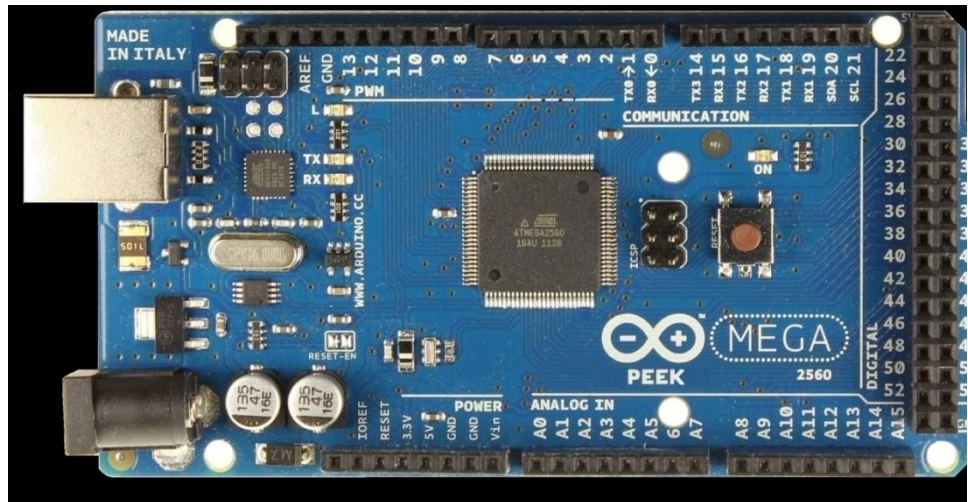


# ARDUINO : Sorties PWM

## I. Les connecteurs "PWM" :

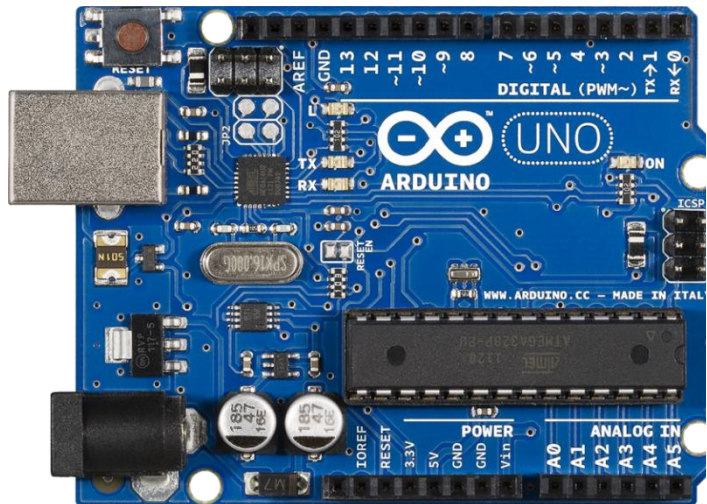
**Q1. Entourer les connecteurs notés PWM sur la carte Arduino MEGA 2560**



**Q2. Donner le symbole permettant de différencier les sorties numériques (DIGITAL) des sorties PWM sur la carte Arduino UNO :**

**Q3. Donner les numéros des sorties PWM :**

**Q4. Entourer les connecteurs et les numéros des broches des sorties PWM.**



## II. Programmation d'une carte Arduino :

Demander une carte Arduino ainsi qu'un câble USB (voir le professeur).



Lancer le logiciel Arduino , sélectionner la bonne carte (c'est comme le port salut, c'est écrit dessus) à l'aide du MD (Menu Déroulant) « Outils » puis « Type de carte ». Sélectionner le port pour que la communication entre le PC et la carte puisse avoir lieu (MD « Outils » puis « Port série », prendre le 3<sup>ème</sup> COM de la liste).

### a) Allumage et extinction d'une led :

Sélectionner dans le MD « Exemples » puis « 01.Basics » le fichier « Fade ».

**Q5. Donner la traduction de fade :**

**Q6. Rappeler sur quelle broche est relié la led "L" de la carte :**

## ARDUINO : Sorties PWM

*Q7. Quelle modification faut-il alors faire dans le programme :*

Lors du programme "blink" on utilisait la fonction digitalWrite.

*Q8. Quels étaient les 2 états possibles de la led ?*

Dans le programme "fade", on utilise la fonction analogWrite.

En vous aidant uniquement de l'intensité lumineuse de la led dans les programmes "blink" et "fade" (retéléverser ces 2 programmes éventuellement).

*Q9. Donner la différence entre les fonctions digitalWrite et analogWrite :*

### b) Explication :

digital signifie numérique (0 ou 1) soit une tension de 0V ou 5V

analog est la contraction de analogique soit une tension de 0V à 5V

analogWrite(broche, val) : change la tension moyenne d'une des sorties marquées PWM. La valeur de la broche peut être 3, 5, 6, 9, 10 ou 11 et val est un nombre compris entre 0 et 255, correspondant à une tension de sortie de 0V à 5V.

Le programme "fade" est le suivant :

```
/*
  Fade
  This example shows how to fade an LED on pin 9
  using the analogWrite() function.
  This example code is in the public domain.
  */

int led = 9;           // the pin that 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);
}
```

## ARDUINO : Sorties PWM

**Q10.** *Rappeler la signification des symboles /\* et \*/ :*

--

**Q11.** *Donner la traduction de : This example shows how to fade an LED on pin 13*

--

**Q12.** *Donner la traduction de brightness et de how bright the LED is :*

--

**Q13.** *Donner la traduction de : fadeAmount et how many points to fade the LED by :*

--

### c) Fonction setup :

**Q14.** *Donner la traduction de : the setup function runs once when you press reset or power the board*

--

**Q15.** *Donner la traduction de : declare pin 13 to be an output*

--

**Q16.** *Quelles autres broches pourrait-on utiliser sur :*

La carte arduino 2560 :	
La carte arduino uno :	

**Q17.** *En déduire la traduction de : pinMode(led, OUTPUT);*

--

### d) Fonction loop :

**Q18.** *Donner la traduction de Loop et de the loop function runs over and over again forever*

--

**Q19.** *Donner la traduction de : // set the brightness of pin 9:*

--

**Q20.** *Donner la traduction de write :*

--

**Q21.** *Donner la traduction de // change the brightness for next time through the loop:*

--

## ARDUINO : Sorties PWM

**Q22.** Donner la traduction de : *reverse the direction of the fading at the ends of the fade:*

**Q23.** Donner la traduction de : *if*

L'instruction "if (brightness == 0 || brightness == 255)" est une instruction de test.

Remarquer les 2 signes "==" consécutifs (**toujours** pour un test) alors qu'un seul "=" signifie ce que l'on appelle une affectation (int led = 9).

**Q24.** Donner la signification de "||" dans un test :

**Q25.** En déduire la traduction de : *if (brightness == 0 || brightness == 255)*

**Q26.** Compléter le tableau suivant en vous aidant du programme (simplifié) :

```

Fade | Arduino 1.6.4
Fichier Édition Croquis Outils Aide

Fade$

int brightness = 0;    // how bright the LED is
int fadeAmount = 5;   // how many points to fade the LED

// the loop routine runs over and over again forever:
void loop()
{
  // change the brightness for next time through the
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends
  if (brightness == 0 || brightness == 255)
  {
    fadeAmount = -fadeAmount ;
  }
}
    
```

		variable	
		brightness	fadeAmount
Etat initial		0	5
	1	5	5
	2		5
	3		
	4		
	5		
	6		
	Etc...	Etc..	
	49		
	50		
	51	255	
	52		
	53		
	54		
	55		

## ARDUINO : Sorties PWM

		Etc...	Etc...	

**Q27.** *Donner la traduction de wait for 30 milliseconds to see the dimming effect*

Donner la traduction de delay :

**Q28.** *En déduire la signification de (préciser l'unité) delay(30) ;*

Réécriture du programme :

**Q29.** *Rappeler la signification de l'abréviation int :*

**Q30.** *Donner la valeur max et min d'une variable de type int :*

**Q31.** *Donner la valeur max et min de la variable "amount" :*

**Q32.** *Donner la valeur max et min de la variable "brightness" :*

**Q33.** *En déduire le type de variable que l'on devrait utiliser pour la variable "amount" :*

**Q34.** *En déduire le type de variable que l'on devrait utiliser pour la variable "brightness" :*

Ce programme est encore modeste, seule la broche 13 est utilisée. On pourrait imaginer qu'il y en ait d'autres et que le programme soit beaucoup plus long. Il devient alors difficile de se souvenir à quoi correspond telle ou telle broche. Il vaut mieux alors donner un nom au numéro de la broche en créant une équivalence :

```
#define led 13 // déclaration d'une équivalence avec un nom "pertinent" (ne pas mettre de ;)
```

La ligne (`#define led 13` // déclaration d'une équivalence avec un nom "pertinent") est à mettre après les commentaires `*/` et avant le `void setup()`

Effectuer les modifications ci-dessus et remplacer tous les nombres 13 par la variable led (avec le `#define`, le logiciel associe tous les noms led à 13)

## ARDUINO : Sorties PWM

De plus dans un souci de lisibilité et de compréhension, mettre les accolades des fonctions setup, loop et du test if alignées (voir le fichier présentation des cartes arduino). Vérifier le bon fonctionnement de votre programme.

Appeler **le professeur pour vérifier de votre programme ainsi modifié.**

(Sauvegarder le)

### III. Nouveau programme :

Recopier le programme ci-dessous :

```

Fade_a_completer | Arduino 1.6.5
Fichier  Édition  Croquis  Outils  Aide

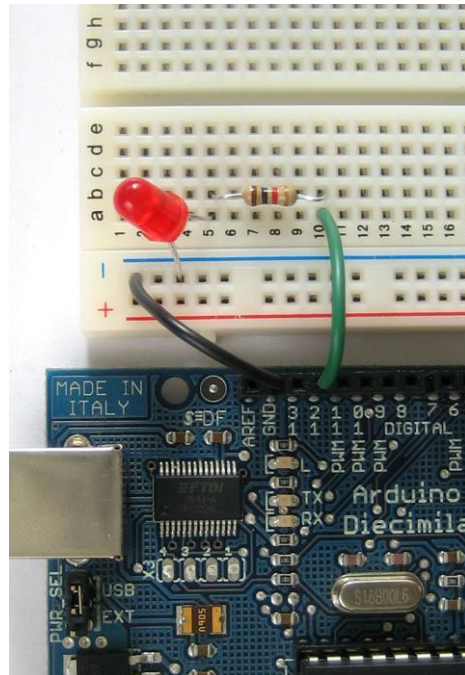
Fade_a_completer$

#define led = A COMPLETER    // the pin that the LED is attached to
A COMPLETER brightness =A COMPLETER;

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

// the loop routine runs over and over again forever:
void loop()
{
  analogWrite(led, A COMPLETER);
}


3                                     Arduino/Genuino Uno on COM1
  
```



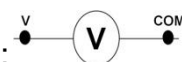
Faire le montage ci-dessus (prendre une sortie PWM !!) et valider votre programme.

**Q35. Quel appareil permet de mesurer une tension :**

**Q36. Pour mesurer une tension continue (selon le modèle de l'appareil) :**

- faut-il se mettre en position AC ou DC :
- faut-il sélectionner le symbole de gauche ou de droite  :

La représentation du branchement de l'appareil est la suivante :



Représenter (avec des flèches de couleurs) le branchement de l'appareil sur la photo ci-dessus pour mesurer la différence de potentiels (ddp) entre la sortie du signal et la masse en respectant les couleurs, V en **rouge** pour le signal et COM en **noir** pour la masse (GND).

**Q37. Compléter le tableau ci-dessous :**

Intensité lumineuse (%)	0	25	50	75	100
Variable brightness					



## ARDUINO : Sorties PWM

ddp entre sortie et masse théorique (V)	0				5
ddp entre sortie et masse mesurée (V)					

L'écart entre la ddp (la tension) mesurée et théorique s'appelle la tension de déchet.

En fait on mesure au voltmètre la valeur moyenne du signal.

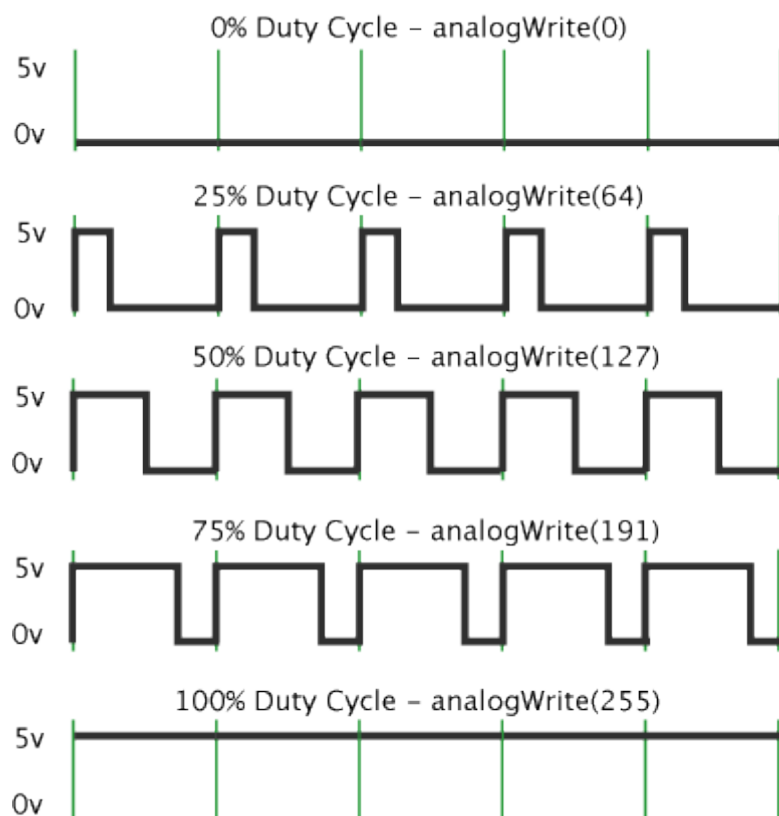
**Q38.** *Donner la traduction de write :*

analog est la contraction de analogique.

**Q39.** *En déduire la signification de `analogWrite(led, brightness)` :*

### a) Signal de sortie (sortie PWM) :

Le signal de sortie suivant la valeur du brightness (de 0 à 255) a cette forme :



On appelle cette forme de signal un signal PWM. La fonction devrait plutôt s'appeler PWMWrite (led,brightness). Si on devait utiliser la fonction analogWrite on devrait avoir analogWrite(led, 3,5) avec 3,5 pour avoir 3,5V en sortie.

Le terme "Analogique", utilisé par Arduino, est un abus de langage car c'est bien un signal numérique qui est produit par le microcontrôleur. Ce signal peut être assimilé à un signal analogique si on ne s'intéresse qu'à sa valeur moyenne. Un "vrai" signal analogique aurait une infinité de valeur, ici on peut "seulement" en avoir 256 (valeurs de 0 à 255).

# ARDUINO : Sorties PWM

**Q40.** Donner la signification (en anglais) de PWM :

**Q41.** Donner la traduction de PWM en français et son abréviation :

**Q42.** Donner la traduction de Duty Cycle :

Le rapport cyclique s'exprime par  $\alpha = \frac{t_{ON}}{T}$

**Q43.** Donner le nom et la définition de T :

**Q44.** Donner la définition de  $t_{ON}$  :

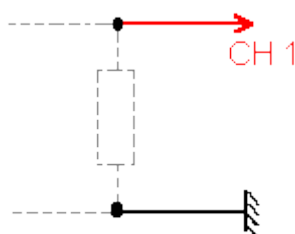
**Q45.** Compléter le tableau ci-dessous :

Intensité lumineuse (%)	0	25	50	75	100
Variable brightness					
ddp entre sortie et masse théorique (V)					
Rapport cyclique	0		0,5		
Rapport cyclique (%)	0				

Visualisation du signal de sortie à l'oscilloscope :

Vous allez visualiser le signal (PWM) précédent à l'aide d'un oscilloscope. Un oscilloscope est un instrument de mesure destiné à visualiser un signal électrique (le plus souvent variable au cours du temps).

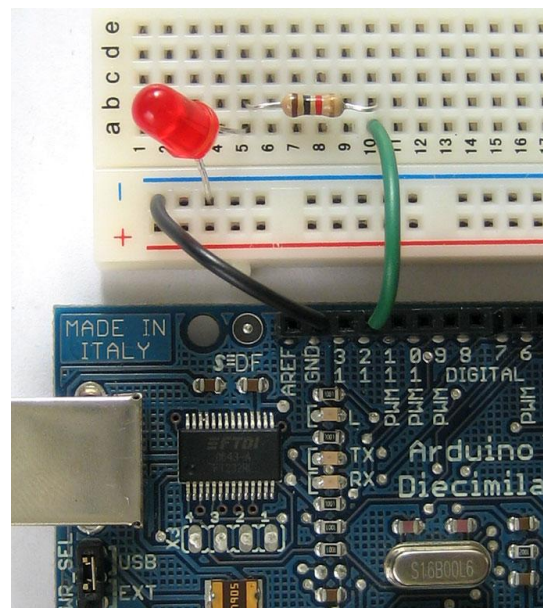
La représentation du branchement d'un oscilloscope est la suivante (exemple pour une résistance) :



CH est l'abréviation de channel. On trouve aussi Voie1 ou VoieA ou Y1 ou YA.

Donner la traduction de channel :

Représenter (avec des flèches de couleurs) le branchement de l'appareil sur

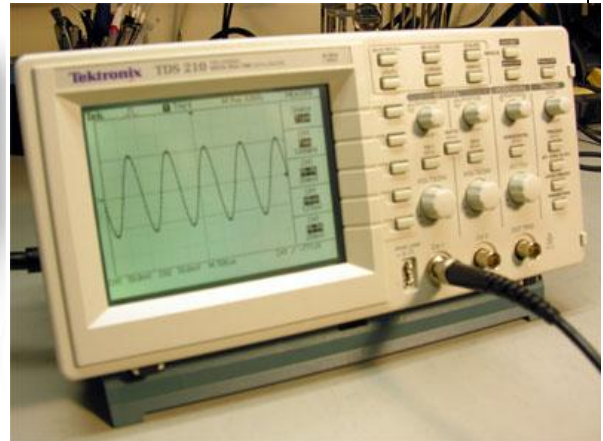




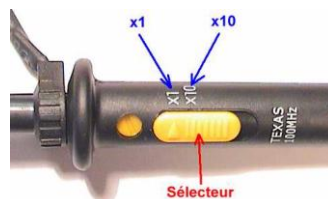
## ARDUINO : Sorties PWM

la photo ci-contre pour visualiser le signal en respectant les couleurs, **rouge** pour le signal et en **noir** pour la masse (GND).

Oscilloscope numérique



Pour relever le signal on utilise une sonde ou 2 cordons et un adaptateur BNC banane.



Sonde et zoom du sélecteur (à mettre sur x1)



Adaptateur BNC banane et cordons

Brancher l'oscilloscope et visualiser le signal (sortie PWM de la photo ci-dessus) en vous aidant du fichier "utilisation oscilloscope" (selon le type de votre oscilloscope). Vérifier que votre signal en PWM varie bien en faisant varier la valeur de la variable "brightness".

### IV. Modification du programme "fade":

Câbler sur une autre patte que la 13 une led et sa résistance puis vérifier le bon fonctionnement.

Brancher l'oscilloscope et visualiser le signal (sortie PWM) en vous aidant du fichier "utilisation oscilloscope" (selon le type de votre oscilloscope).

**Q46.** *Comment varie la PWM du signal ?*

--

### V. A retenir pour les sorties PWM

digital signifie numérique (0 ou 1) soit une tension de 0V ou 5V (uniquement 2 valeurs)

analog est la contraction de analogique soit une tension de 0V à 5V (une infinité de valeurs)

analogWrite(broche, val) : change la tension moyenne d'une des sorties marquées PWM. La valeur de la broche peut être 3, 5, 6, 9, 10 ou 11 et val est un nombre compris entre 0 et 255, correspondant à une tension de sortie de 0V à 5V.

Les connexions notées PWM peuvent fonctionner en "DIGITAL" et PWM : sinon la led L (relié à la broche 13) ne s'allumerait jamais.