

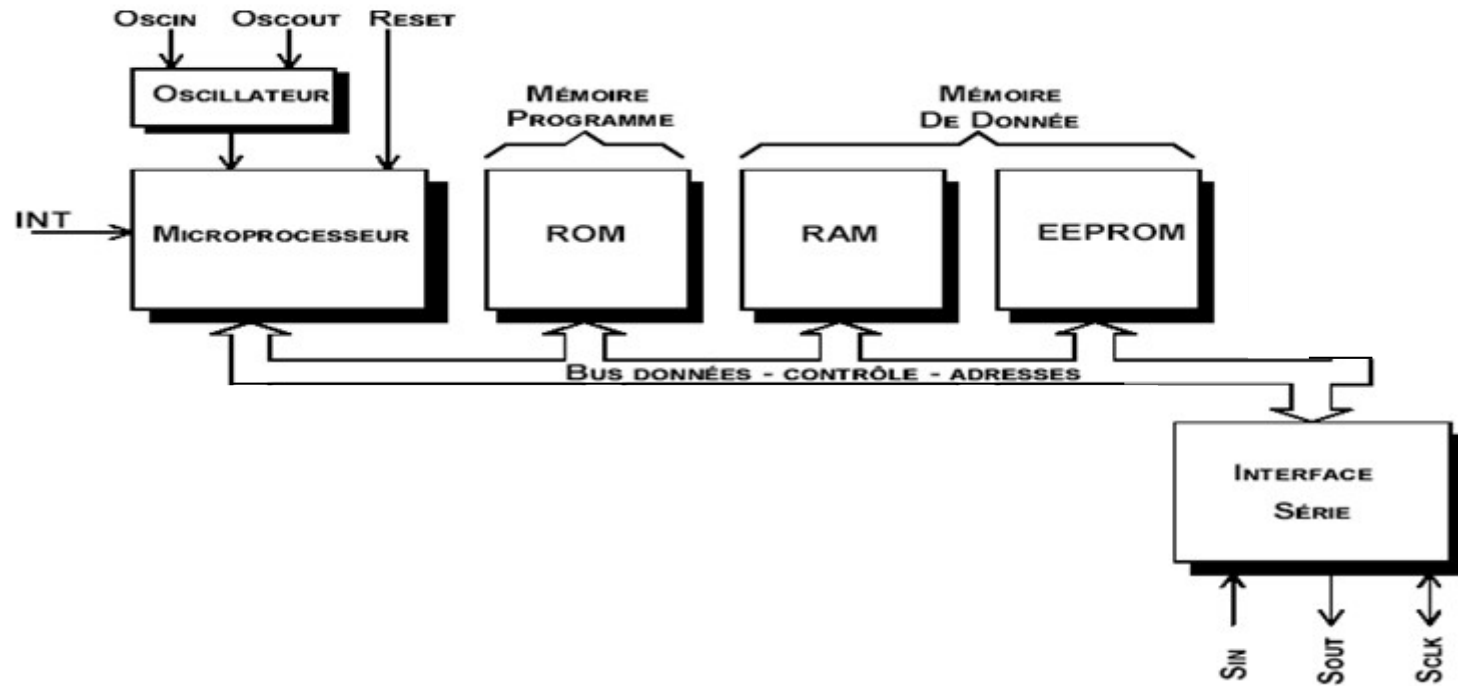
Complément Mécatronique

Halim Djerroud
hdd@ai.univ-paris8.fr

Plan

- Microcontrôleurs
- Arduino Uno (Atmega328P)
- GPIO (E/S)
- Communication Serial
- Programmation du Arduino
- Project Board
- LEDs et Résistances
- Datasheet
- Les Entrées Analogiques
- PWM
- Photorésistances
- Potentiomètres
- TMP36
- Bascules : RS, JK, D
- 74hc74
- Les registres
- TDs et Projets

Microcontrôleur



Mémoires

Type de Mémoire	ATmega 168	ATmega 328	ATmega 1280	ATmega 2560
Flash	16K octets	32K octets	128K octets	256K octets
SRAM	1024 Octets	2048 octets	8K octets	8K octets
EEPROM	512 Octets	1024 octets	4K octets	4K octets

Atmel Atmega328P

ATmega328P pin mapping

Arduino function

- reset
- digital pin 0 **RX**
- digital pin 1 **TX**
- digital pin 2
- digital pin 3 **PWM**
- digital pin 4
- VCC
- GND
- crystal
- crystal
- digital pin 5 **PWM**
- digital pin 6 **PWM**
- digital pin 7
- digital pin 8

- PC6 1
- PD0 2
- PD1 3
- PD2 4
- PD3 5
- PD4 6
- VCC 7
- GND 8
- PB6 9
- PB7 10
- PD5 11
- PD6 12
- PD7 13
- PB0 14



- 28 PC5
- 27 PC4
- 26 PC3
- 25 PC2
- 24 PC1
- 23 PC0
- 22 GND
- 21 AREF
- 20 AVCC
- 19 PB5
- 18 PB4
- 17 PB3
- 16 PB2
- 15 PB1

Arduino function

- analog input 5
- analog input 4
- analog input 3
- analog input 2
- analog input 1
- analog input 0
- GND
- analog reference
- AVCC
- digital pin 13
- digital pin 12
- PWM** digital pin 11
- PWM** digital pin 10
- PWM** digital pin 9

When using ISP to program the chip

- SCK**
- MISO**
- MOSI**

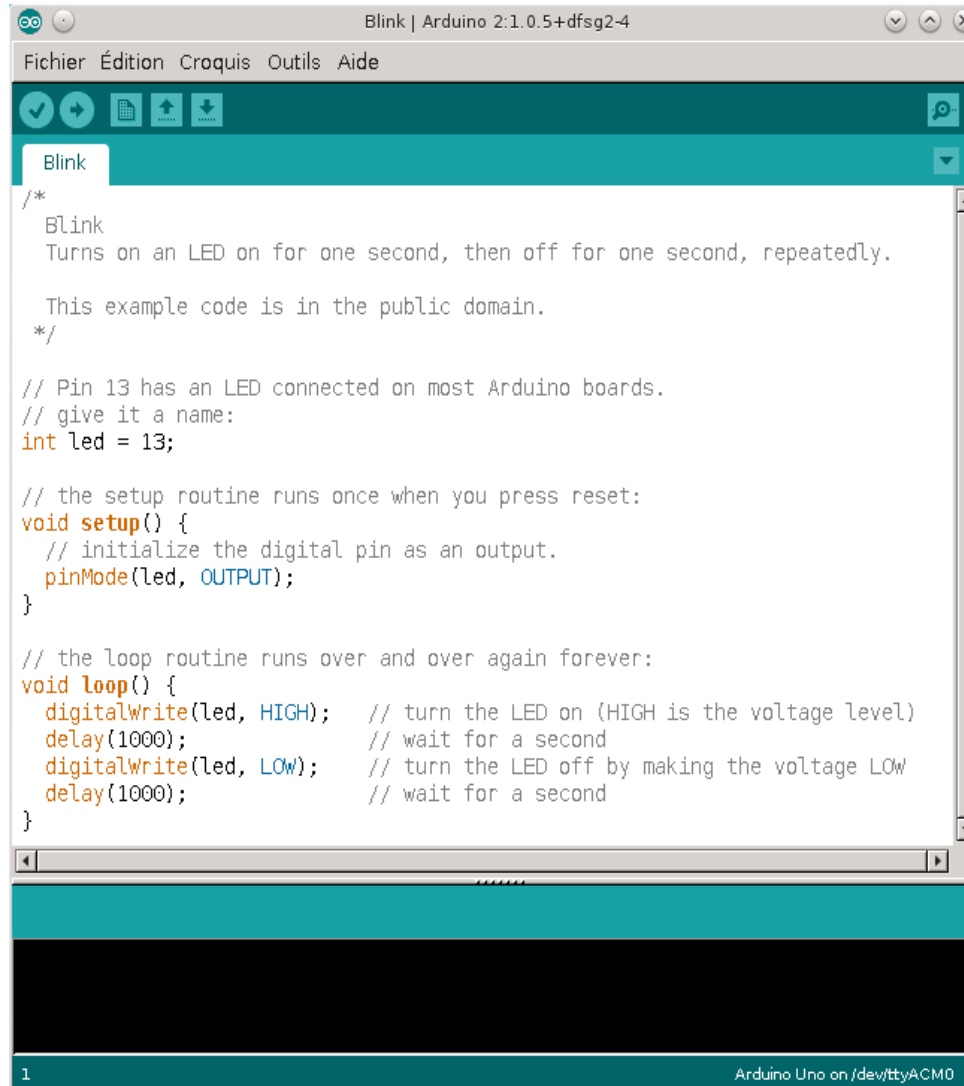
Arduino

- Arduino est un circuit imprimé en matériel libre
 - les plans de la carte elle-même sont publiés en licence libre,
 - certains composants de la carte, comme le microcontrôleur par exemple, ne sont pas en licence libre
- Un module Arduino est généralement construit autour d'un microcontrôleur Atmel AVR
- Le logiciel de programmation des modules Arduino est une application Java, libre et multi-plateforme
 - éditer de code
 - Compiler
 - transférer le firmware et le programme au travers d'une liaison série

Arduino Uno



Arduino



The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 2:1.0.5+dfsg2-4". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for saving, undo, redo, and other editing functions. The main text area displays the following code:

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

  This example code is in the public domain.
  */

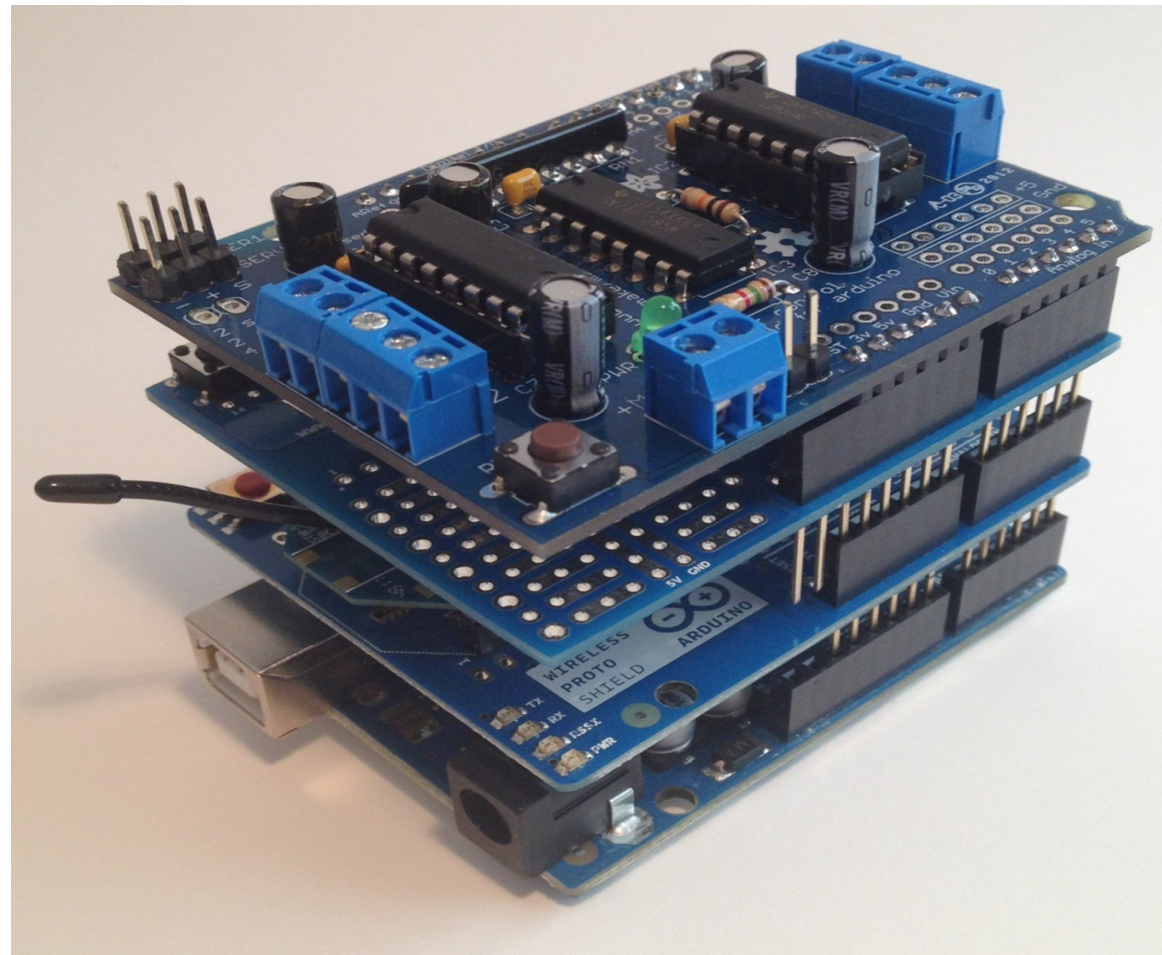
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

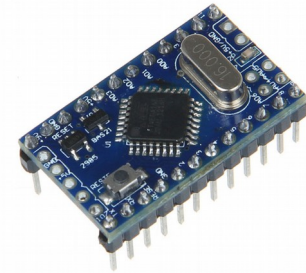
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
```

At the bottom of the window, the status bar shows "1" on the left and "Arduino Uno on /dev/ttyACM0" on the right.

Les modules



Arduino – Autres types



Alimentation Arduino

- Tension de fonctionnement
 - 5 V
 - 3,3 V
- Alimentation externe
- le courant de sortie de chaque broche (D0 à D13) ne doit pas dépasser 40mA
- le courant issu du port USB ne doit pas dépasser 500mA
- le courant soutiré à la broche «3,3V» ne doit pas dépasser 50mA

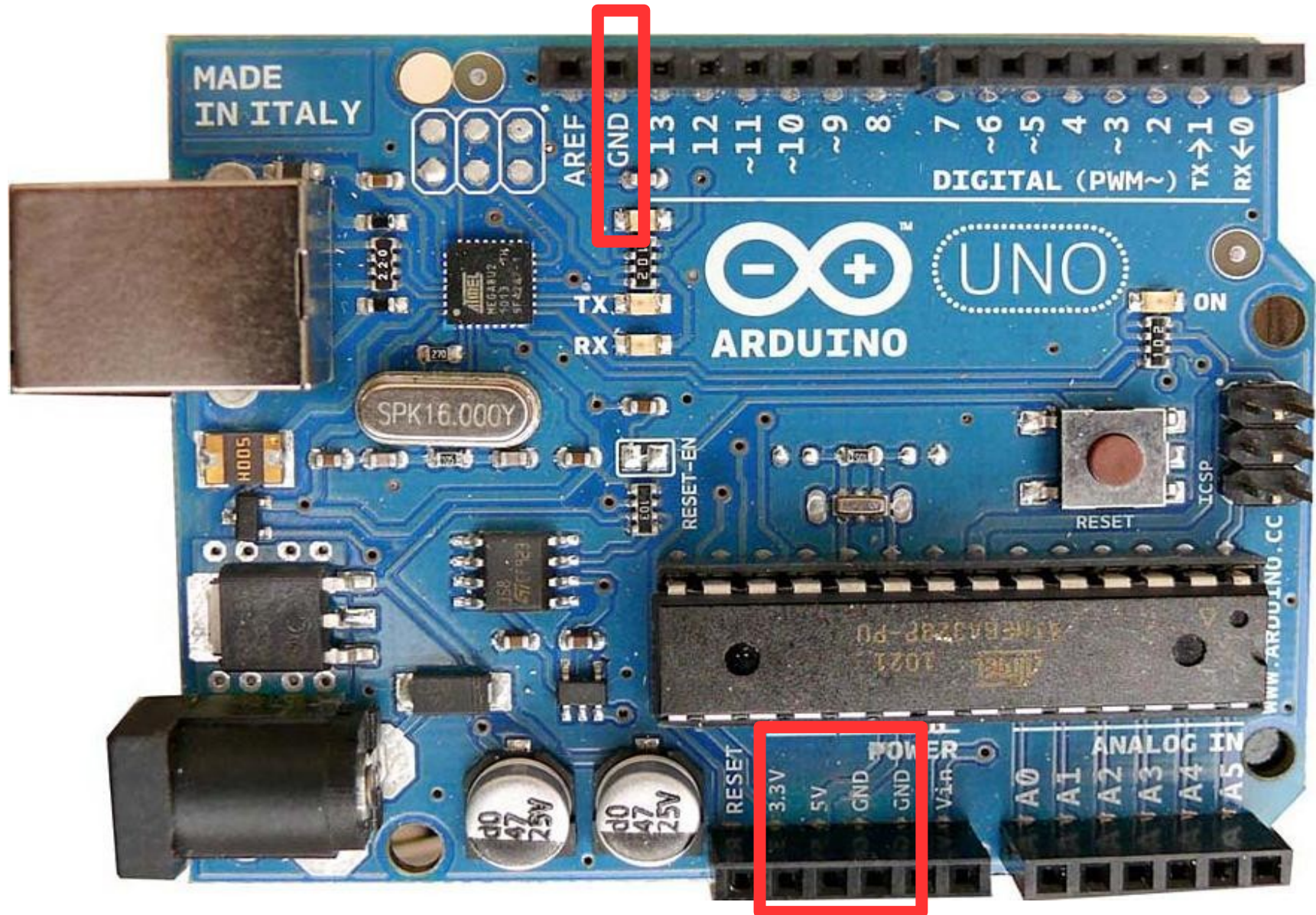
Exemple

On souhaite actionner un moteur de 6W sous 5V par l'Arduino, alimenté par l'USB, est-ce réalisable ?

$$P = U.I \quad \Rightarrow \quad I = P/U \quad \Rightarrow \quad I = 1,2A$$

- P : puissance (W)
- U : tension (V)
- I : courant (A)

Alimentation



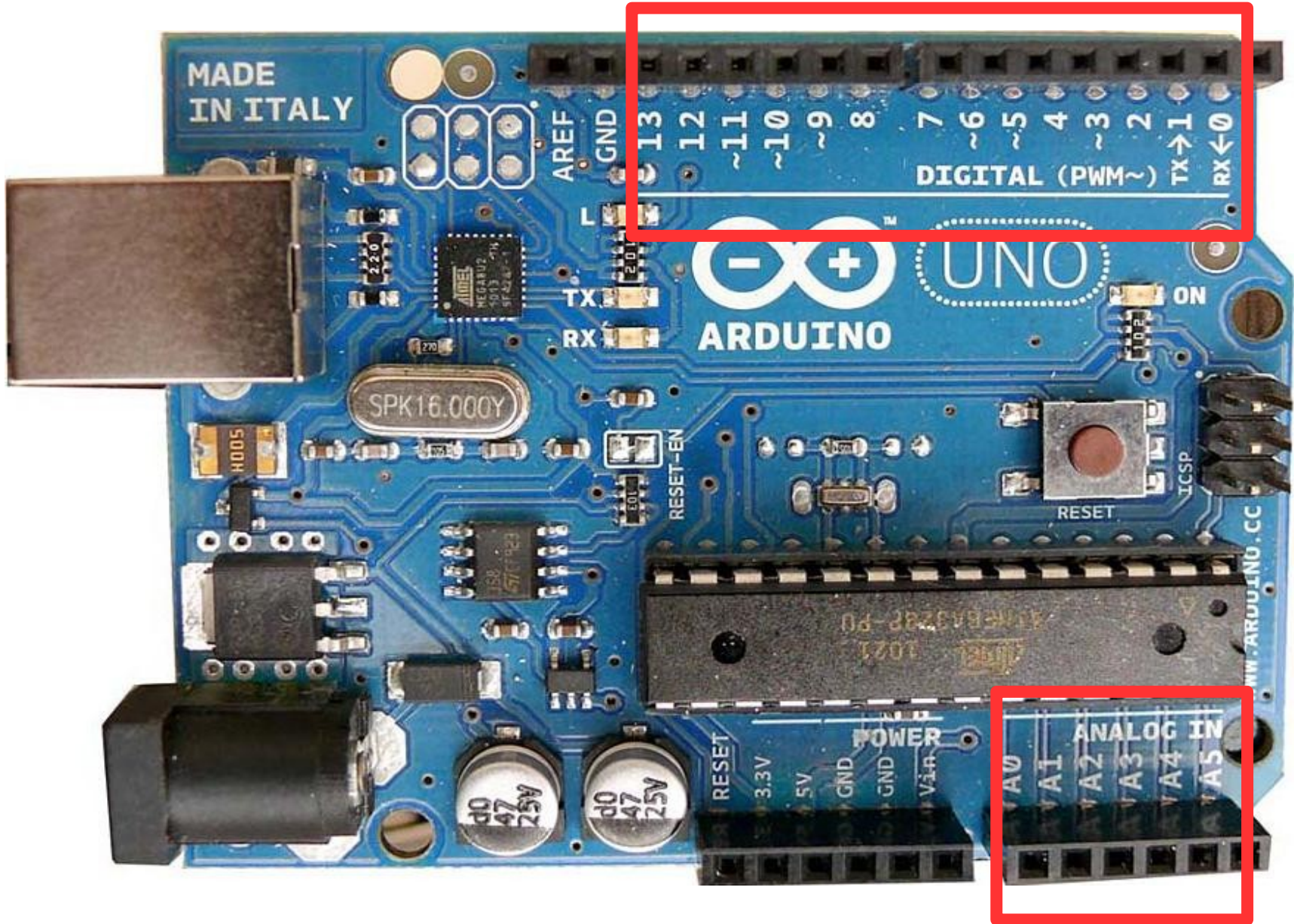
Programme

```
// Le plus petit programme Arduino  
void setup() {  
}  
  
void loop() {  
}
```

General Purpose Input/Output GPIO

- Entrée/Sortie pour un Usage Général
- La possibilité de communiquer avec d'autres circuits électroniques
- Un connecteur GPIO ne peut traiter que des signaux numérique, 1 ou 0
- Un connecteur GPIO peut être configuré :
 - Entrée (réception d'un signal)
 - Sortie (émission d'un signal)

GPIO



GPIO

- `pinMode(pin, mode) ;`
- `digitalWrite(ledPin, HIGH);`
- `digitalRead(pin) ;`

GPIO

```
int ledPin = 13;           // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()
{
  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
}
```

GPIO

```
int ledPin = 13; // LED connected to digital pin 13
int inPin = 7;   // pushbutton connected to digital pin 7
int val = 0;    // variable to store the read value

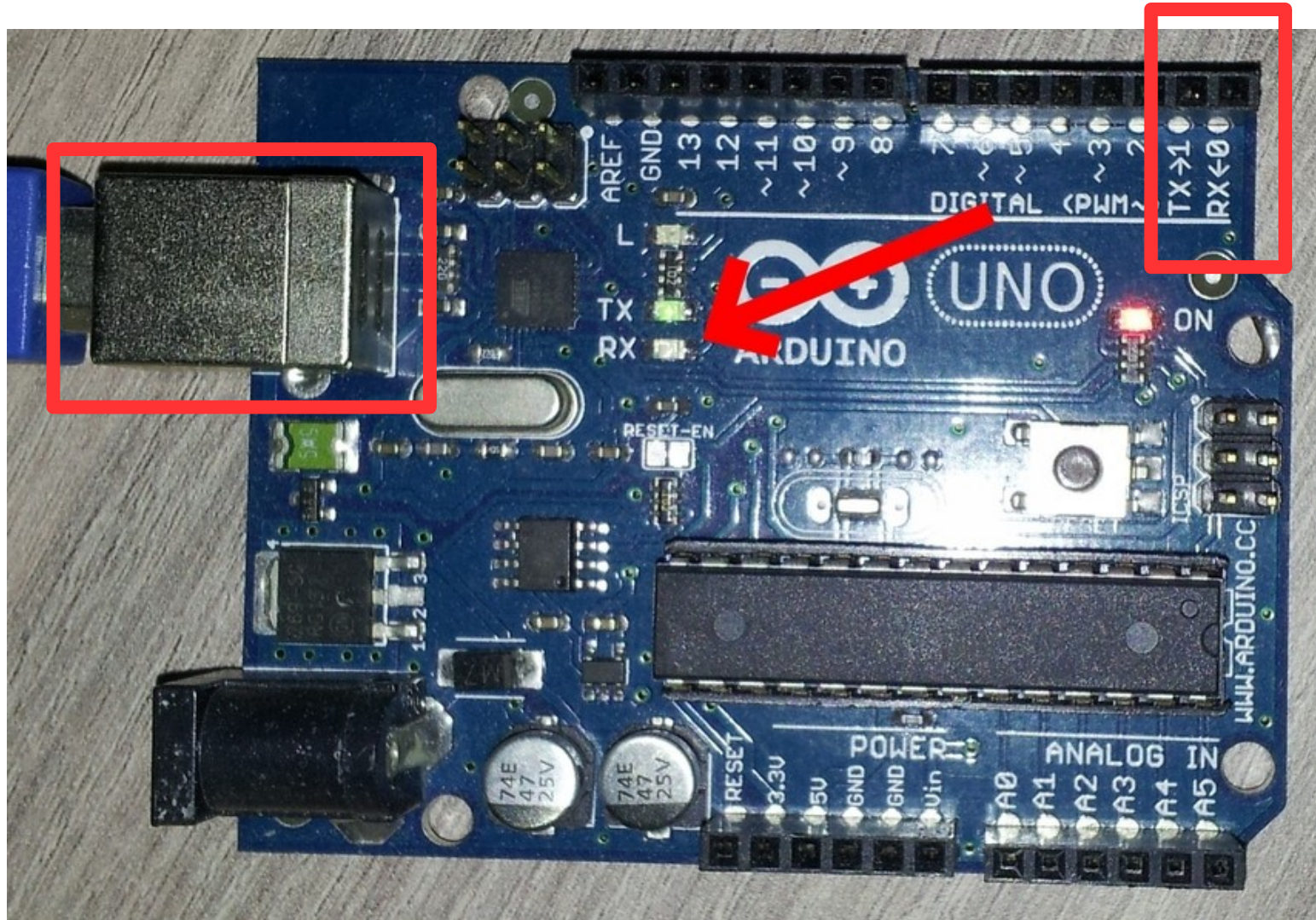
void setup()
{
    pinMode(ledPin, OUTPUT); // sets the digital pin 13 as output
    pinMode(inPin, INPUT);   // sets the digital pin 7 as input
}

void loop()
{
    val = digitalRead(inPin); // read the input pin
    digitalWrite(ledPin, val); // sets the LED to the button's value
}
```

Communication Série (Serial)

- En communication série, on découpe l'information à transmettre en petits blocs de taille fixe avant de la transmettre.
- S'oppose à la communication parallèle
- L'avantage de la communication série sur la communication parallèle est qu'elle nécessite moins de lignes

Communication Série (Serial)



Communication Série (Serial)

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.println("Coucou");  
}
```

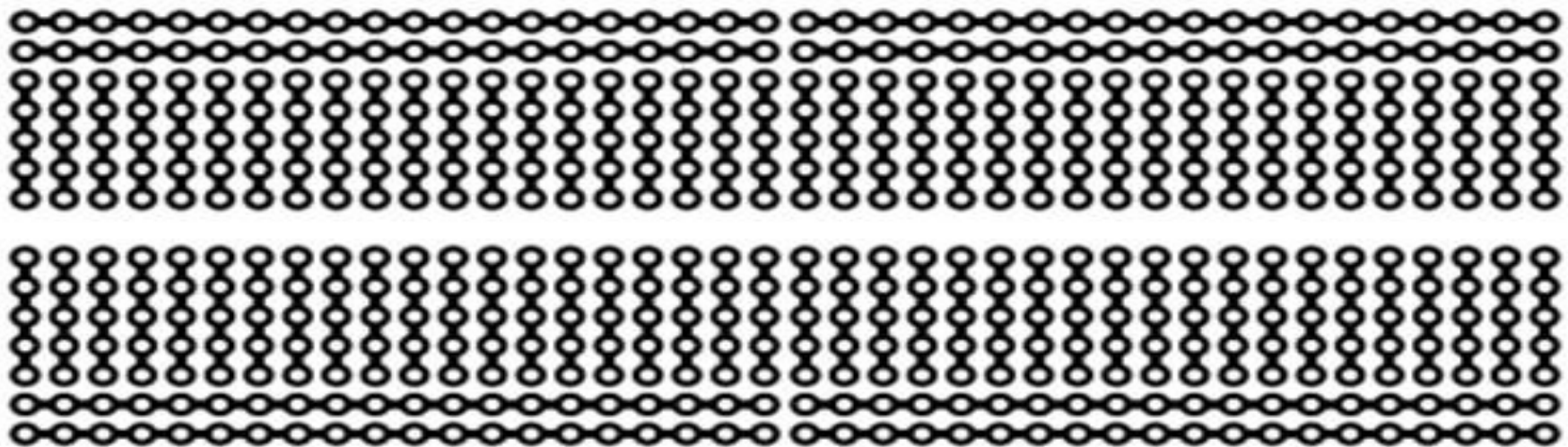
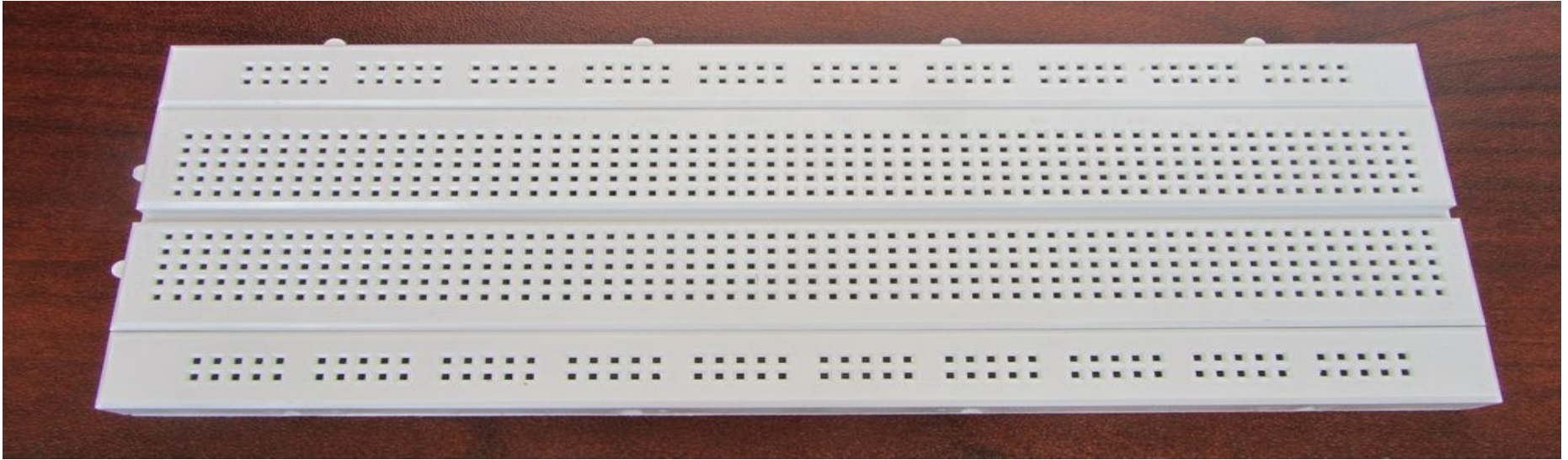
Communication Serial

```
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
    if ( Serial.available() ) {  
        int lu = Serial.read();  
        Serial.println(lu);  
    } else {  
        Serial.println("Rien");  
    }  
    delay(2000);  
}
```

Communication Série (Serial)



Project Board



Project Board

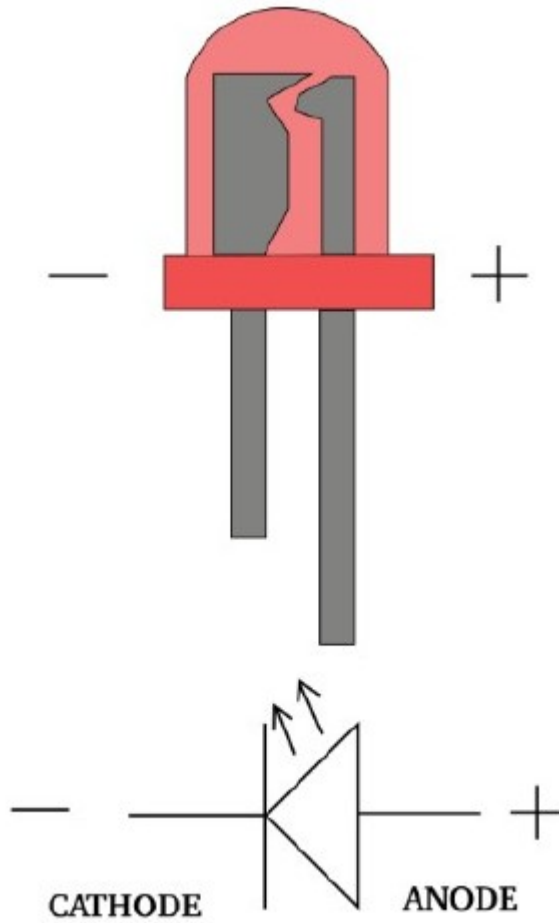
Demo fritzing

LEDs

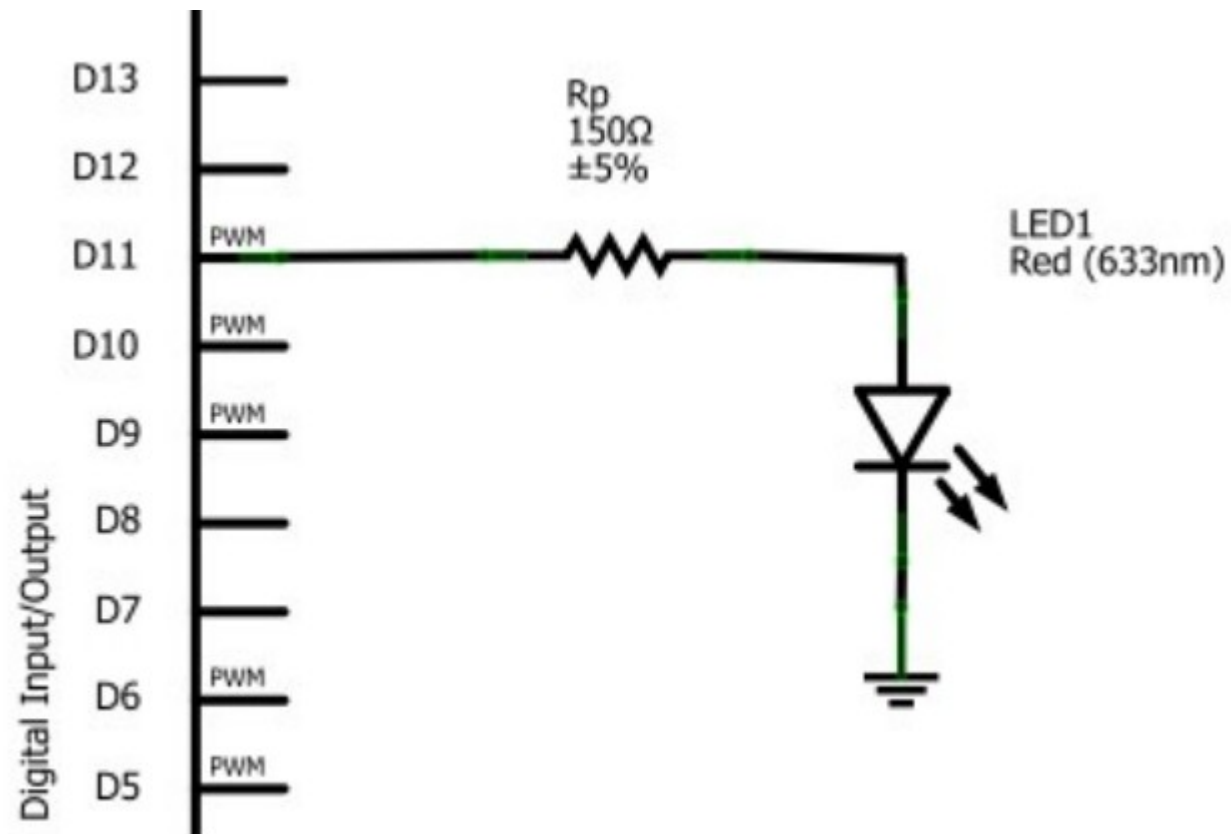
- Diode électroluminescente (DEL ou LED)
- Une diode électroluminescente ne laisse passer le courant électrique que dans un seul sens (le sens passant).
 - 1,9V pour une LED rouge,
 - 3,2V pour les diodes blanches, ou autres couleurs
 - voir les notices des diodes avant d'utiliser
 - Le courant dans la LED est aussi limité pour les LED ordinaires de 5mm, 24mA environ.



LEDs



LEDs

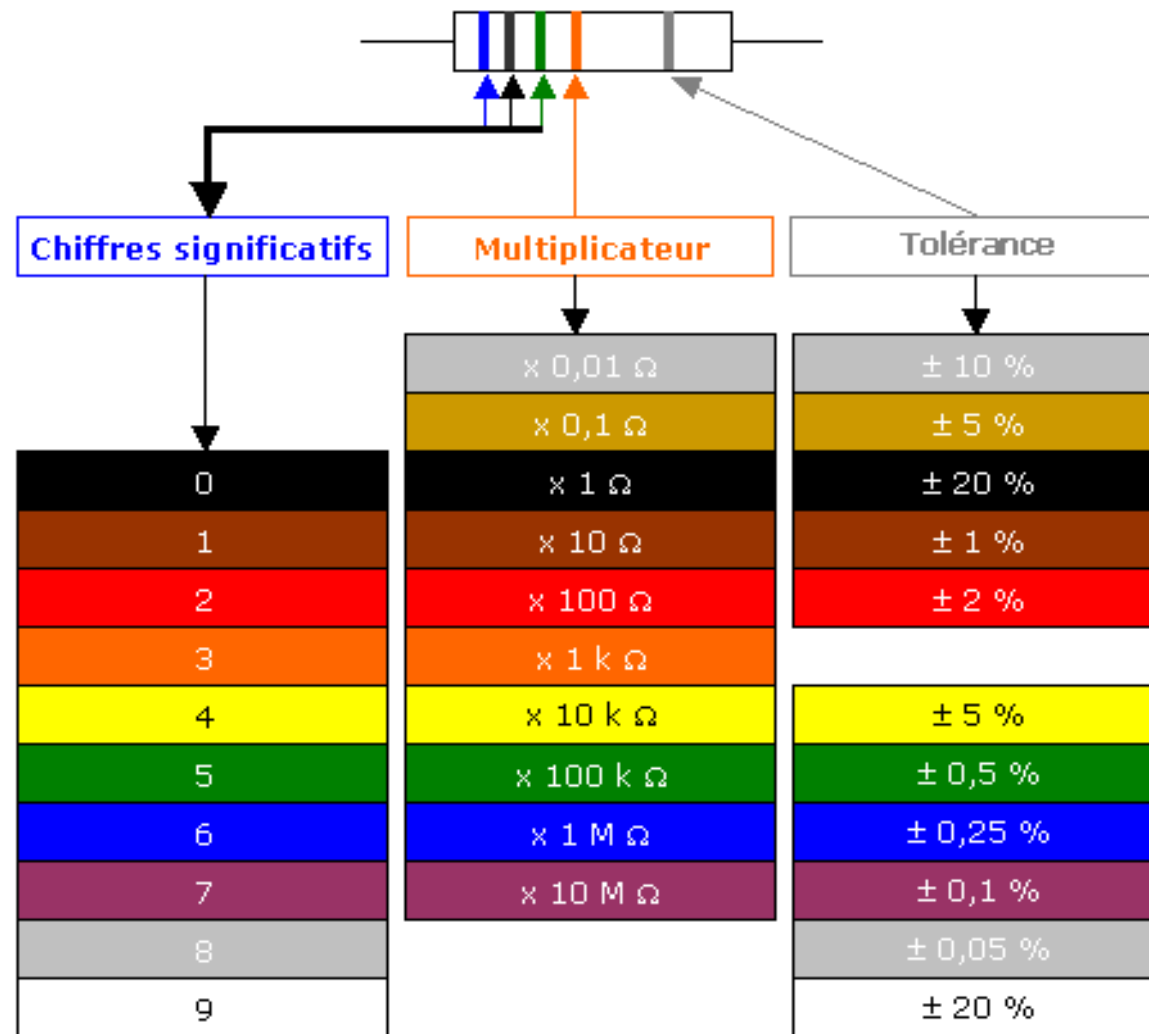


Résistances

- $U = R I$
 - U : tension (V)
 - R : résistance (Ohm - Ω)
 - I : courant (A)



Résistances



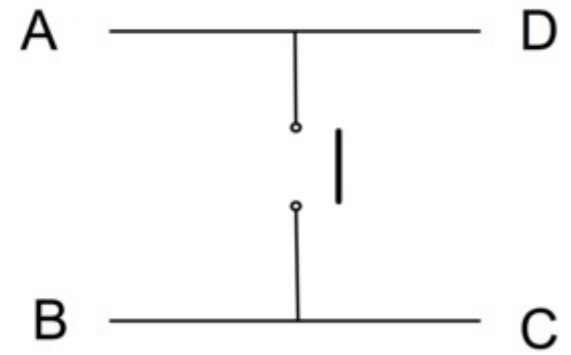
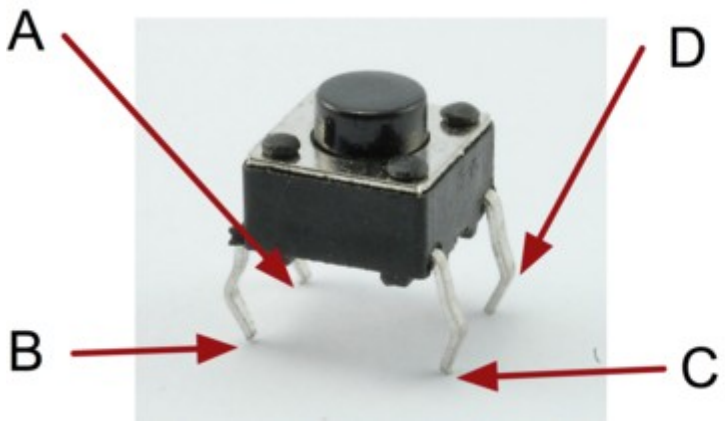
Datasheet

Demo

Led + résistance

Demo fritzing

Switch



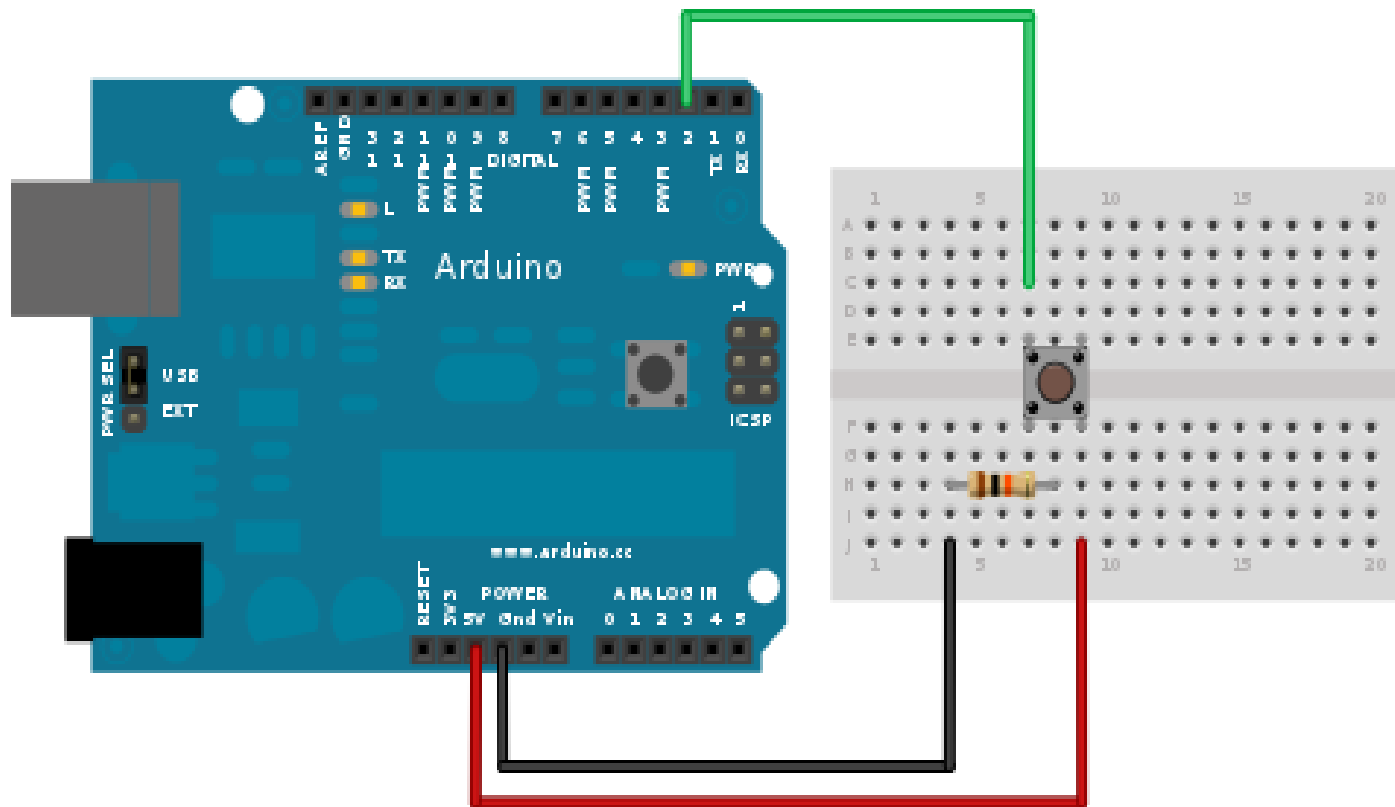
Switch

```
int ledPin = 13; // LED connected to digital pin 13
int inPin = 7;   // pushbutton connected to digital pin 7
int val = 0;    // variable to store the read value

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin 13 as output
  pinMode(inPin, INPUT);   // sets the digital pin 7 as input
}

void loop()
{
  val = digitalRead(inPin); // read the input pin
  digitalWrite(ledPin, val); // sets the LED to the button's value
}
```

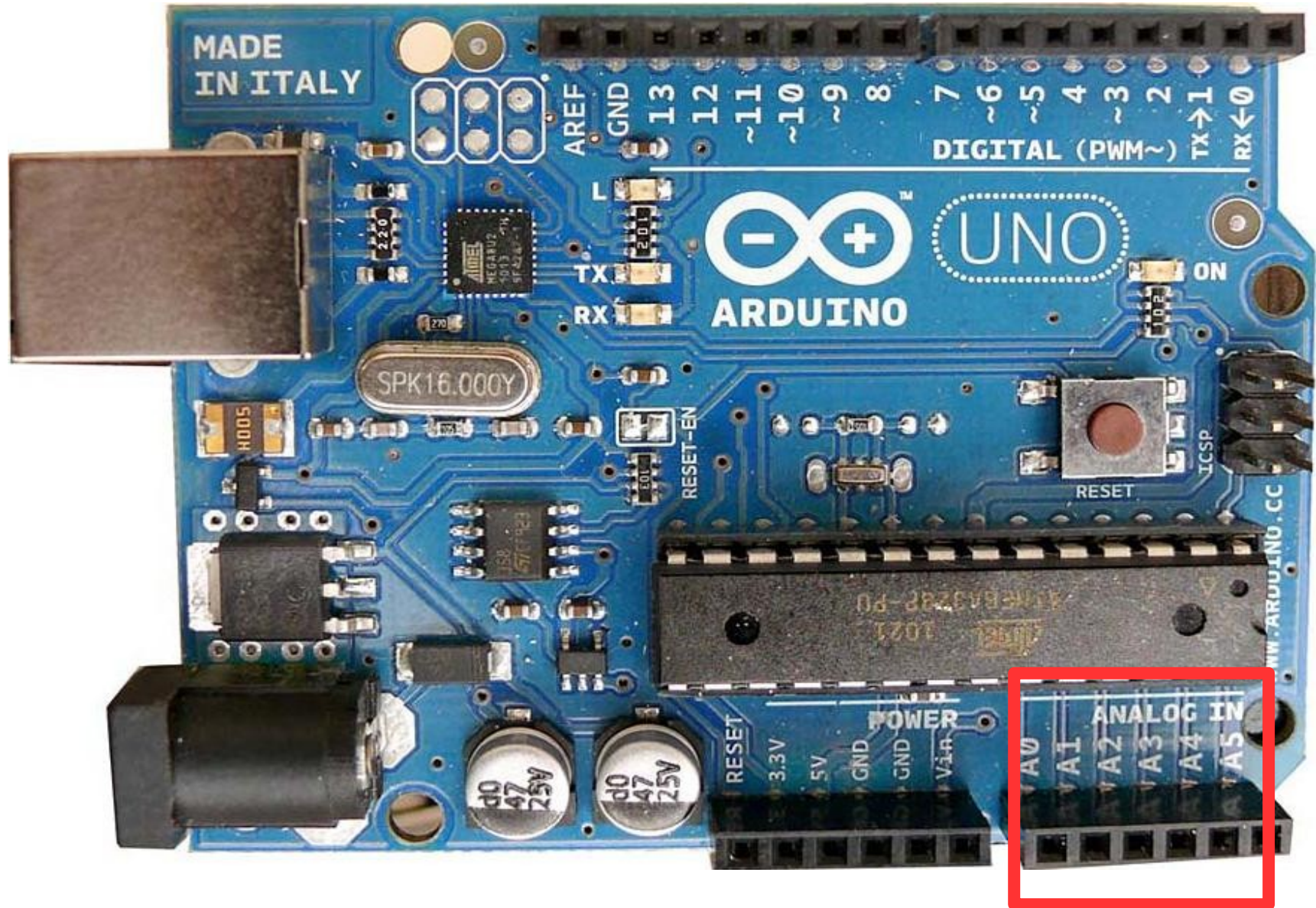
Résistance de tirage



Les Entrées Analogiques

- Lire des valeurs différentes de 0, 1
- # GPIO
- Lire une valeur entre 0 et 1023
- Entrée entre 0v et 5v

Les Entrées Analogiques



Les Entrées Analogiques

```
int analogPin = A0;
int val = 0;

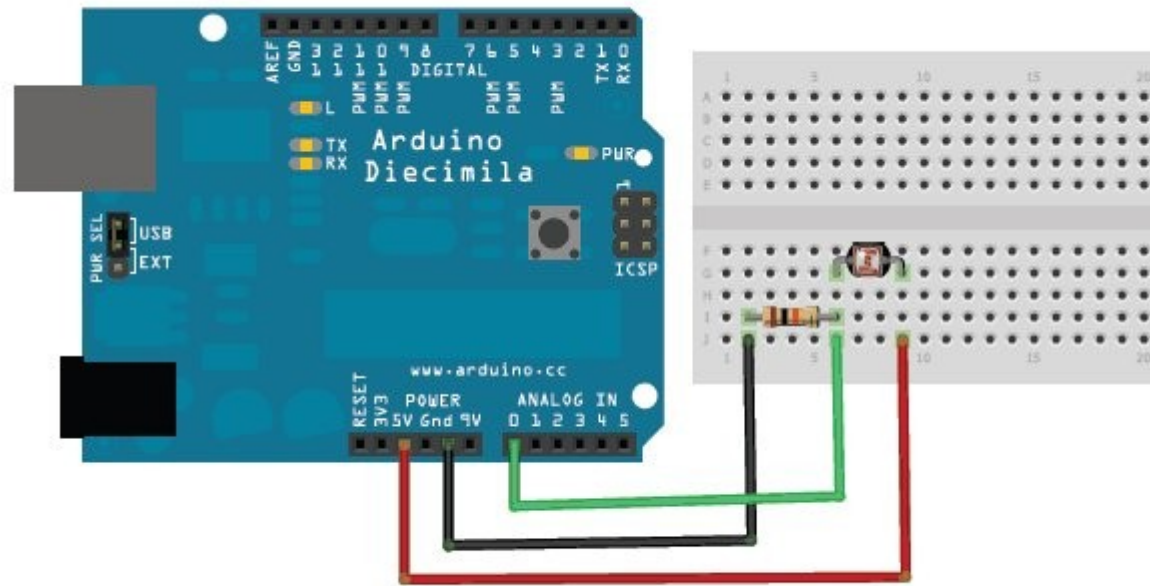
void setup() {
    Serial.begin(9600);
}

void loop() {
    val = analogRead(analogPin);
    Serial.println(val);
}
```

Photorésistances

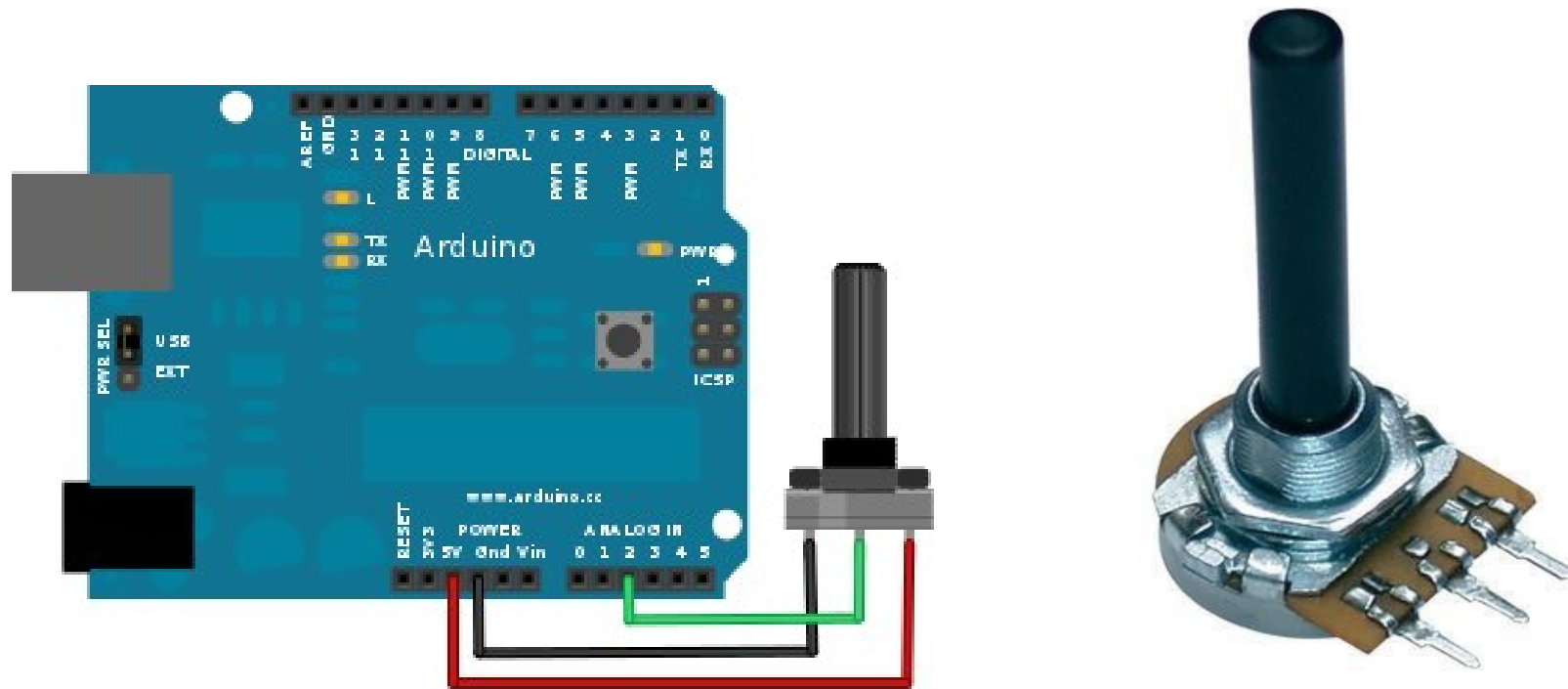


Photorésistances



Potentiomètres

- Résistance variable



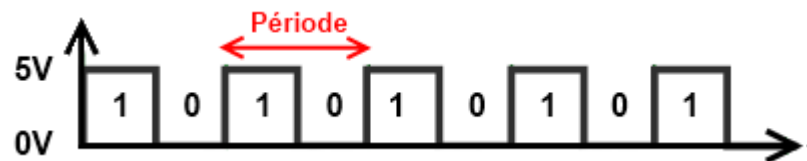
Made with  Fritzing.org

Les Sorties Analogiques ou presque

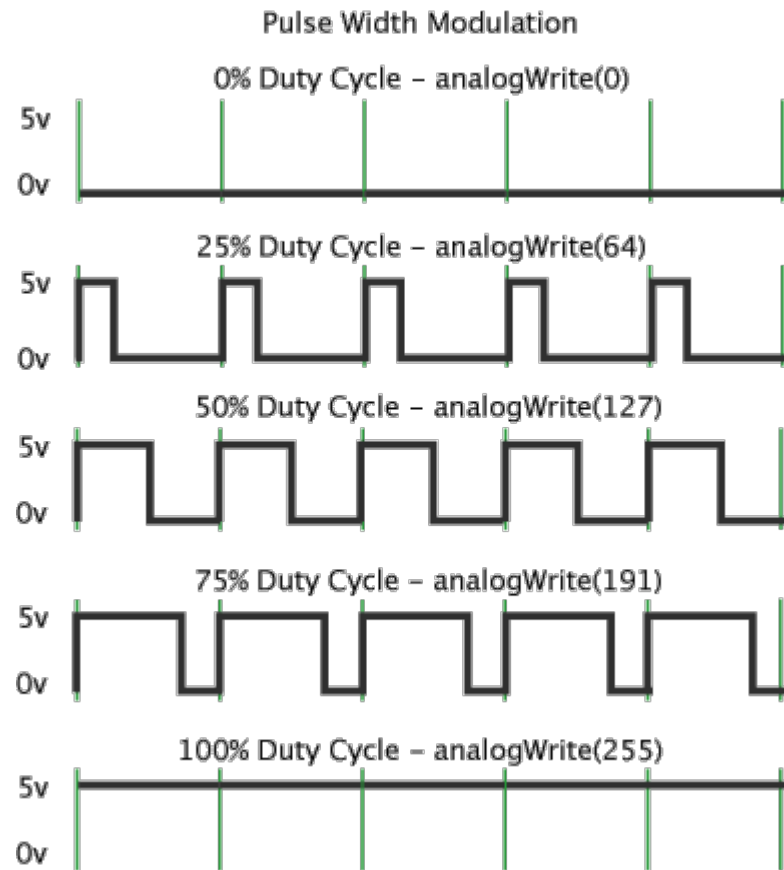
- Convertir des données binaires en signal analogique
- Comment faire ?

PWM ou MLI

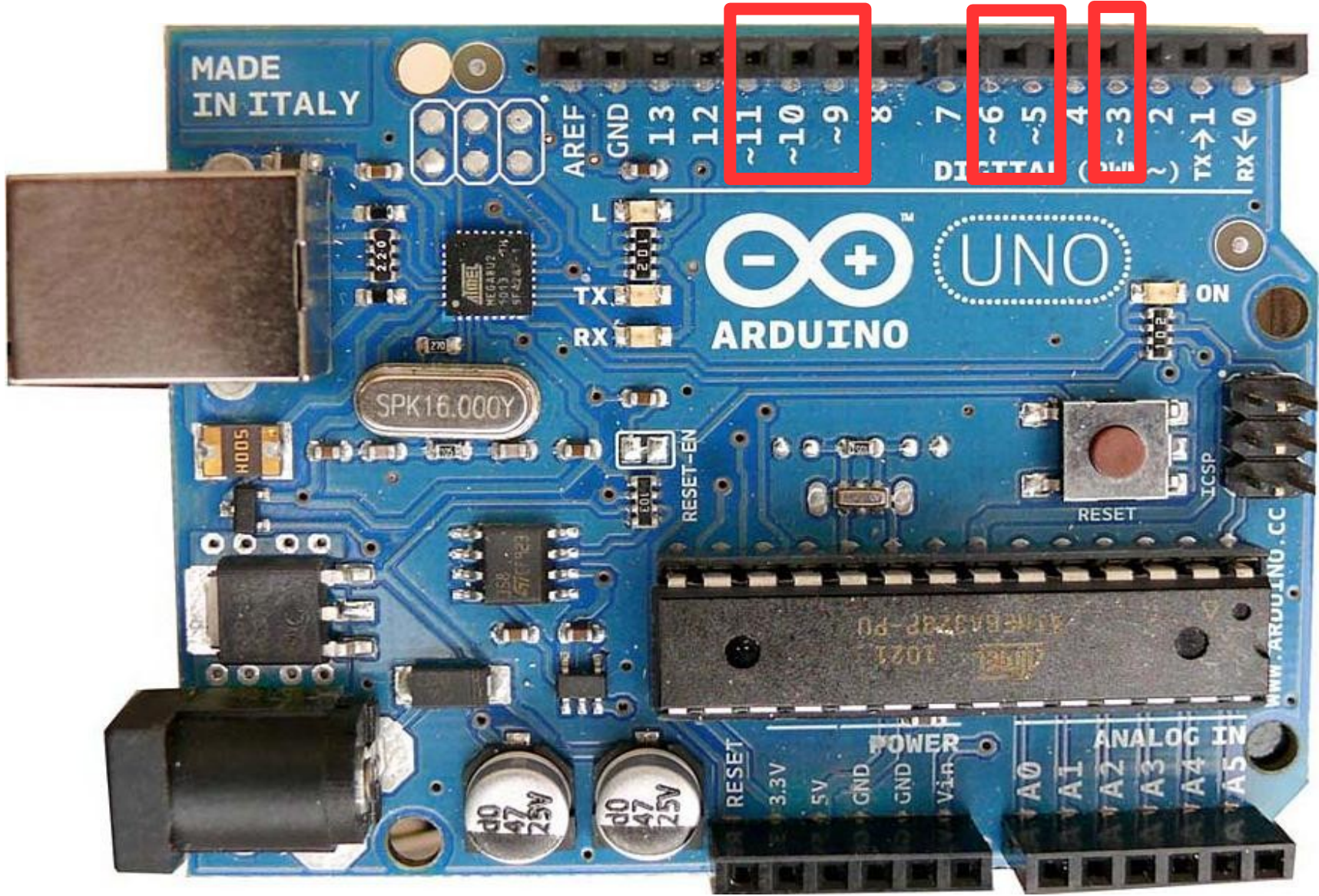
- Pulse Width Modulation
(Modulation à Largeur d'Impulsion)
- La fréquence d'un signal périodique correspond au nombre de fois que la période se répète en une seconde
- On la mesure en Hertz, noté Hz



PWM ou MLI



PWM



PWM

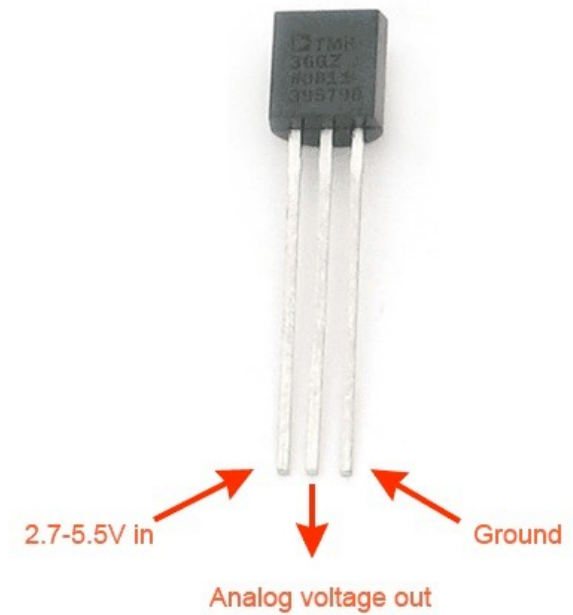
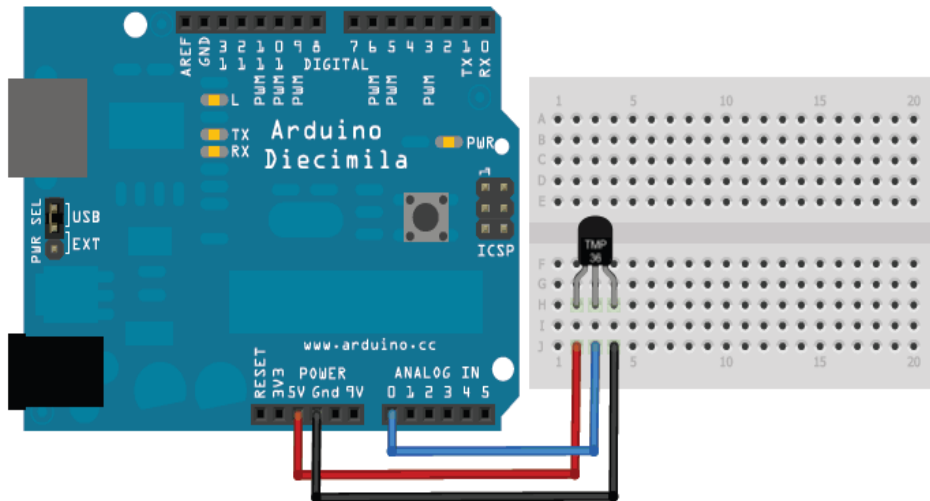
```
const int sortieAnalogique = 6;
void setup()
{
    pinMode(sortieAnalogique, OUTPUT);
}
void loop()
{
    analogWrite(sortieAnalogique, 127);
}
```

FADE

Demo

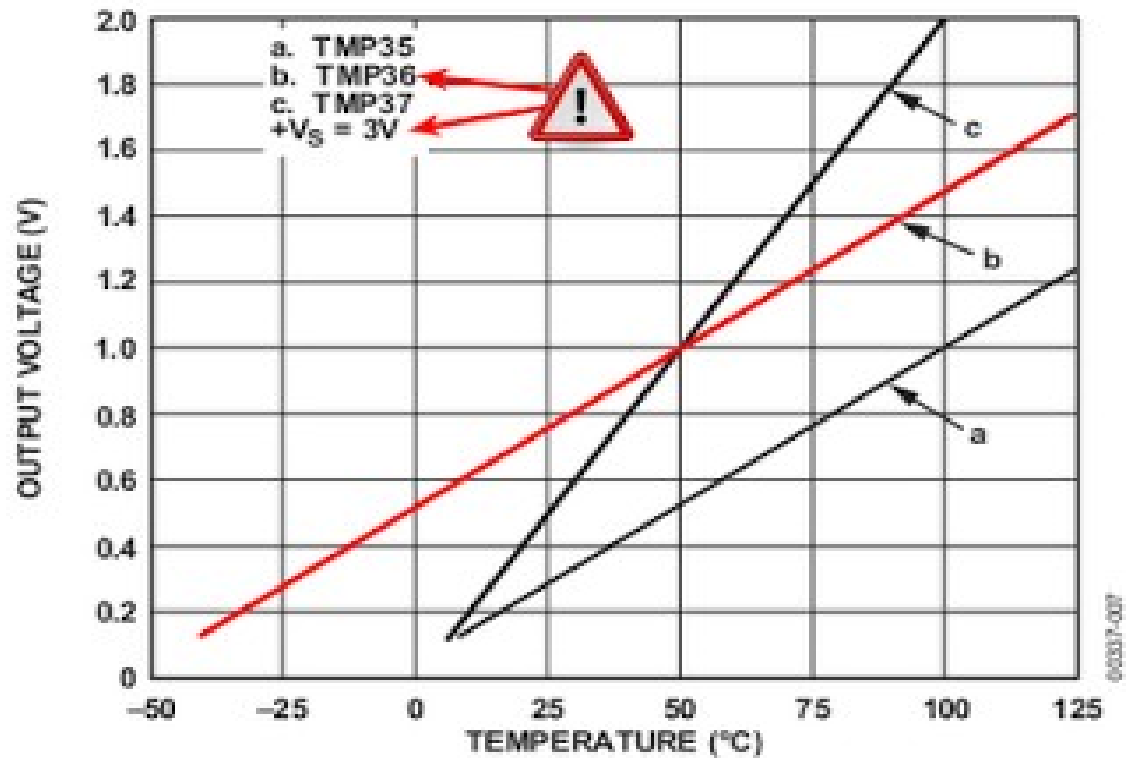
TMP36

- Capteur de température



TMP36 Datasheet

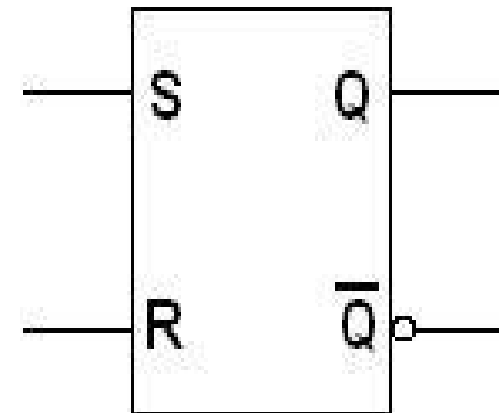
- Demo Datasheet



Bascules RS

- Bascule (Reset-Set)
- Table de vérité

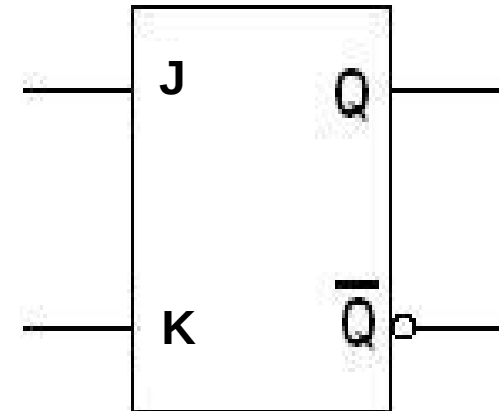
R	S	Q+
0	0	Mémoire
0	1	1
1	0	0
1	1	X



Bascules JK

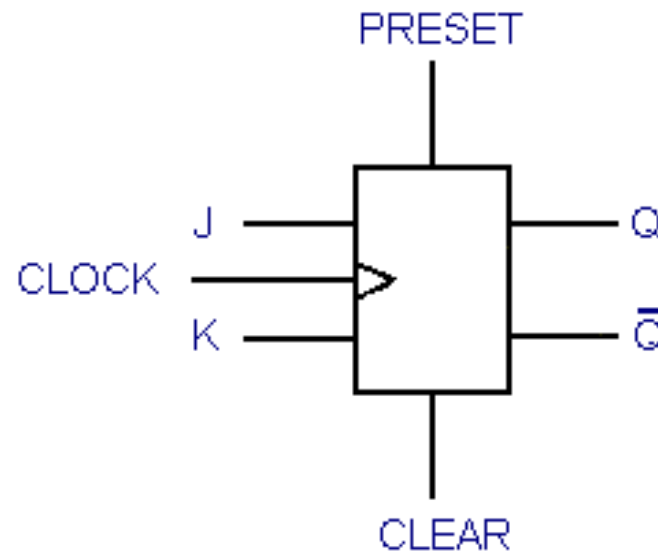
- Bascule (J-K)
- Table de vérité

J	K	Q+
0	0	Mémoire
0	1	1
1	0	0
1	1	\bar{Q}



Bascules JK

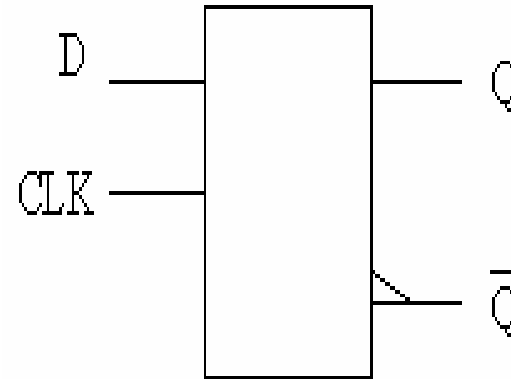
- Bascule (J-K)
 - Preset
 - Clear
 - Clock



Bascules D

- **Bascule (D)** : Recopie sa seule entrée D au rythme des impulsions d'une horloge : $Q^+ = 1$ si $D = 1$ et $Q^+ = 0$ si $D = 0$
- **Table de vérité**

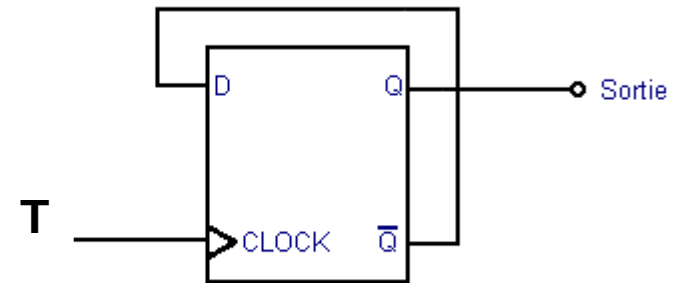
D	Q	Q ⁺
0	0	0
0	1	0
1	0	1
1	1	1



Bascules T

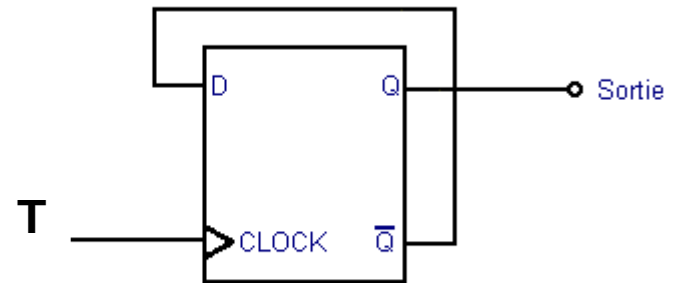
- **Bascule (T)** : Change d'état à chaque fois qu'une impulsion est reçue sur sa seule entrée
- **Table de vérité**

D	Q	Q+
0	0	0
0	1	1
1	0	1
1	1	0



Diviseur de fréquence

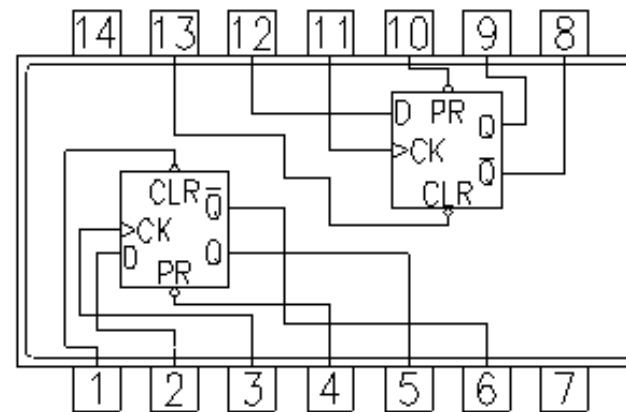
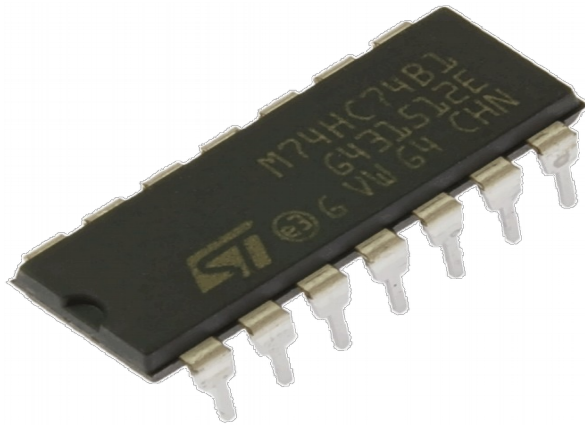
- **Bascule (T)** : Change d'état à chaque fois qu'une impulsion est reçue sur sa seule entrée



CI – Circuits intégrés



74hc74 – 2 bascules D



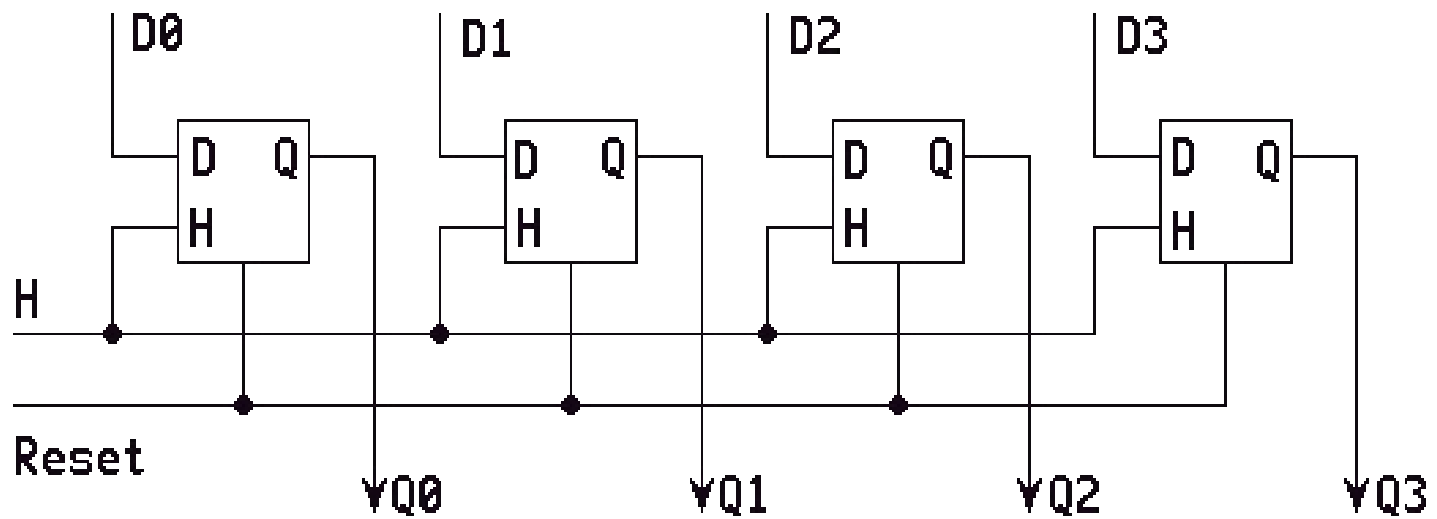
7474

Dual D Flip-Flop
with Preset and Clear

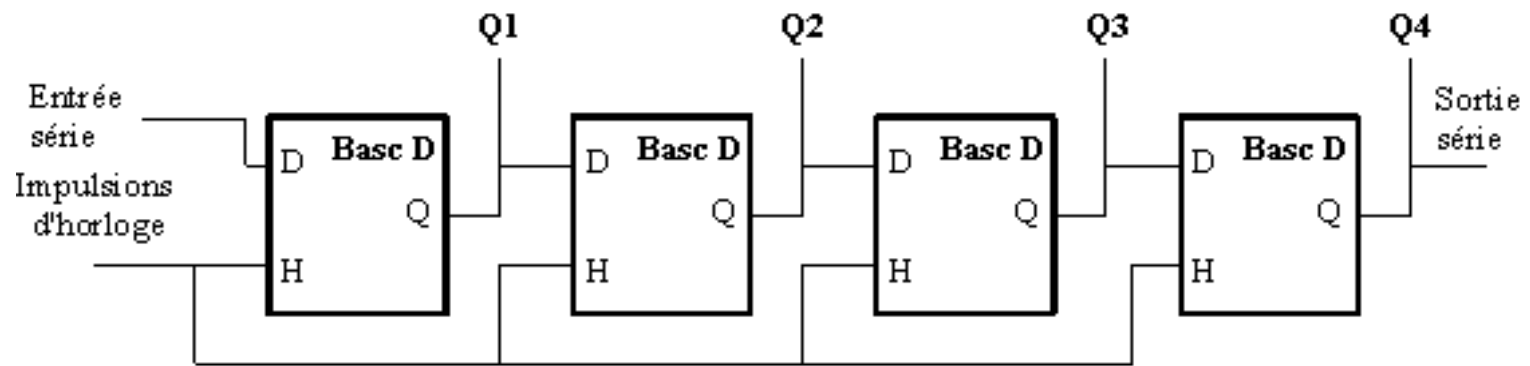
74hc74

Demo Datasheet

Les registres RW //



Registre RW Série



Exercices

- Réaliser un registre lecture écriture parallèle
- Réaliser un registre lecture en série et écriture en série
- Réaliser un registre lecture parallèle écriture en série

74hc595

- Registre 8 Bits

TDs

Demo