

Introduction à l'électronique digitale

Microcontrôleurs



Section de Physique

D. Mari, C. Meinen

Programme

Qu'est ce qu'un microcontrôleur ?

Arduino board

Architecture, Applications and Matériel, communication

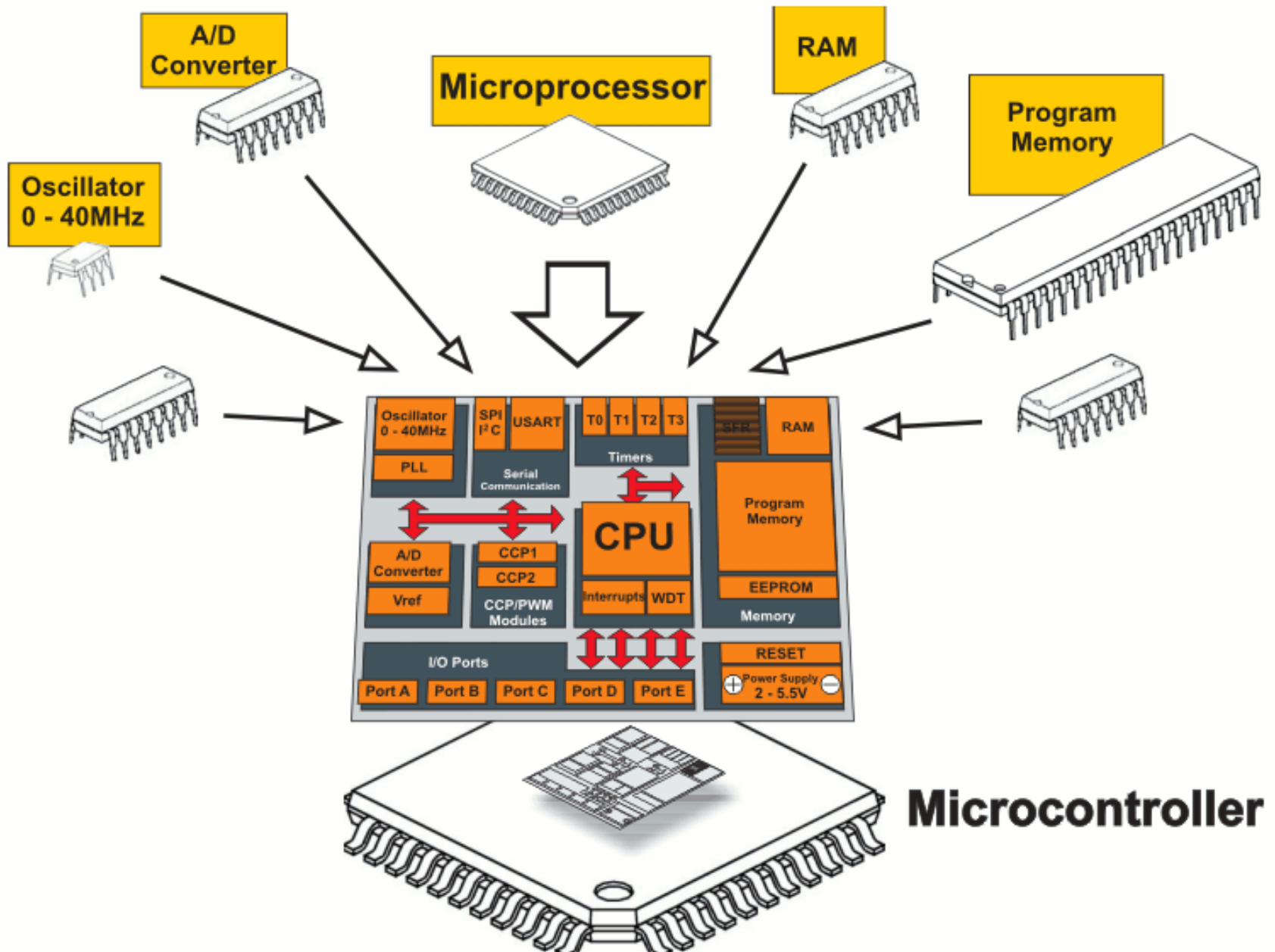
Prototypes:

Schémas et montages des PCB, diodes LED

Programmation:

Présentation du langage de programmation

Microcontrôleurs: de quoi on parle ?



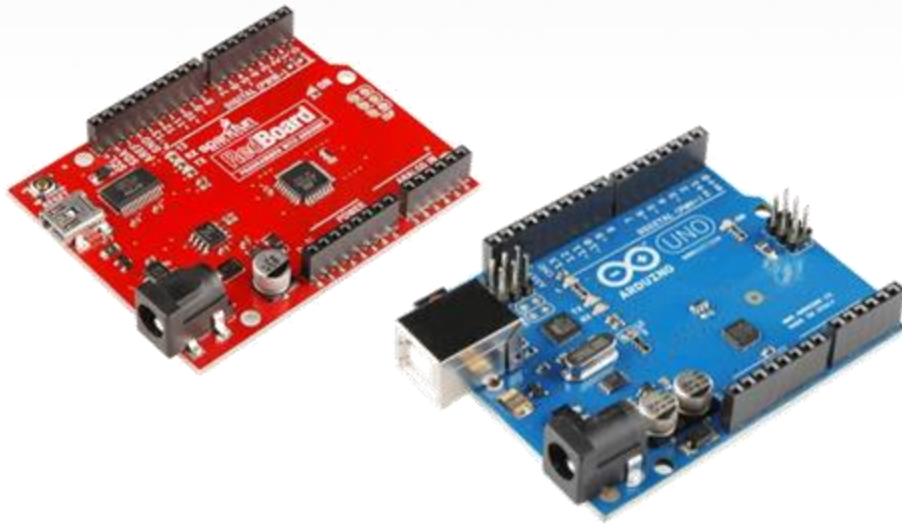
Arduino Board

Created in Ivrea, Italy
in 2005 by Massimo Banzi & David Cuartielles
Open Source Hardware



 Processor

Coding is accessible & transferrable → (C++, Processing, java)



Arduino...

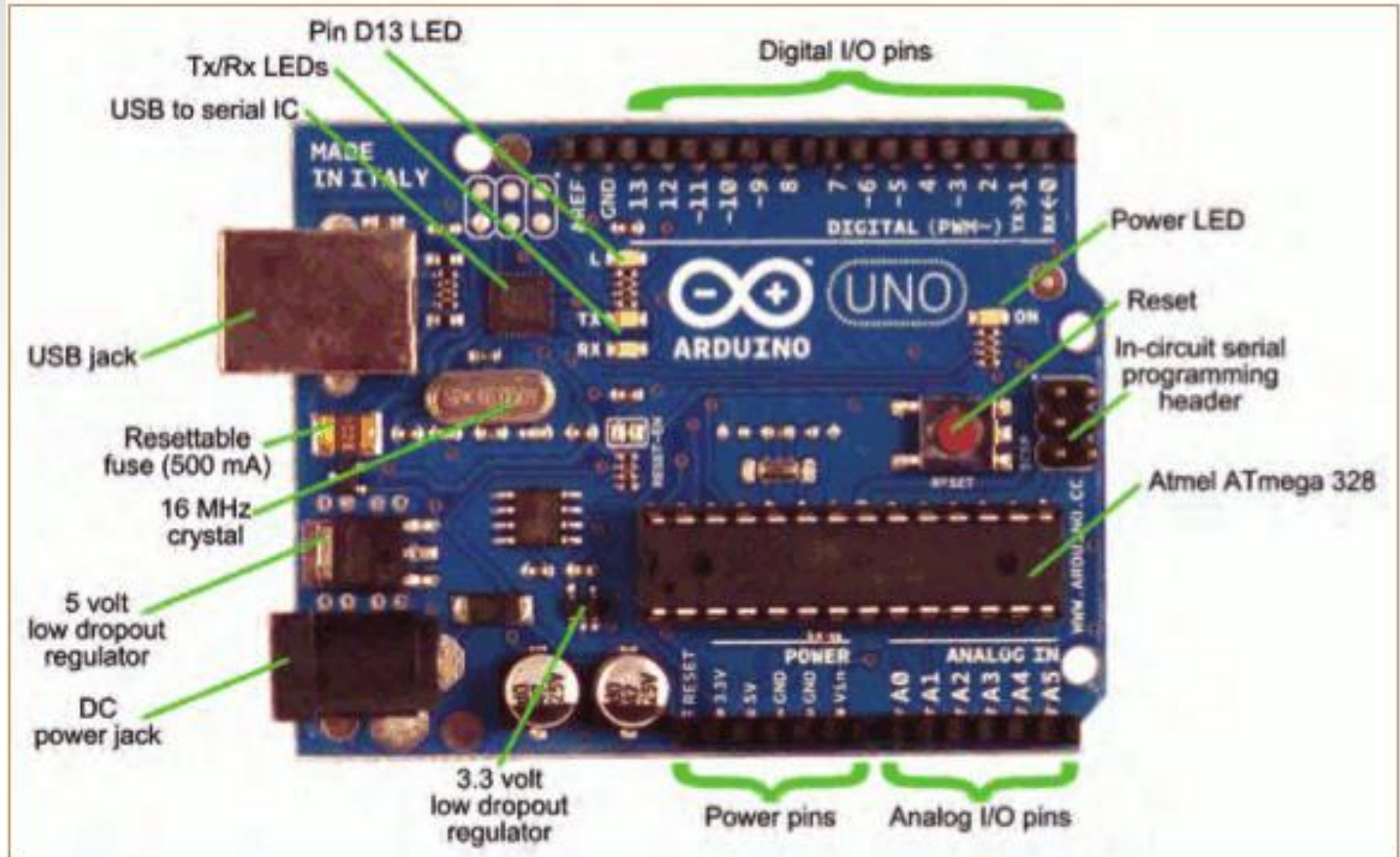
is the go-to gear for artists, hobbyists, students, and anyone with a gadgetry dream.

But....

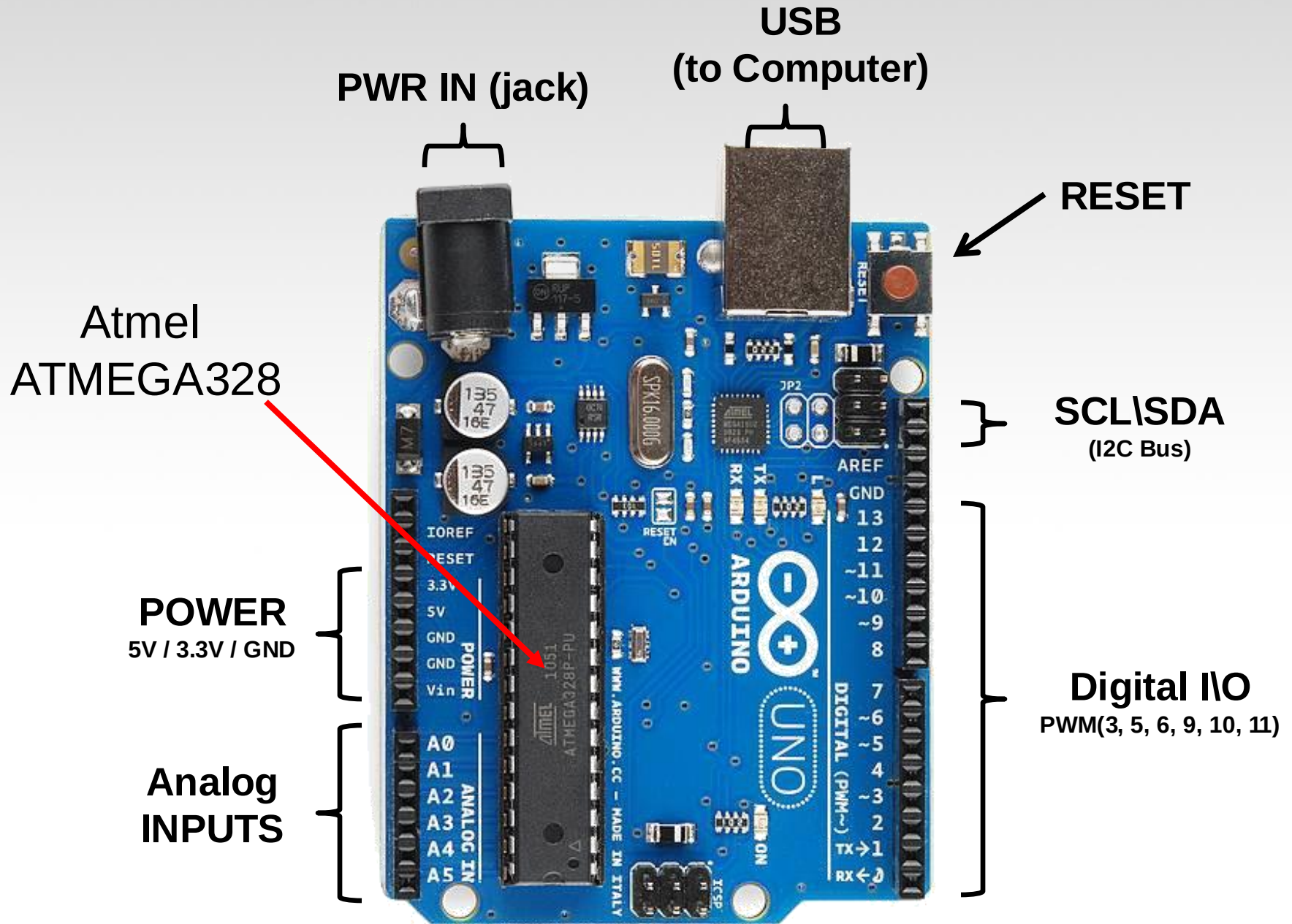
It rose out of another formidable challenge: how to teach students to create electronics... fast.

C'est l'objectif de
ce cours

The Arduino Development Board



Communication



ATmega 328 P

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 **SCK**
digital pin 12 **MISO**
digital pin 11 **PWM**
digital pin 10 **PWM**
digital pin 9 **PWM**

When using
ISP to program
the chip

Correspondance entre les broches de l'ATmega328 et celles de l'Arduino.

Microcontrôleur vs. Ordinateur

Un microcontrôleur gère des entrées – sorties digitales et analogiques

Il est programmable mais à l'étage le plus basique par un set d'instructions

Il n'a pas de système opérationnel

Tous les périphériques (display, clavier ou boutons..etc.) sont externes

Il effectue des tâches en série et non pas en parallèle

Il est très efficace pour des tâches spécifiques et consomme énormément moins d'énergie qu'un ordinateur

Concept: Entrée (input) vs. Sortie (output)

Vu par le microcontrôleur

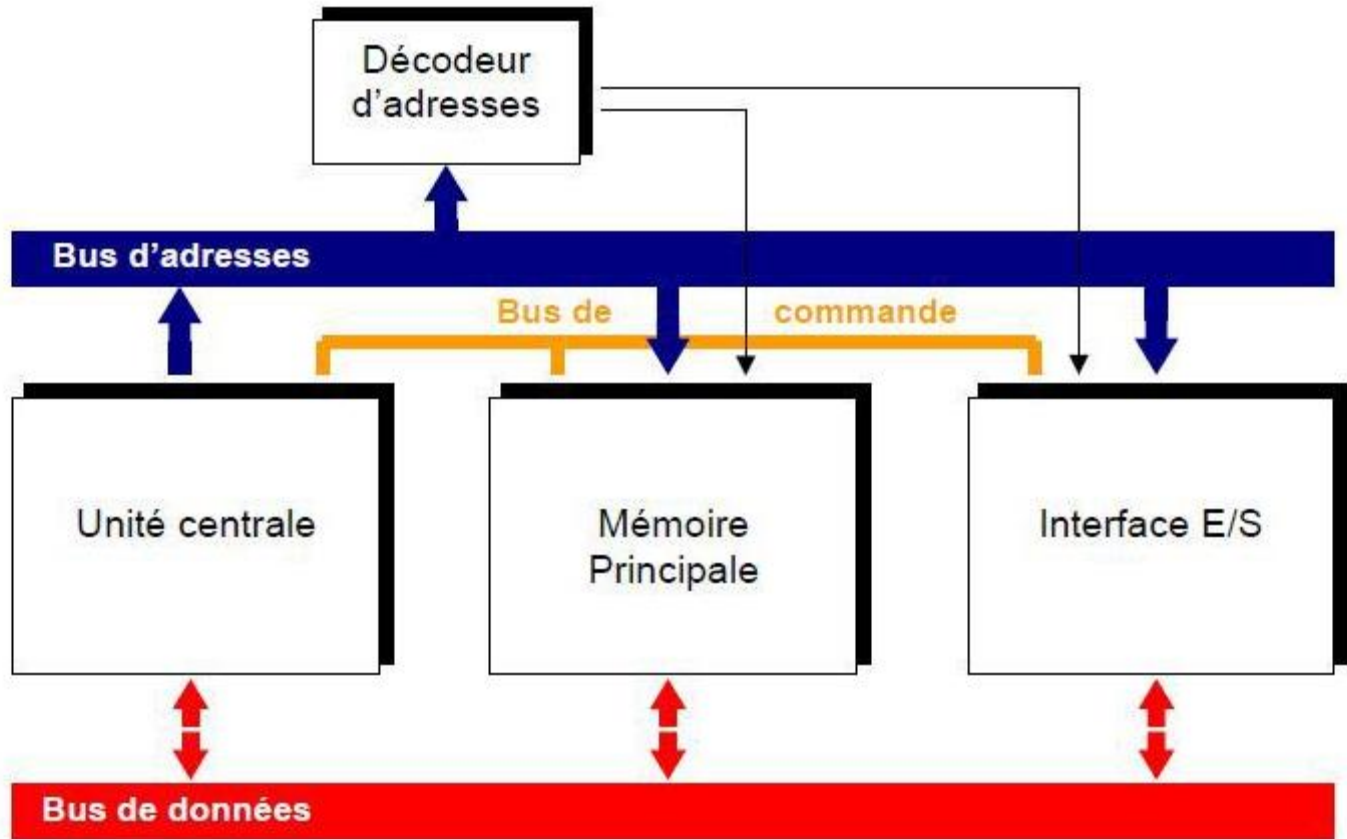
Input est un signal/ information qui va vers le microcontrôleur

Output est un signal qui sort du microcontrôleur

<u>Exemples</u>: Boutons Interrupteurs, Capteurs de lumière, force, humidité, temperature etc.	<u>Exemples</u>: LEDs, moteur DC servo-moteur, piezo buzzer, relay, actuateurs etc.
---	--

Architecture d'un processeur (von Neumann)

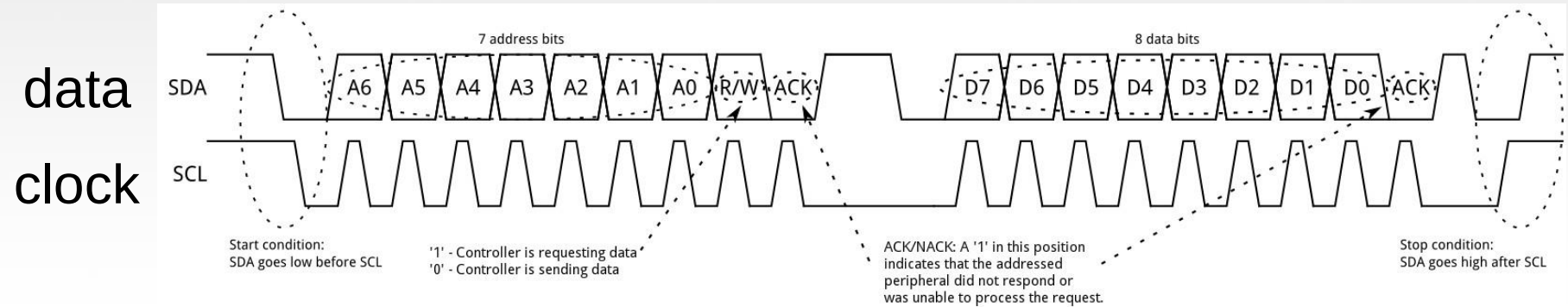
Instructions



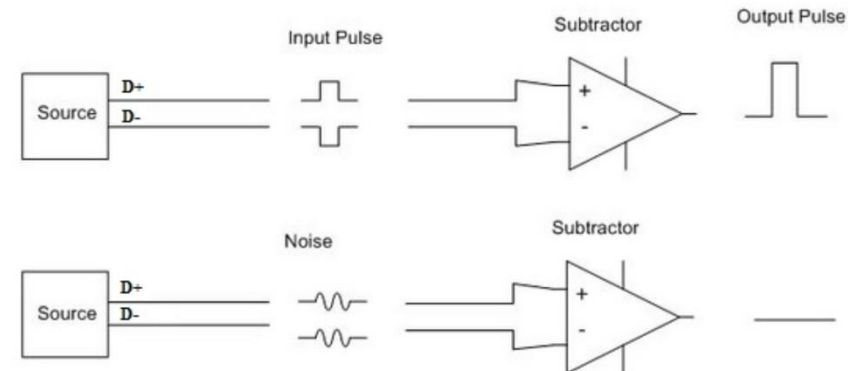
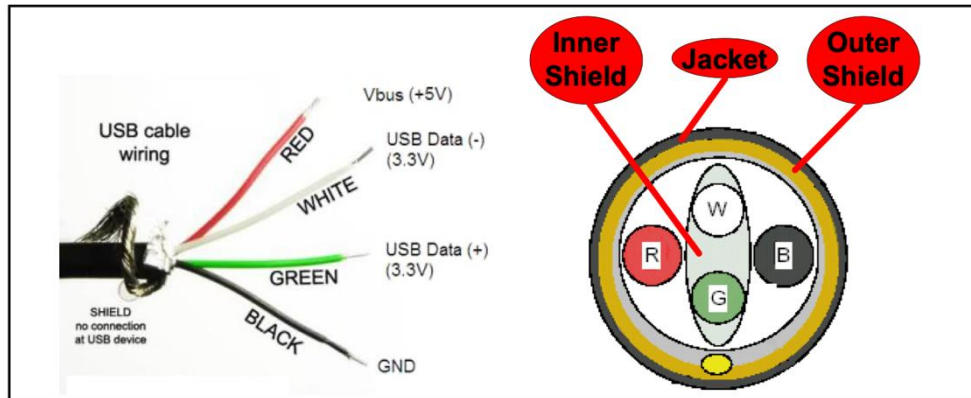
BUS de données

Transmettre des bytes de données

IIC ou I²C (Inter-Integrated Circuit)

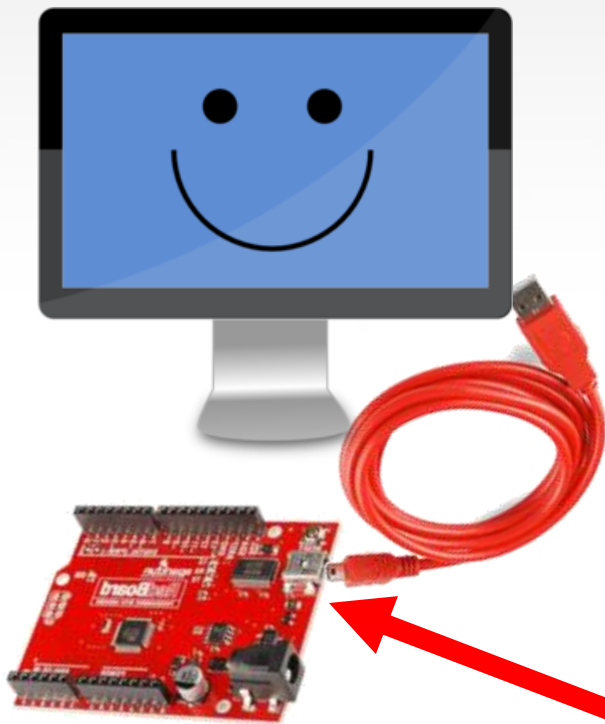


USB (Universal Serial Bus)



Communication sériele

Méthode utilisée pour transférer les données entre deux appareils



Entre l'ordinateur et et Arduino on utilise le BUS USB.
Les données sont transmises séquentiellement
comme ('0') et ('1').



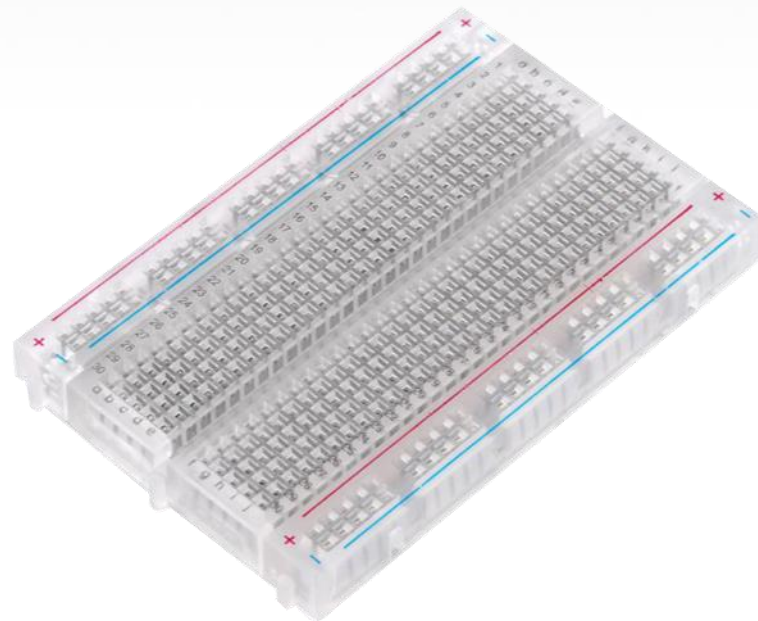
Arduino réserve le Digital I/O pin # 0 à la
reception (RX) et le Digital I/O pin #1 à la
transmission (TX)

Mais on a également un socle
pour le cable USB

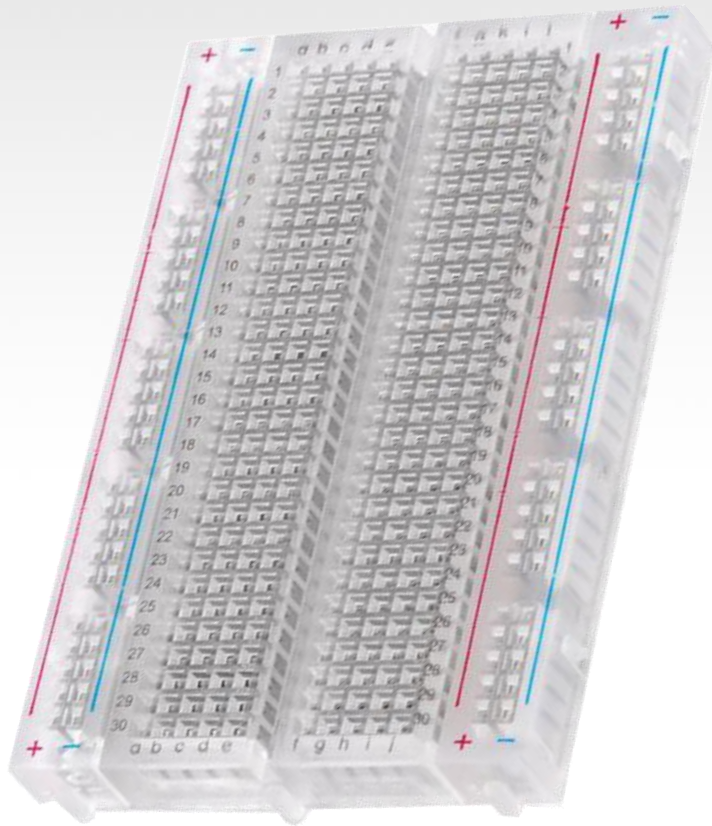
Prototypage de Circuits Solderless Breadboard

L'outil de base pour la conception de circuits:
Le plus important :

- Un breadboard c'est plus facile que souder
- Bien savoir quels trous sont connectés



Comment c'est fait ?

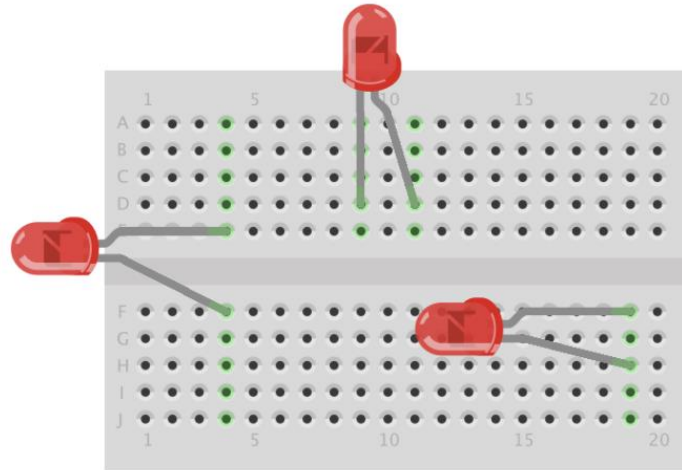


Les connexions

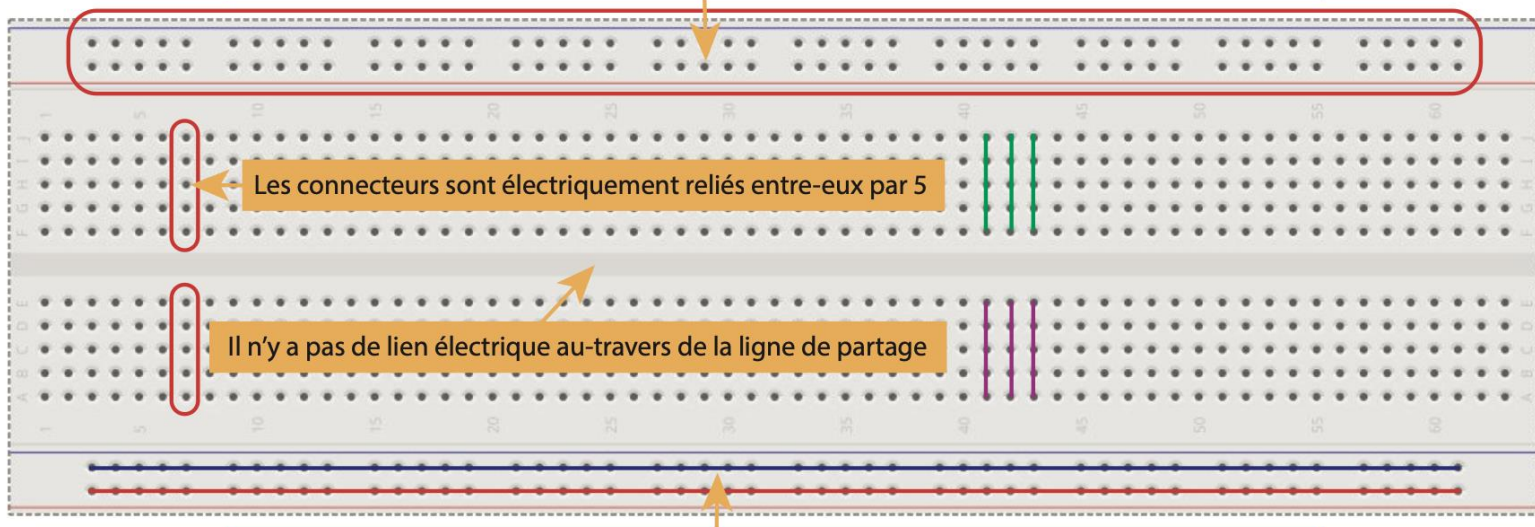
Juste

Juste

Faux



Les longues rangées sont utilisées pour connecter la masse (GND), en bleu et le +5V, en rouge

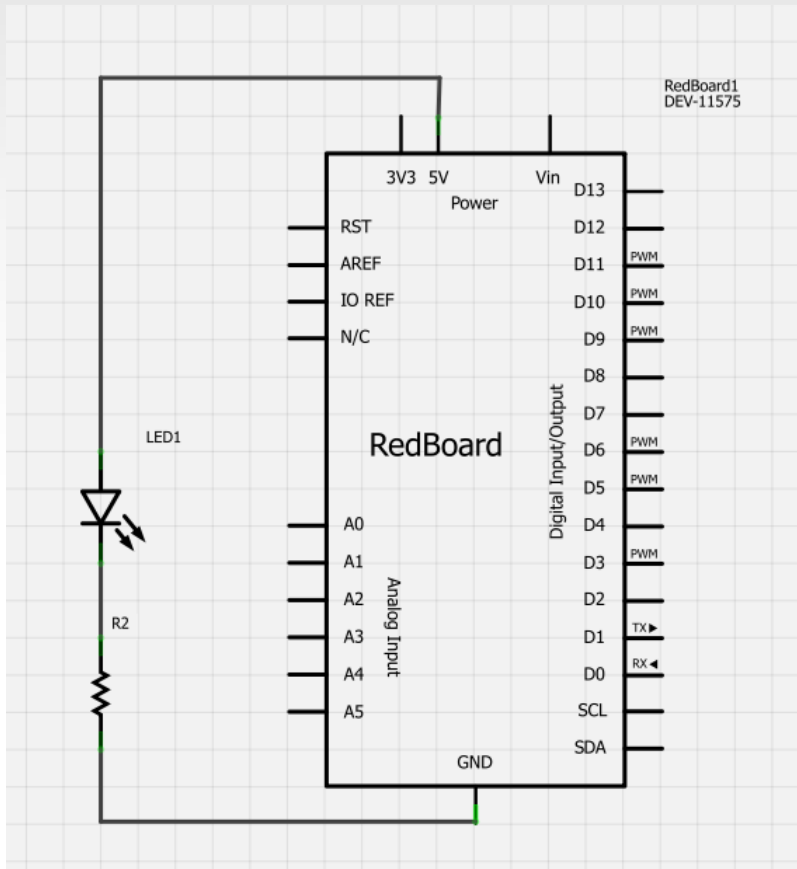


Les connecteurs sont électriquement reliés entre-eux par 5

Il n'y a pas de lien électrique au-travers de la ligne de partage

Tous les connecteurs d'une ligne extérieure sont électriquement reliés entre-eux

Alimenter un circuit simple



Utiliser le breadboard pour illuminer une LED. On a 5V à l'alimentation donc on limite le courant à 15 mA avec une résistance de 330 Ohm (Orange-Orange-Marron).

LED (rappel)

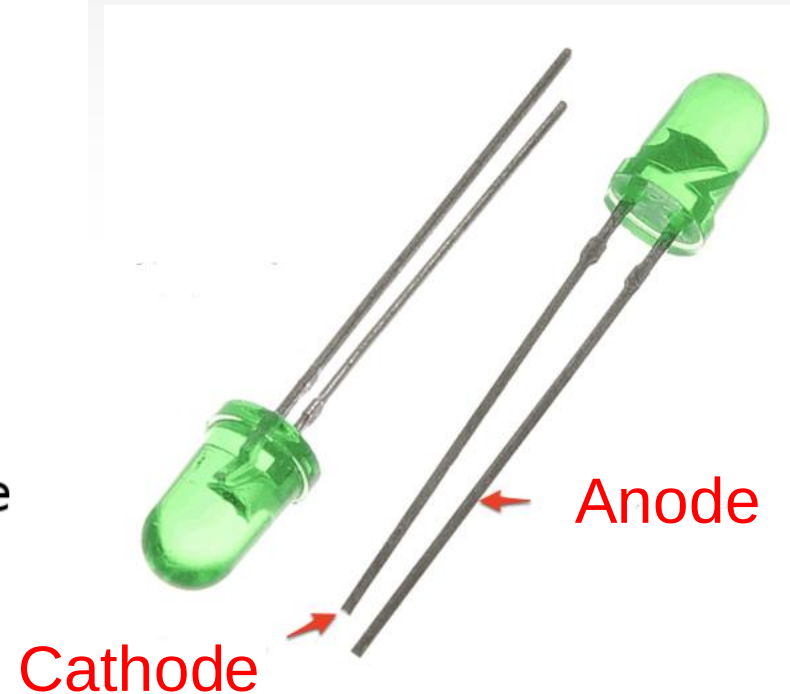
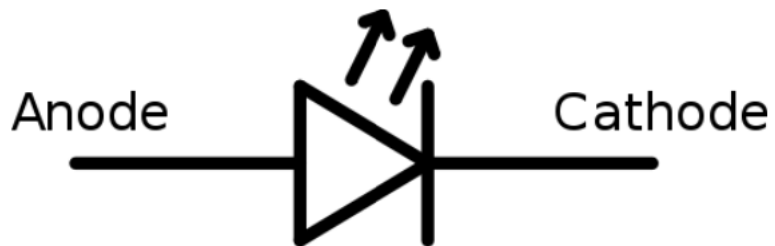
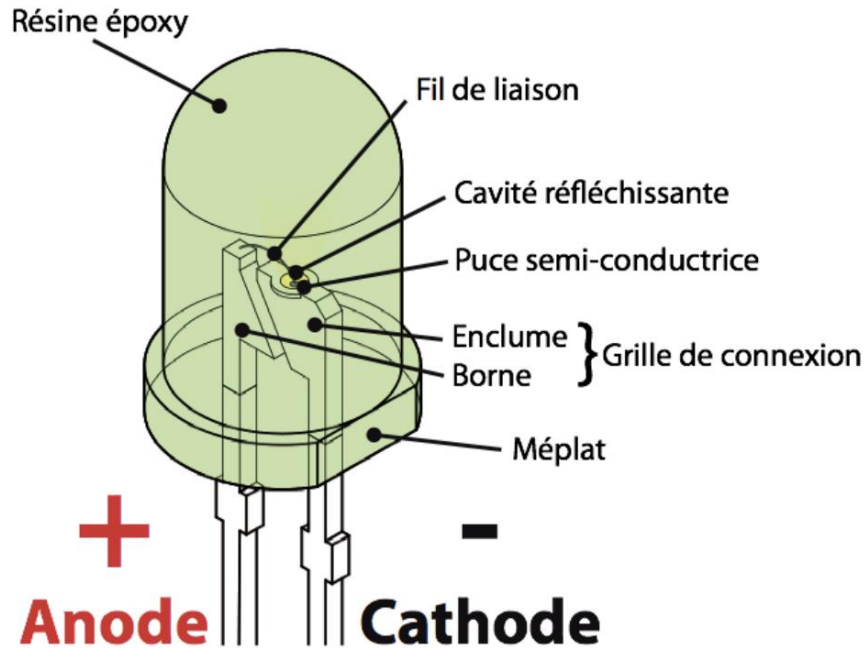
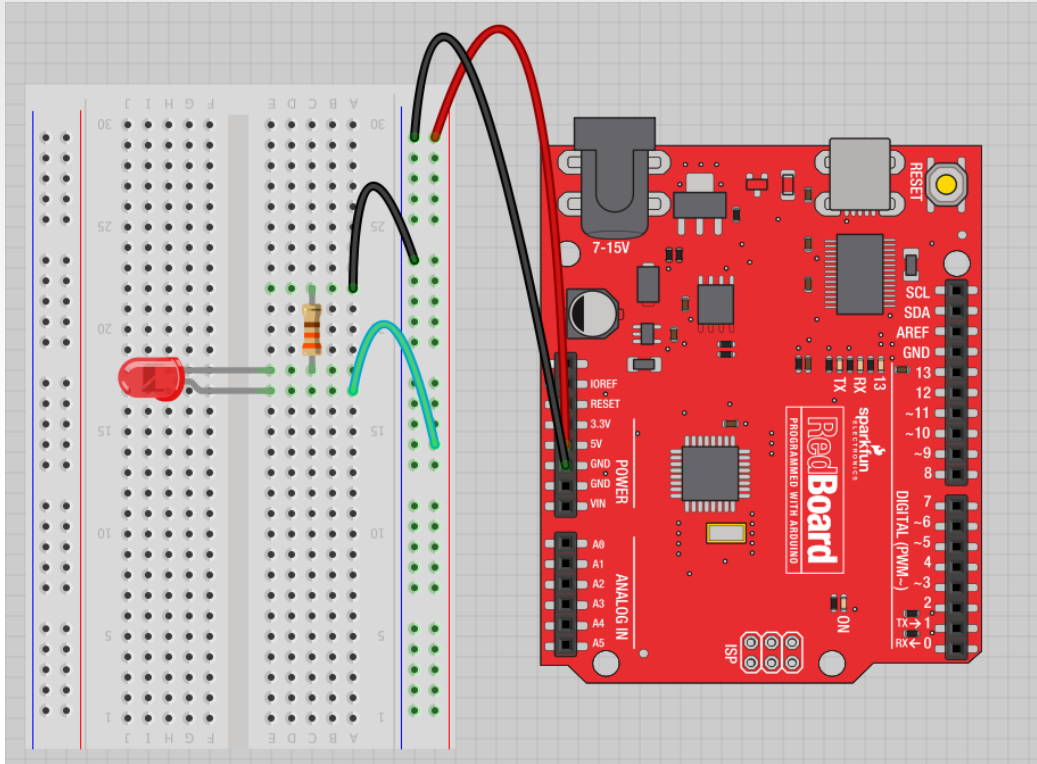
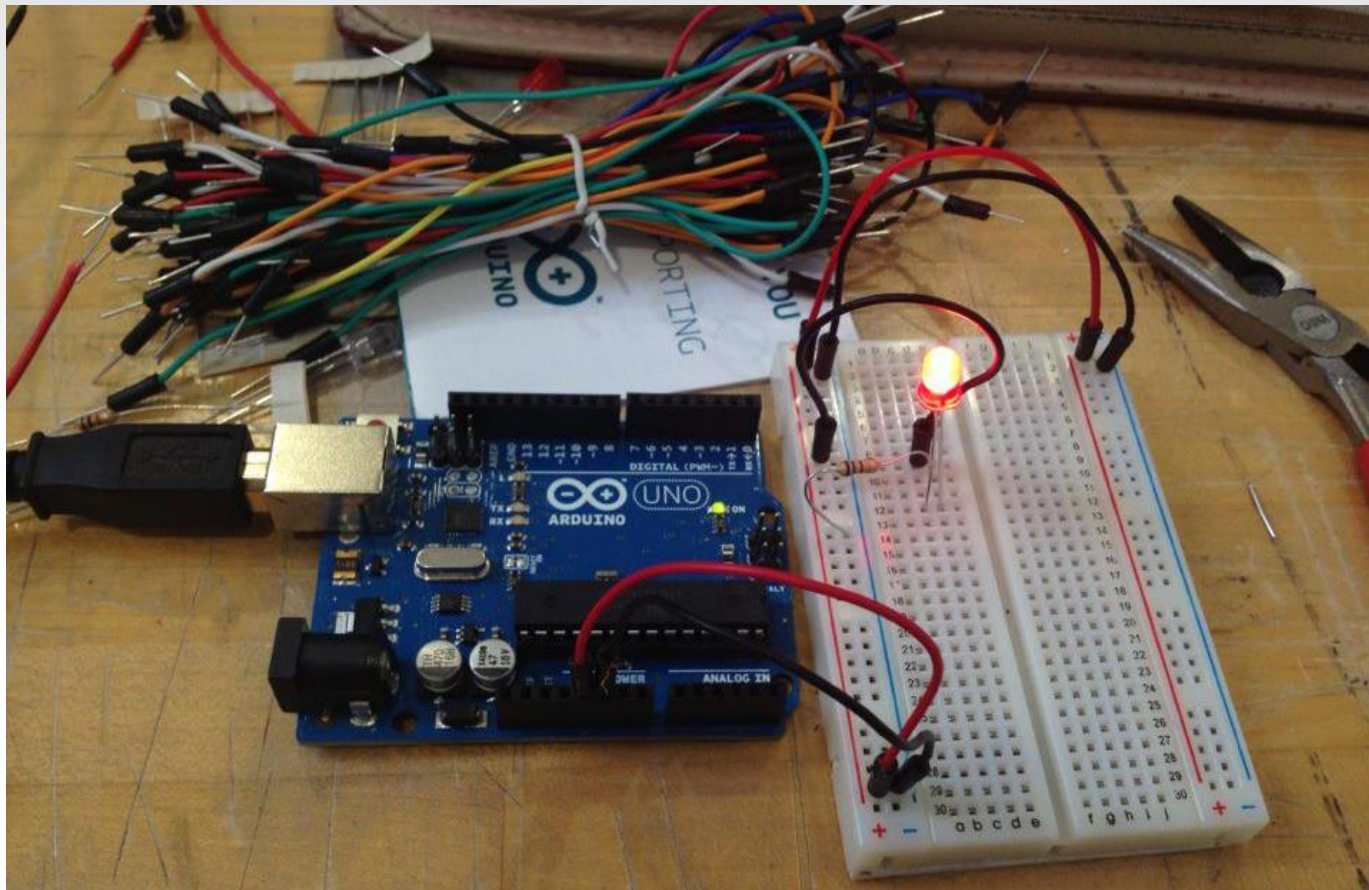


Image du Circuit



On utilise la sortie analogique à 5 V.

Contrôle de la sortie Programmer Arduino !

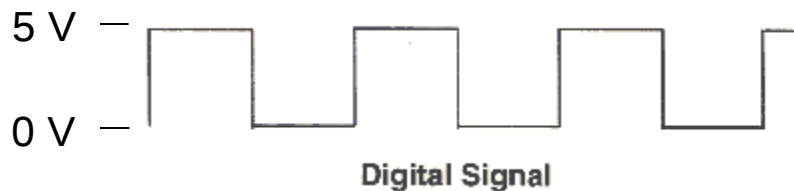


Rappel: Analogique vs. Digitale

Les microcontrolleurs traitent seulement des informations **digitales** : ON - OFF.

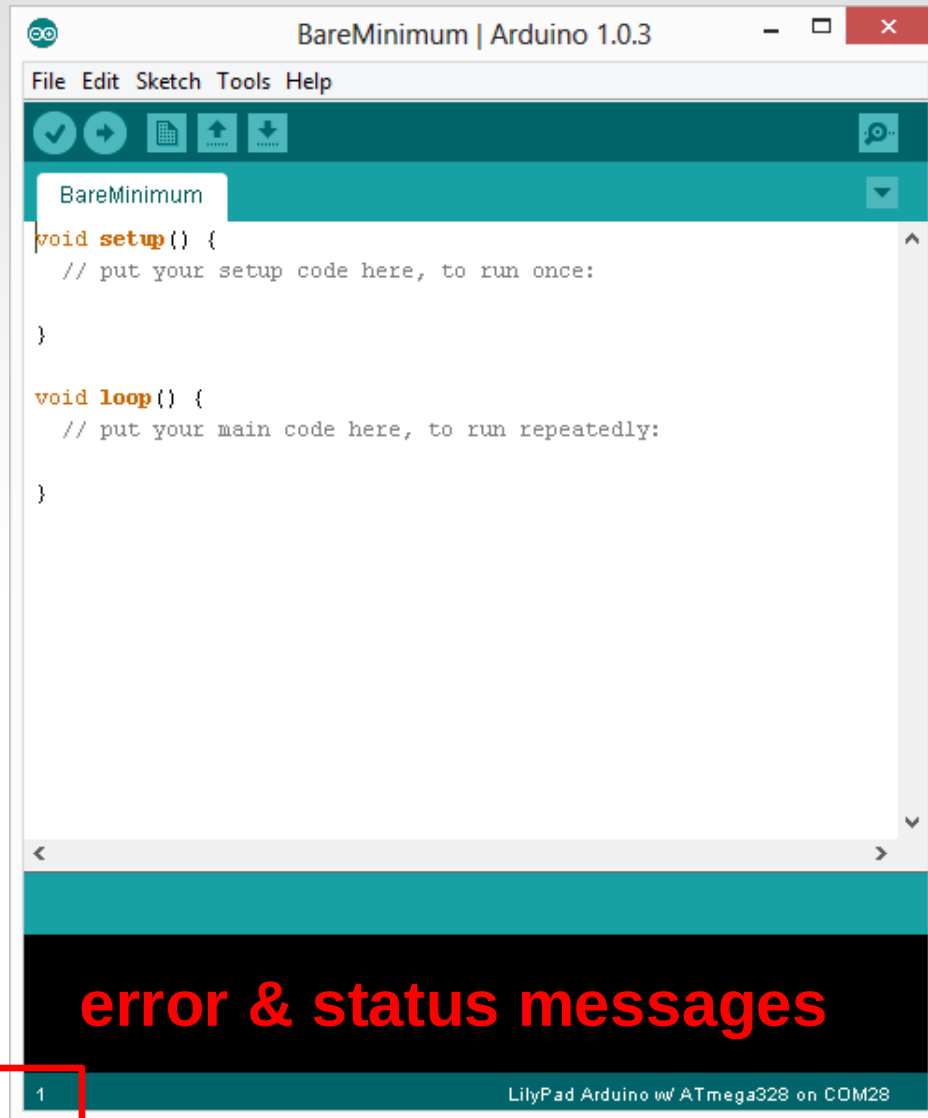
Les signaux analogiques en entrée (signal électrique de tension continu) peuvent aussi être traités par Arduino grâce à un convertisseur analogique-digital.

Pas de signaux analogiques en sortie



Arduino

Integrated Development Environment (IDE)

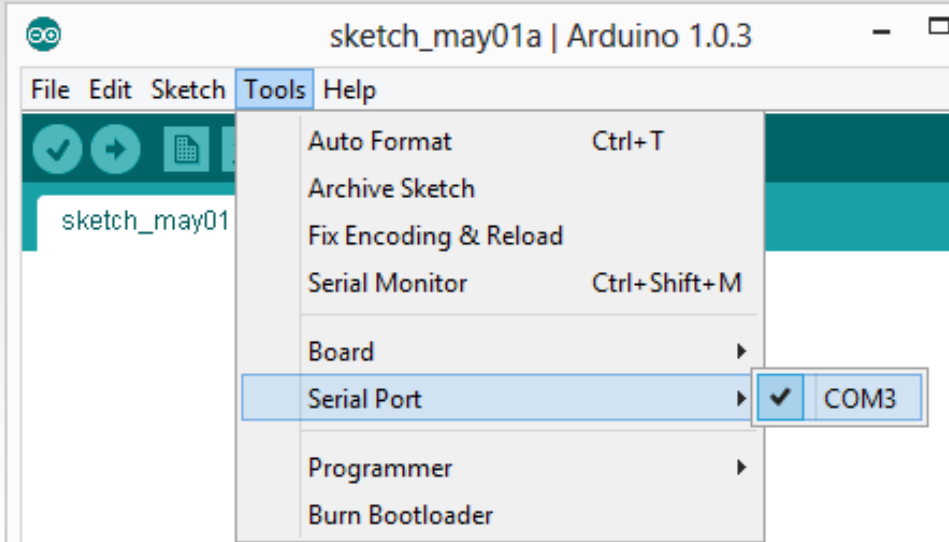


Two required functions /
methods / routines:

```
void setup()  
{  
    // runs once  
}
```

```
void loop()  
{  
    // repeats  
}
```

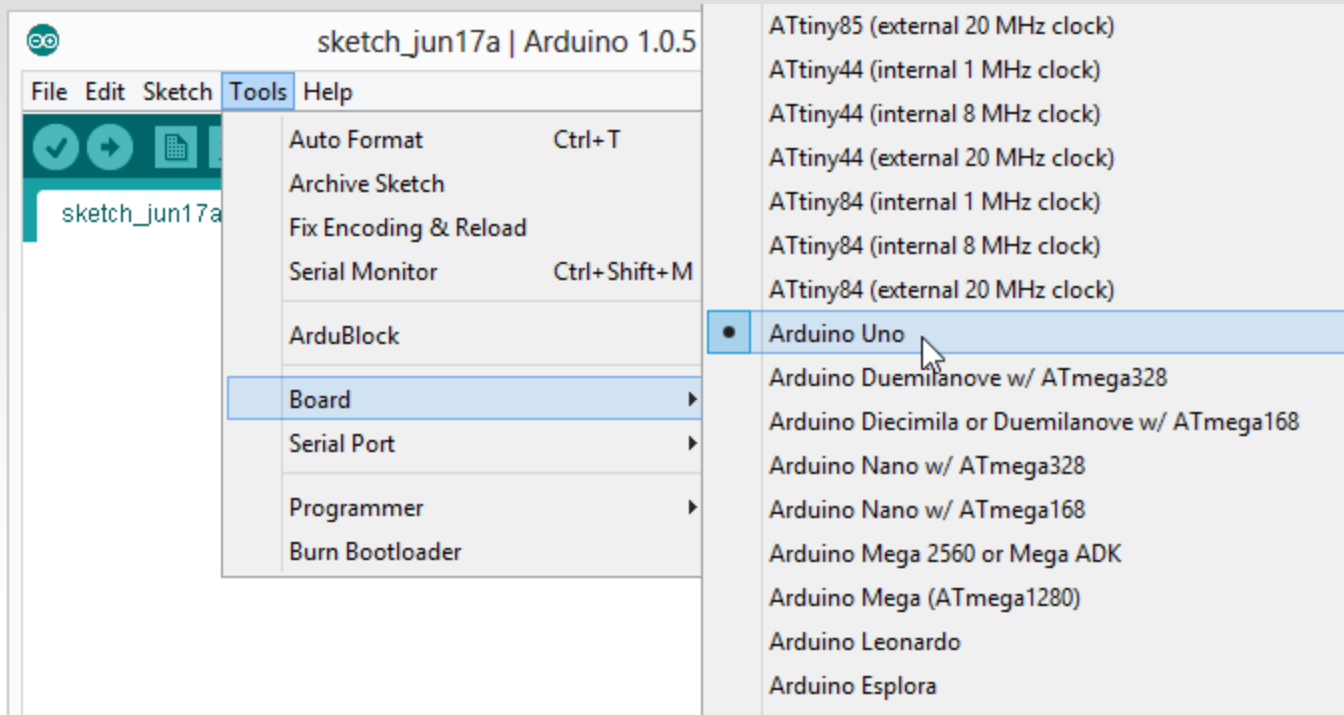
Settings: Tools → Serial Port



Your computer communicates to the Arduino microcontroller via a serial port → through a USB-Serial adapter.

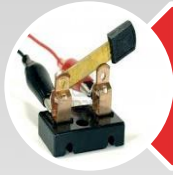
Check to make sure that the drivers are properly installed.

Settings: Tools → Board



Next, double-check that the proper board is selected under the Tools→Board menu.

BIG 6 CONCEPTS



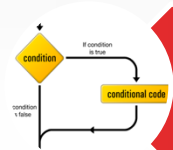
digitalWrite()



analogWrite()



digitalRead()



if() statements / Boolean



analogRead()



Serial communication



Let's get to coding...

Project #1 – Blink

“Hello World” of Physical Computing

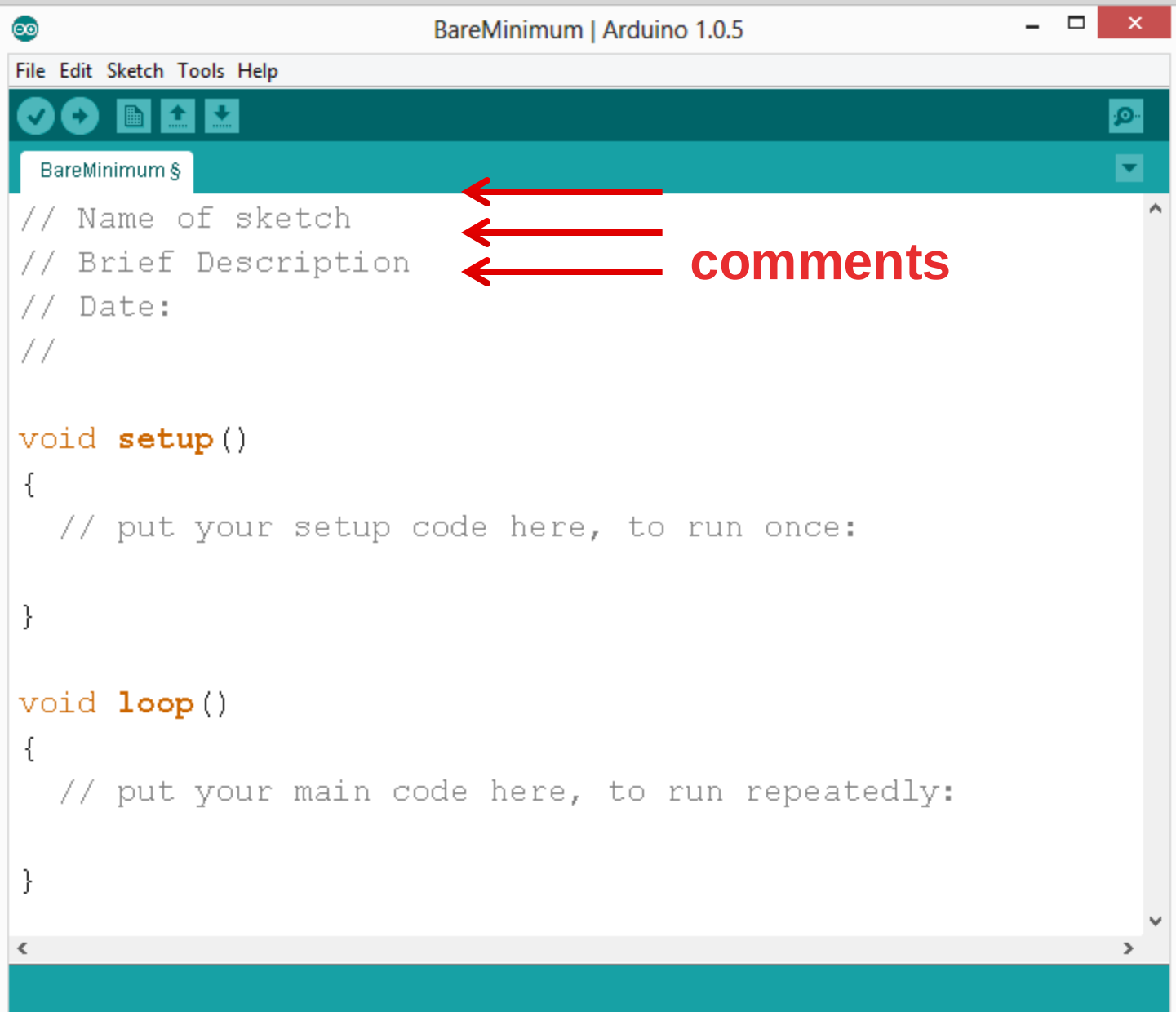
Pseudo-code – how should this work?



Commentaires

Comments are for you – the programmer and your friends...or anyone else human that might read your code.

```
// this is for single line comments
// it's good to put a description at the
  top and before anything 'tricky'
/* this is for multi-line comments
  Like this...
  And this...
*/
```



3 Commandes à savoir...

```
pinMode (pin, INPUT/OUTPUT) ;
```

```
ex: pinMode (13, OUTPUT) ;
```

```
digitalWrite (pin, HIGH/LOW) ;
```

```
ex: digitalWrite (13, HIGH) ;
```

```
delay (time_ms) ;
```

```
ex: delay (2500) ; // delay of 2.5 sec.
```

```
// NOTE: -> commands are CASE-sensitive
```

Project #1: Blink

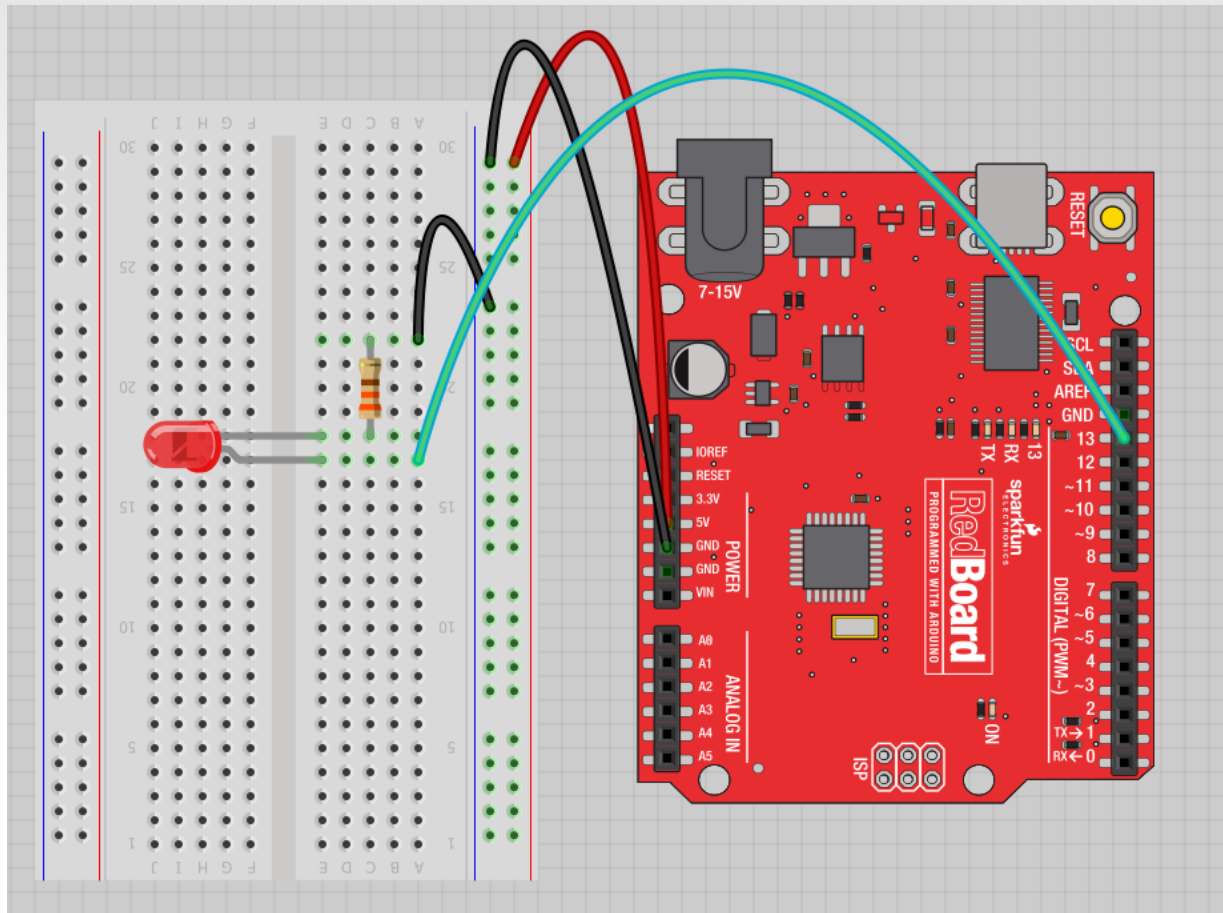


Image created in Fritzing

Déplacer le pin d'alimentation (green wire) depuis le power bus au pin 13 (ou autre Digital I/O) pin sur Arduino.

Programming Concepts: Variables

ProtosnapProMiniExample2 \$

```
// Comments go here
// Written by:  Joesephine Jones
// Date:  April 12, 2013

int sensorValue;
int ledPin;

void setup()
{
    // put your setup code here, to run once:
    int setupVariable;

}

void loop()
{
    // put your main code here, to run repeatedly:
    int loopScopeVariable
}
```

Variable Scope

Global

Function-level

Programming Concepts: Variable Types

Variable Types:



8 bits

byte
char



16 bits

int
unsigned int



32 bits

long
unsigned long
float

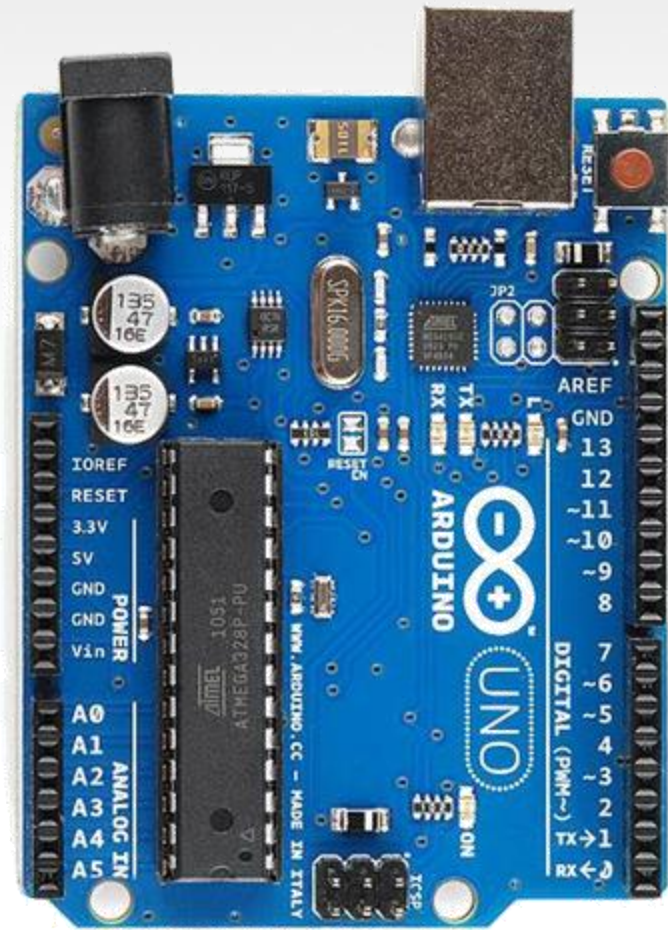
(Analogique ou Digital ?)

Arduino n'a pas de sortie analogique pilotable.

Mais certains pins sur Arduino permettent de modifier l'output et mimer un signal analogique.

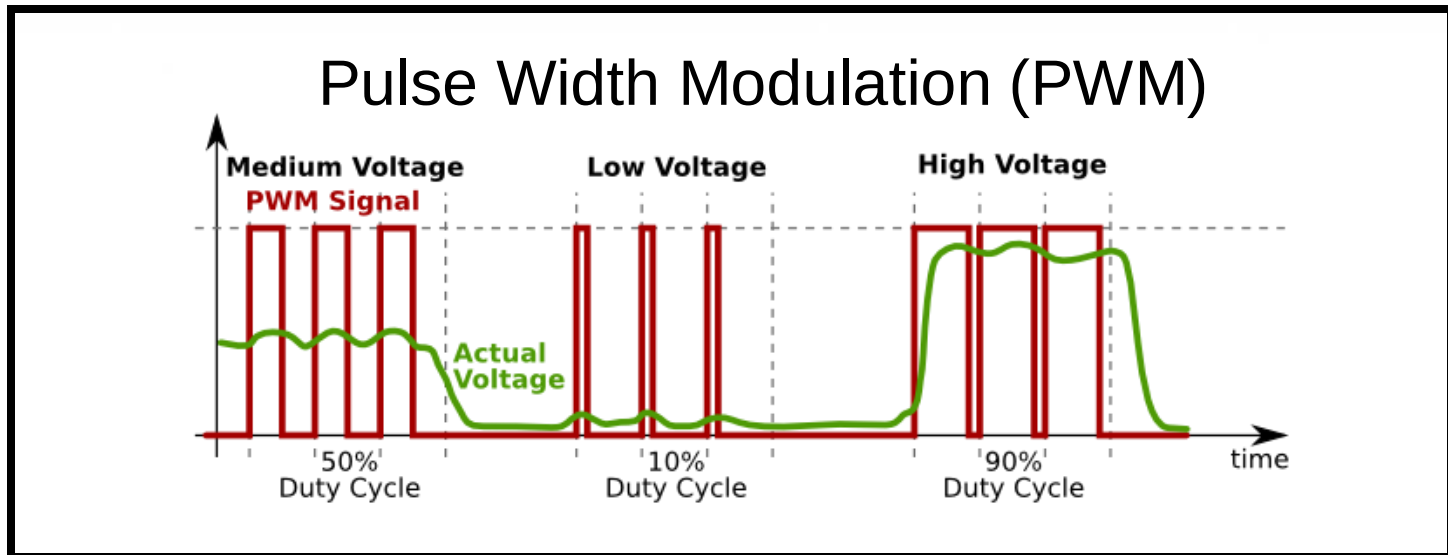
Obtenu par la technique :

Pulse Width Modulation (PWM)



(Pulse Width Modulation)

Pour créer un signal analogique, le microcontrôleur utilise la technique **PWM**. En variant le duty cycle, on peut mimer un signal analogique ayant une certaine valeur efficace ou RMS.



Utiliser la commande analogWrite

```
analogWrite(pin, val);
```

pin – se réfère au OUTPUT pin (limité aux pins 3, 5, 6, 9, 10) – marqué par ~ symbole

val – 8 bit value (0 – 255).

0 => 0V | 255 => 5V

