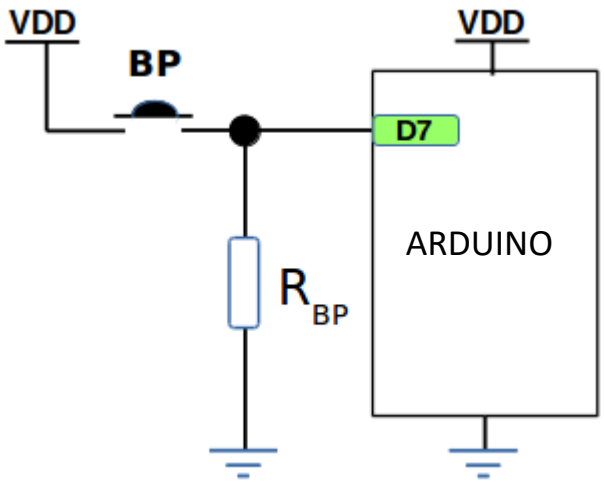
Blink §

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

Câbler un bouton poussoir

Récupérer une donnée numérique

- R_{BP} pour protéger l'alimentation



Blink §

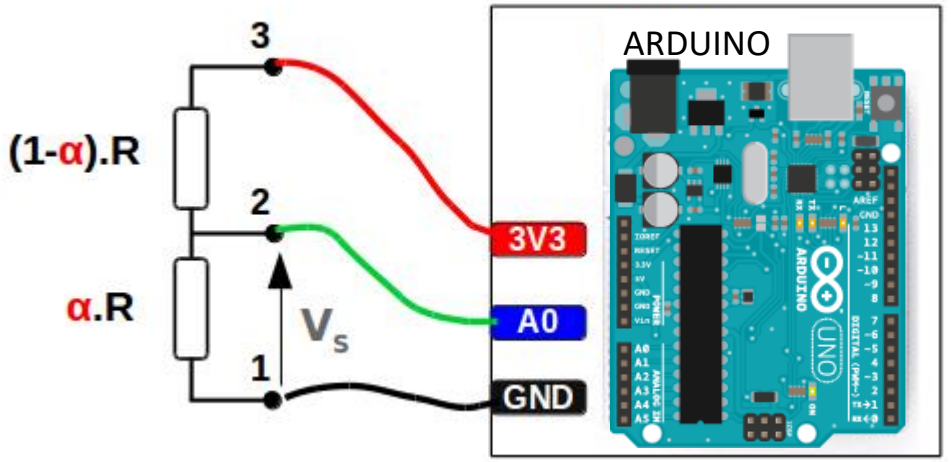
```
int val = 0;

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

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

Récupérer une donnée analogique

- Câblage potentiomètre

AnalogReadSerial §

```
void setup() {

}

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

  delay(1);
}
```



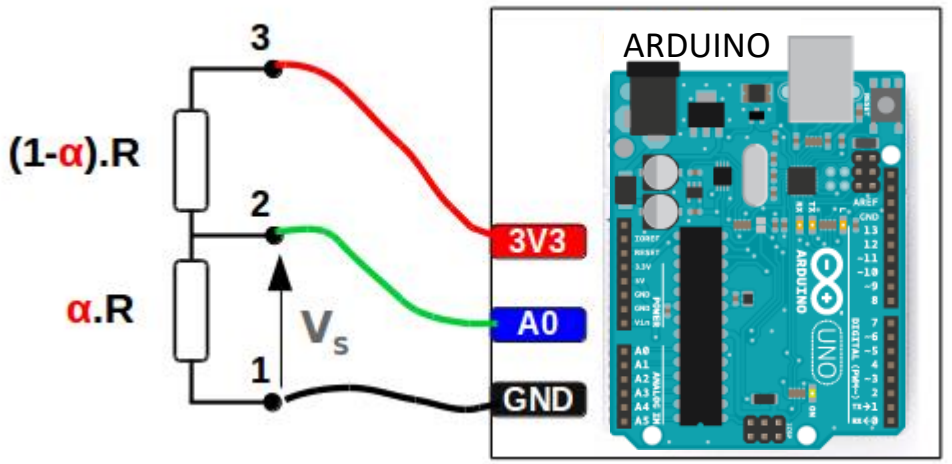
- Comment vérifier que la donnée est bien convertie ?
  - CAN intégré – 10 bits

Récupérer 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

Récupérer une donnée analogique

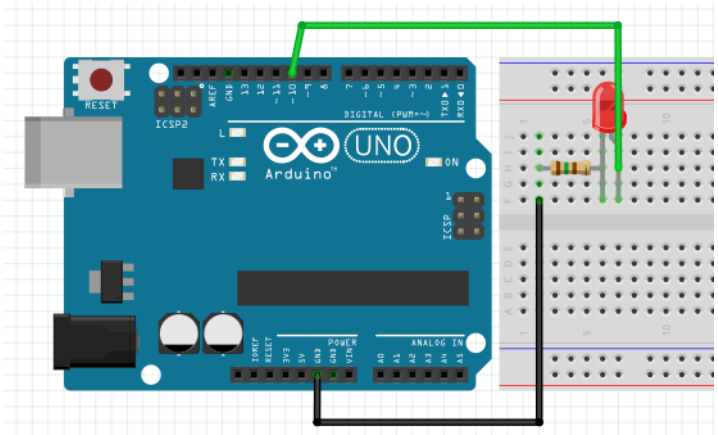
- Câblage potentiomètre
- **Affichage console + Traceur** / Exemple : Basics/AnalogReadSerial

AnalogReadSerial §

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

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

Contrôler la luminosité d'une LED

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



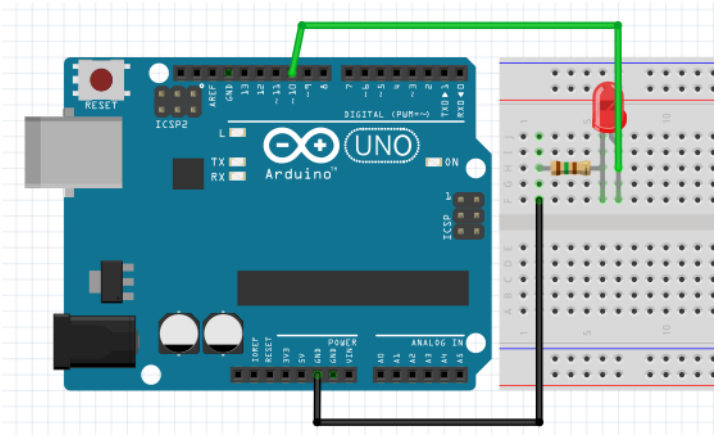
- Quel est le signal appliqué sur la LED ?

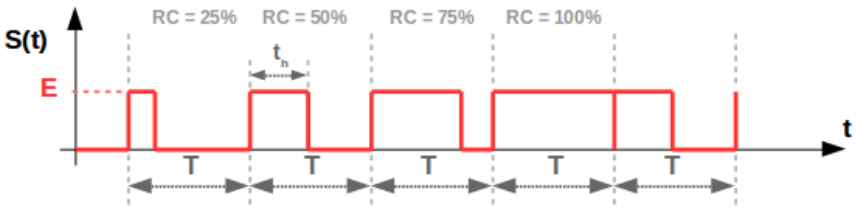
Fade §

```
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



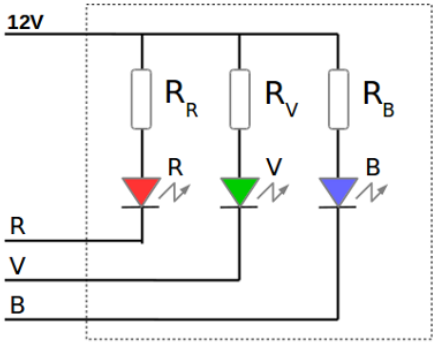


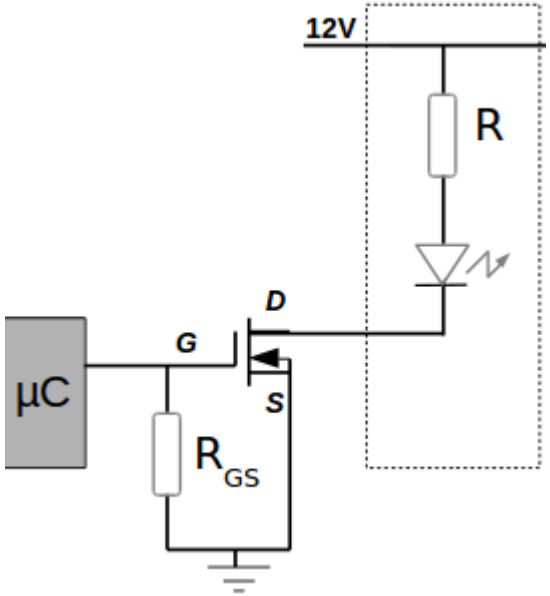
Fade §

```
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);
}
```

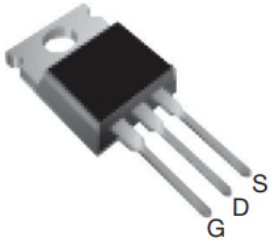
Piloter un bandeau de LEDs

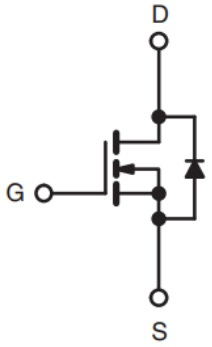
- Transistor IRL540 ou BS170
- Bandeau de LED





TO-220AB

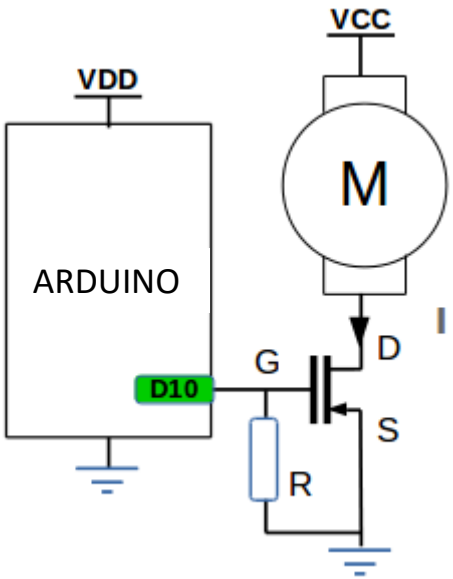




N-Channel MOSFET

Piloter un moteur à courant continu

- Une seule direction / Transistor

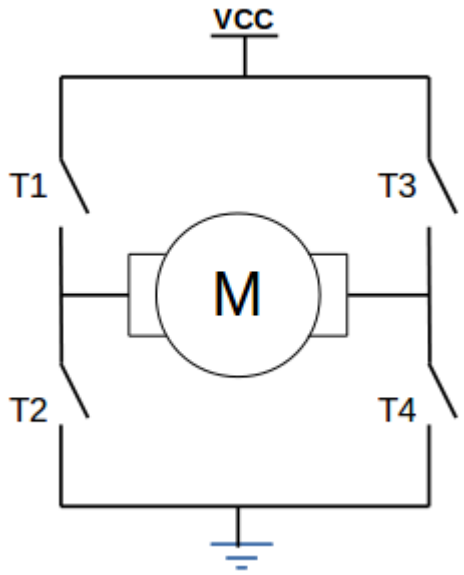


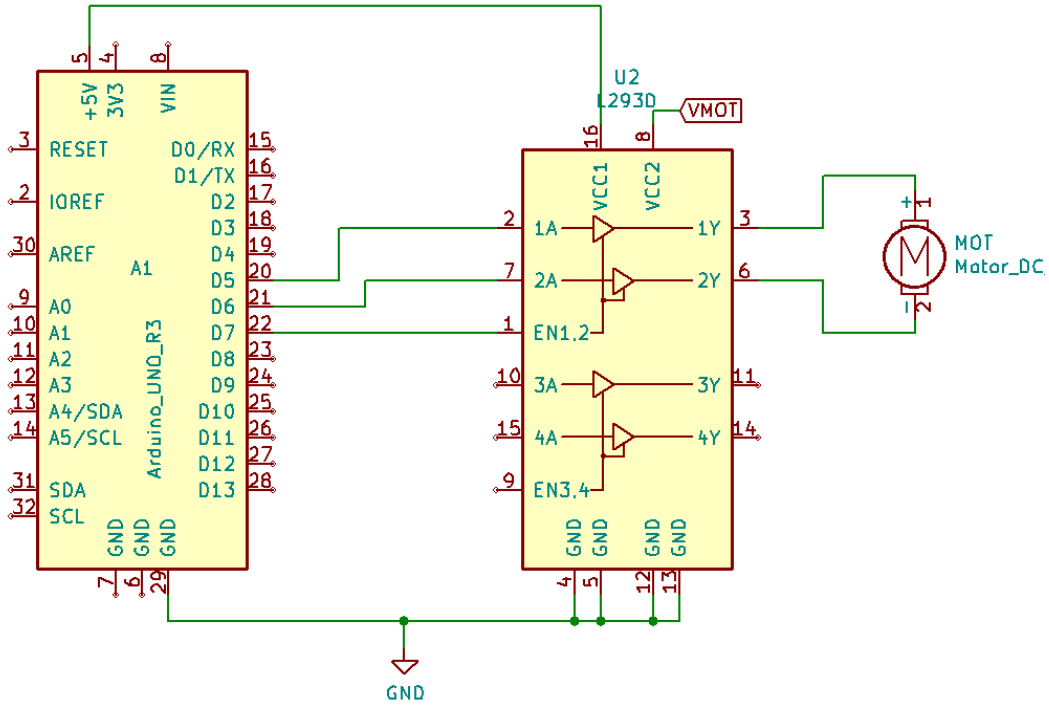

```
Fade §

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**



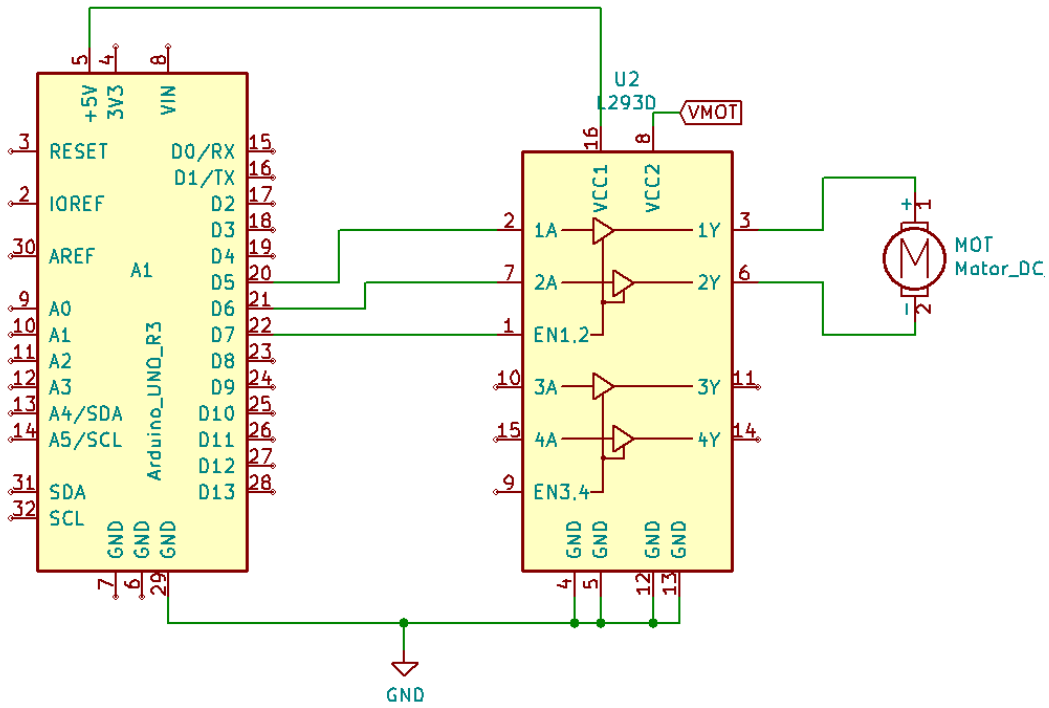


Piloter un moteur à courant continu

- Deux directions / Pont en H / **L293-D**

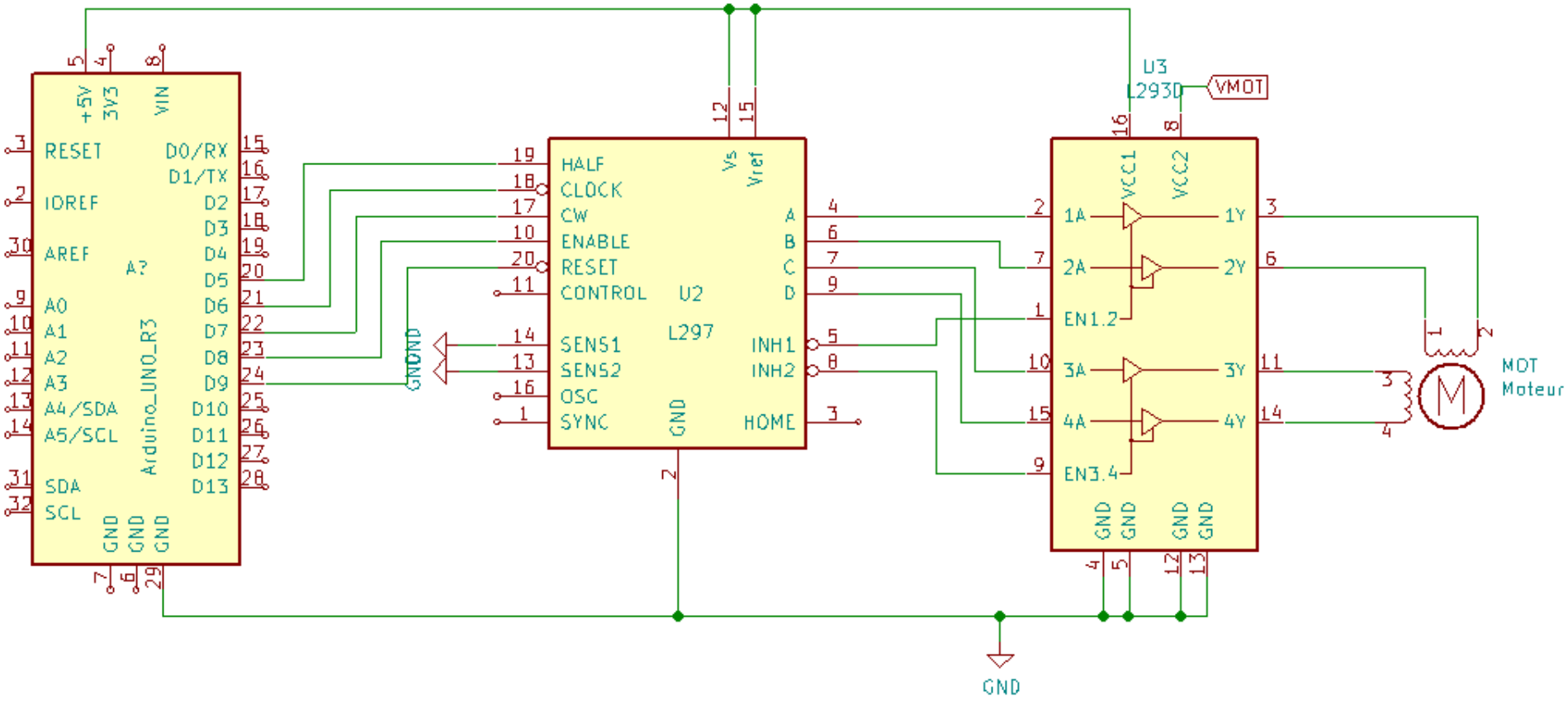
MCC8PWM

```
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

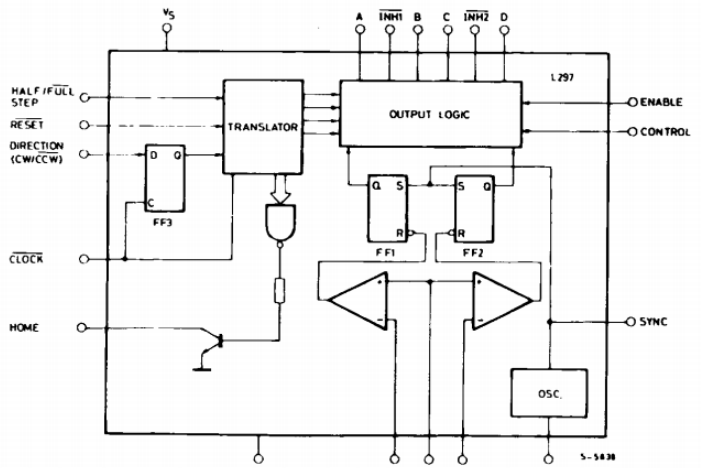
- Pont en H / **L293D**
- Driver / **L297**

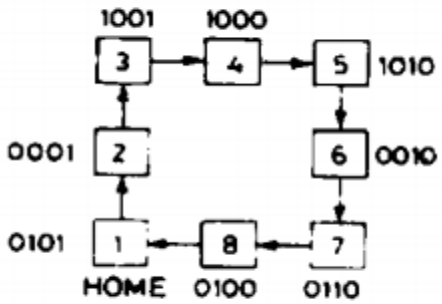


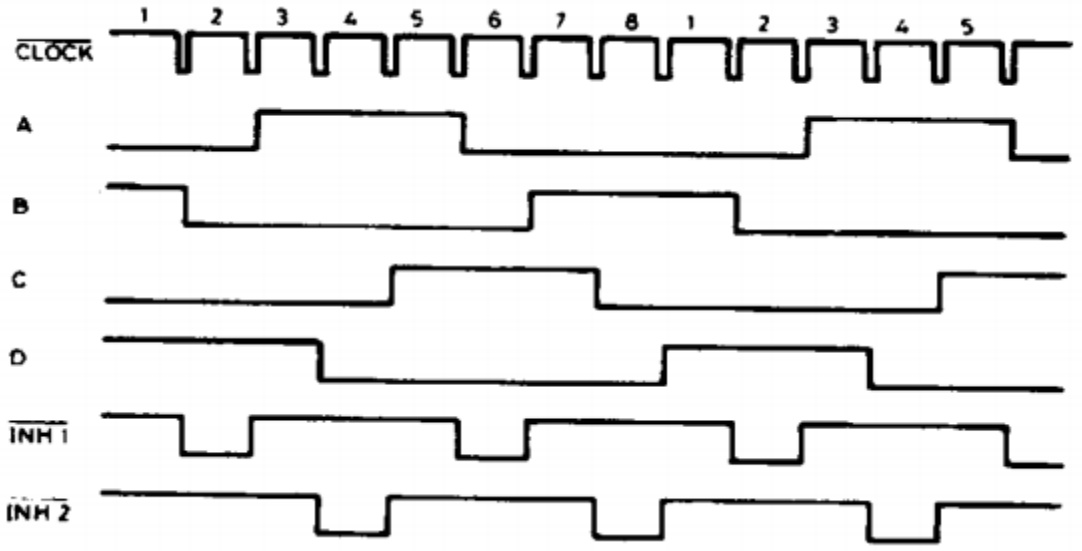
Piloter un moteur pas à pas

- Pont en H / L293D
- Driver / L297

- Commande en Demi-pas







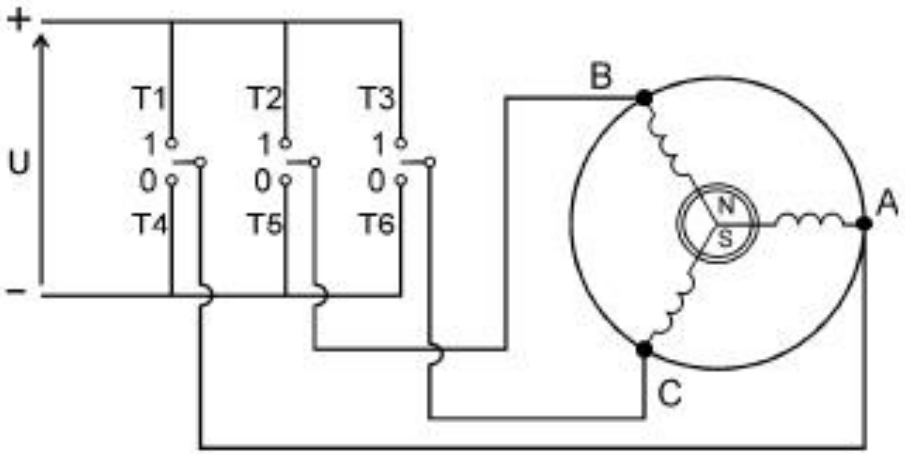
Paris-Saclay

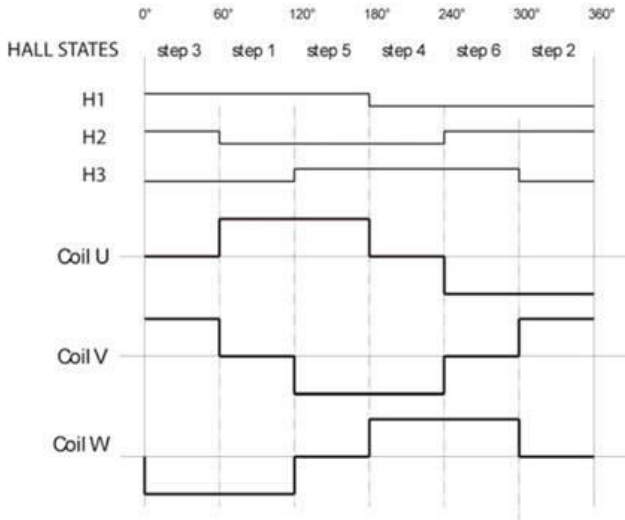
Saint-Étienne

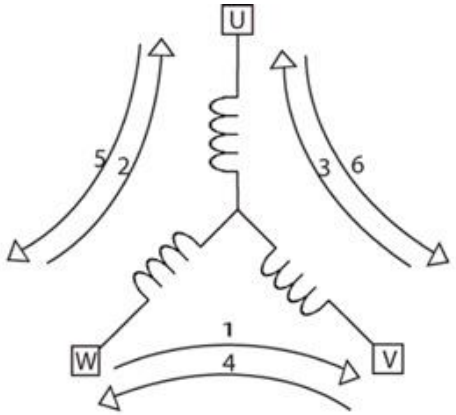
Bordeaux

Piloter un moteur *brushless*

- Transistors







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

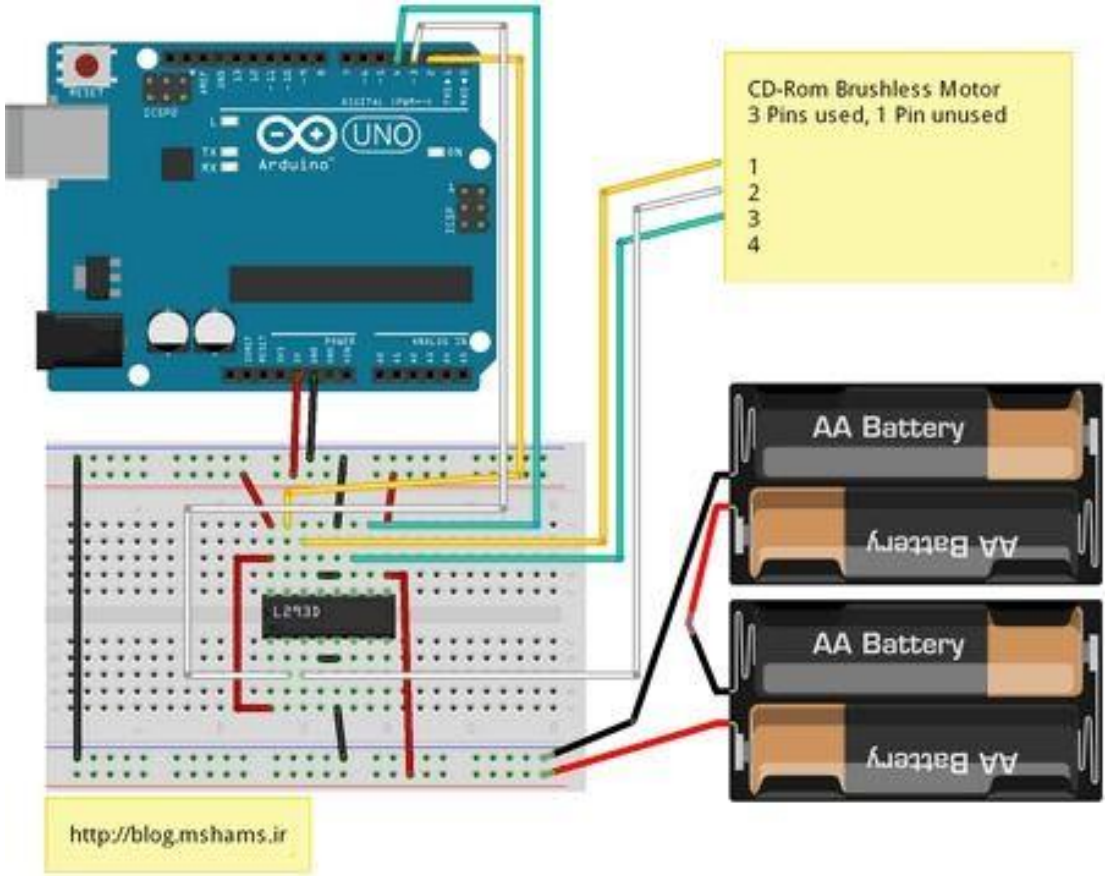
https://www.digikey.com/en/articles/techzone/2013/mar/an-introduction-to-brushless-dc-motor-control

Piloter un moteur *brushless*

- Pont en H type L293D

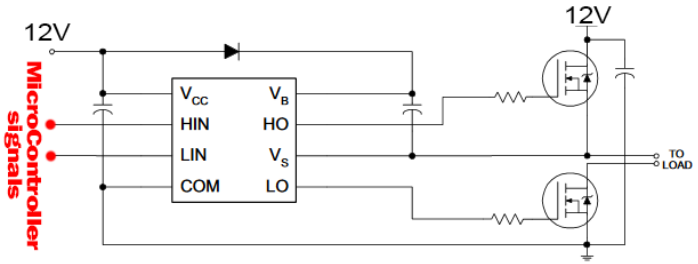
Commande_BLDC_TEST

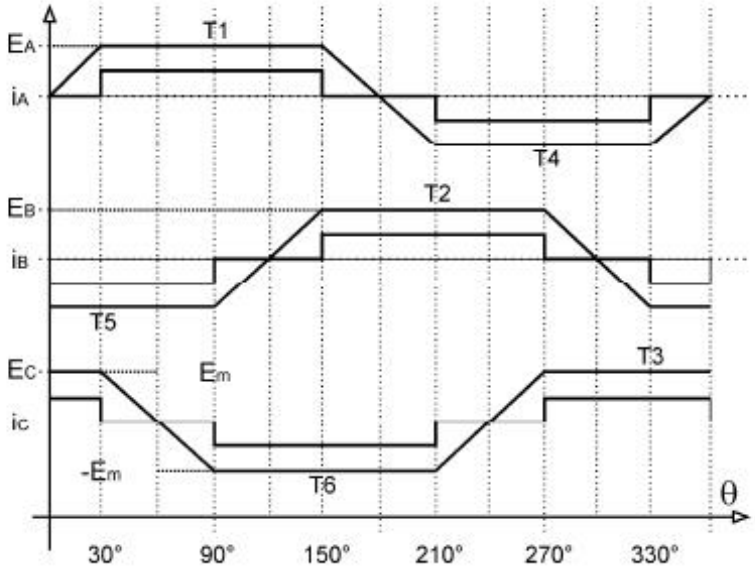
```
case 1:
  digitalWrite(phase1, HIGH);
  digitalWrite(phase2, LOW);
  digitalWrite(phase3, LOW);
  myDelay(stepLength);
  break;

case 2:
  digitalWrite(phase1, LOW);
  digitalWrite(phase2, HIGH);
  digitalWrite(phase3, LOW);
  myDelay(stepLength);
  break;

default:
  digitalWrite(phase1, LOW);
  digitalWrite(phase2, LOW);
  digitalWrite(phase3, HIGH);
  myDelay(stepLength);
  break;
```



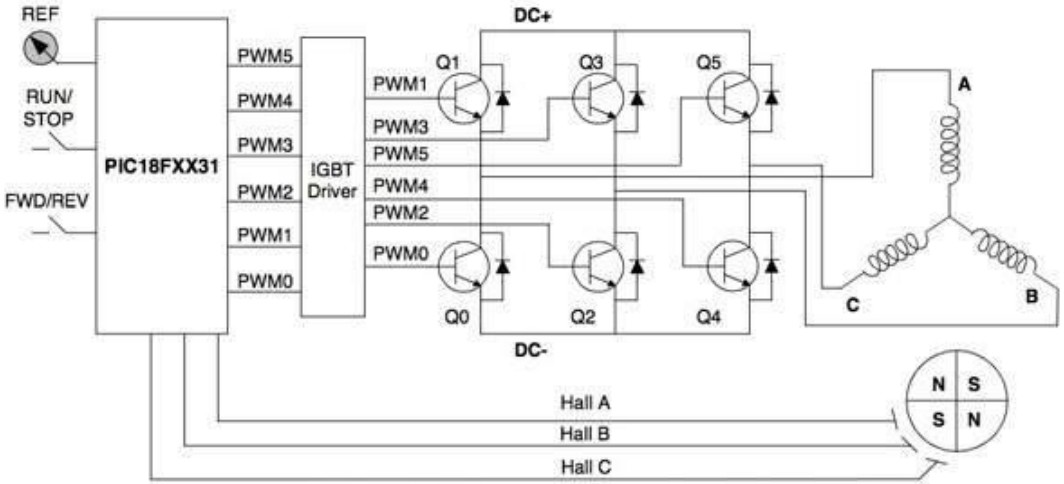
Piloter un moteur *brushless*

- Avec drivers de MOS





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



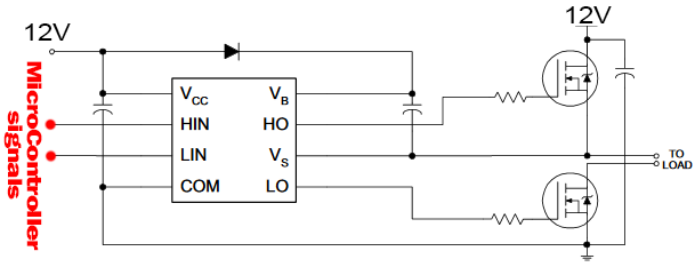
https://www.digikey.com/en/articles/techzone/2013/mar/an-introduction-to-brushless-dc-motor-control

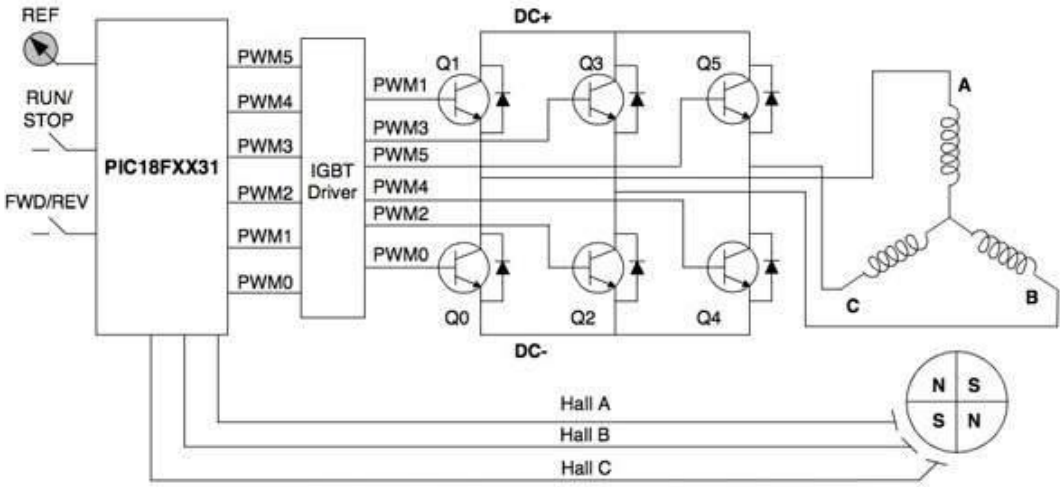
Piloter un moteur *brushless*

- Avec drivers de MOS

CommandeBLDC

```
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;
```



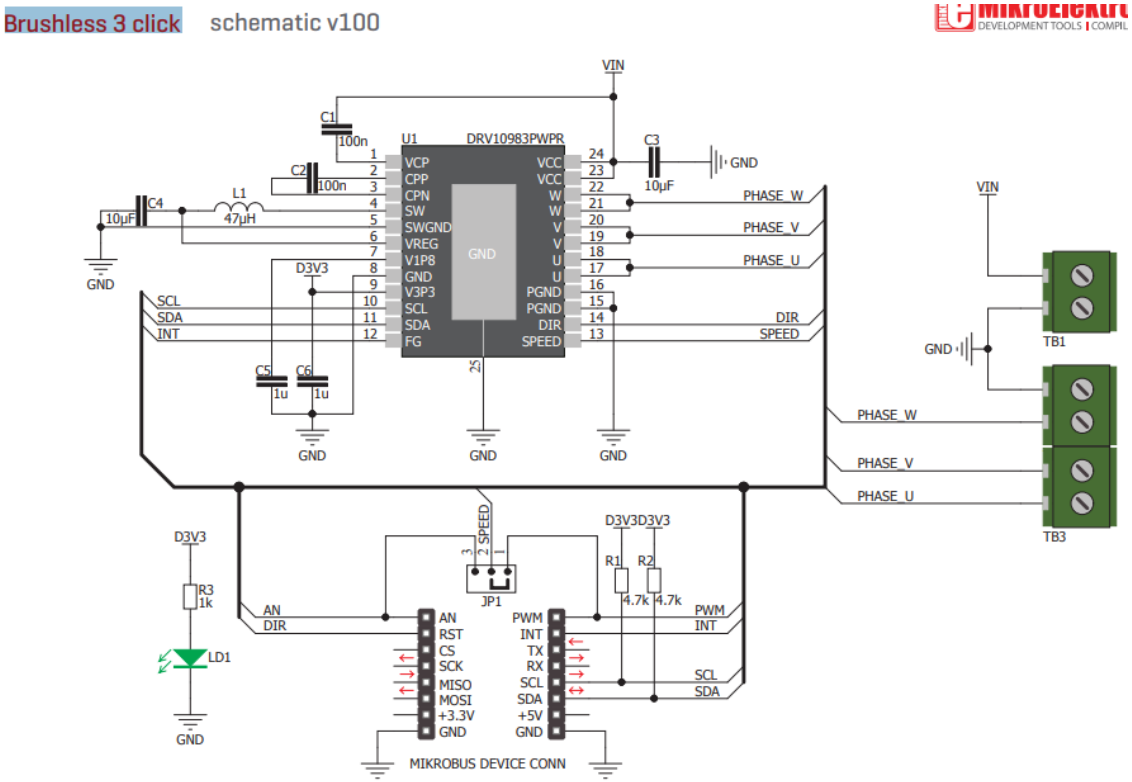


https://www.digikey.com/en/articles/techzone/2013/mar/an-introduction-to-brushless-dc-motor-control

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

Piloter un moteur *brushless*

- Utilisation du module ***Brushless 3 click*** MikroE
- Commande en **I2C**



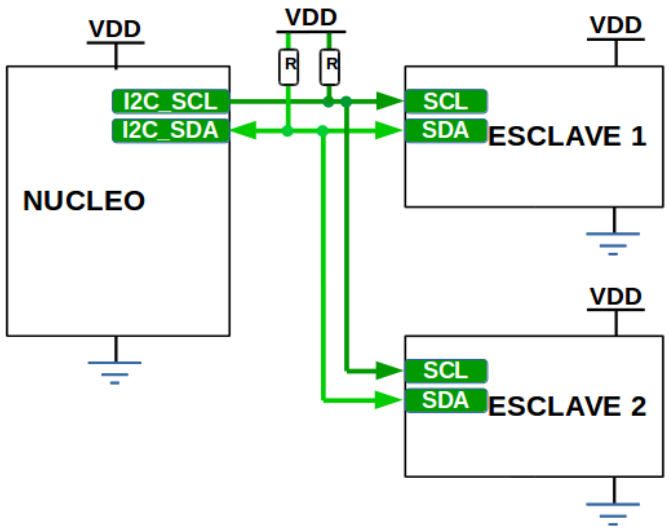
Paris-Saclay

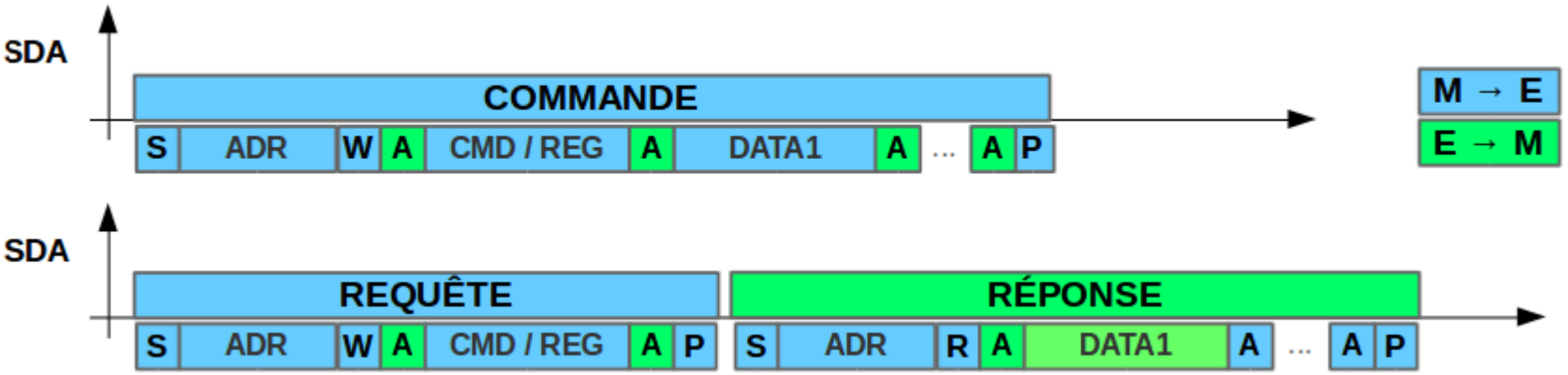
Saint-Étienne

Bordeaux

Communiquer en I2C

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





Communiquer en I2C

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

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

Command Byte Description		
Command	**Code**	**Function**
RWD	00h	Read/$\overline{\text{Write}}$ Data (DATA)
RWCR	01h	Read/$\overline{\text{Write}}$ 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
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	LSB	X	X	X	X	X	X

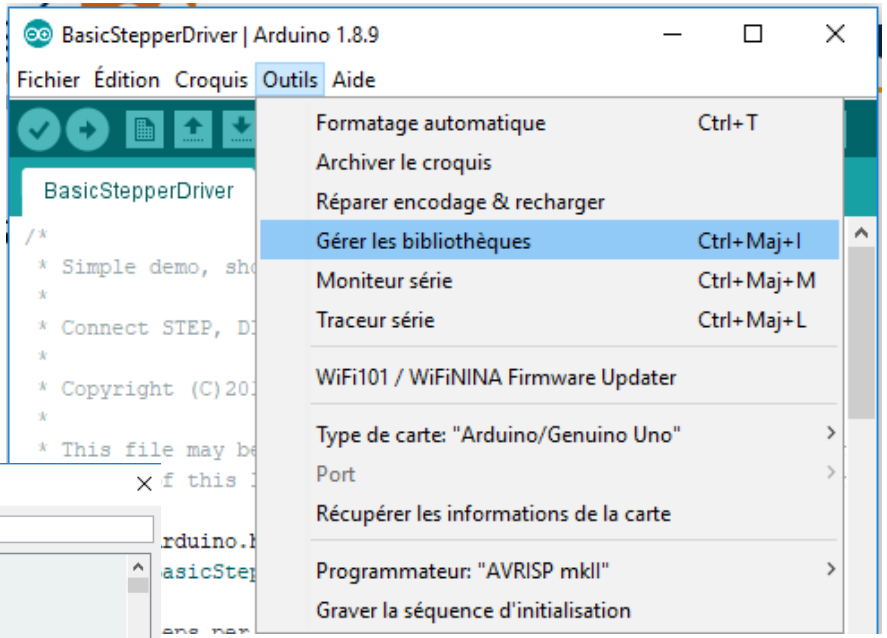
- Bibliothèque **WIRE**

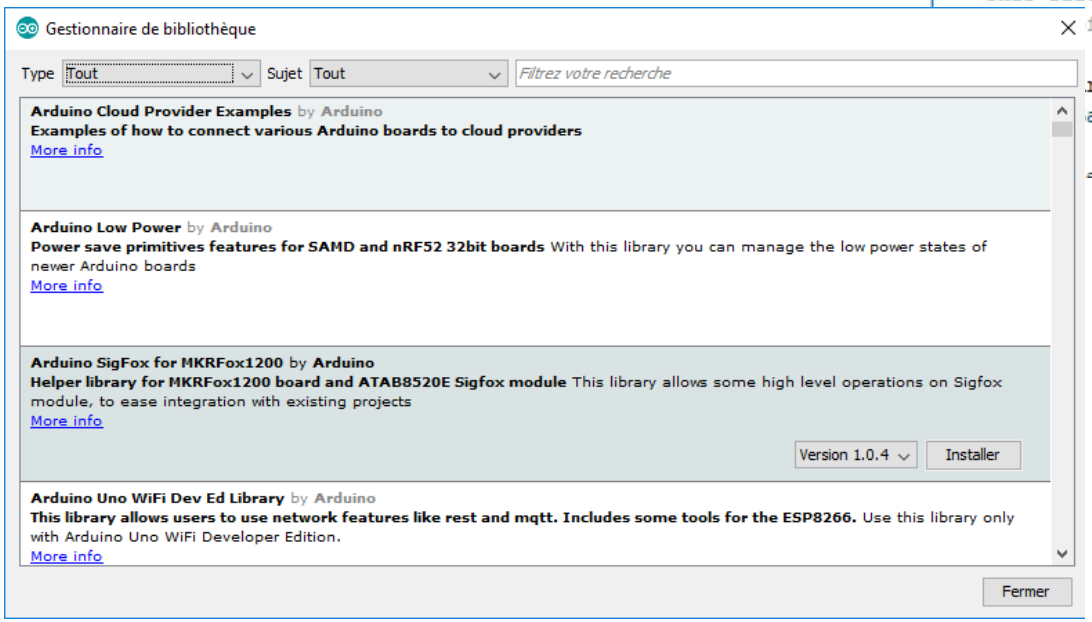
Functions

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

Des tonnes de bibliothèques !!

- Brushless Servo
- Stepper Driver / DRV8825





Paris-Saclay

Saint-Étienne

Bordeaux