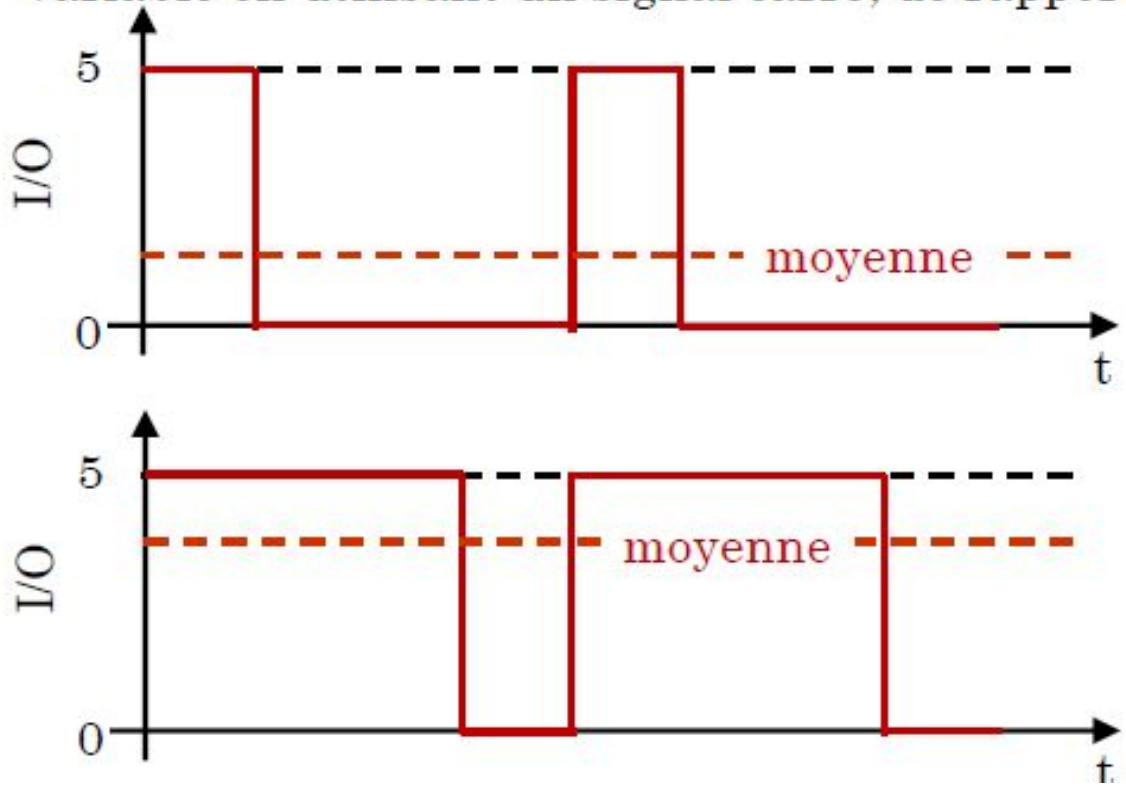
Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM)

Notion de PWM

- Il n'est pas possible d'obtenir une tension analogique en sortie de l'arduino (nous sommes restreint aux valeurs 0 V et 5 V)

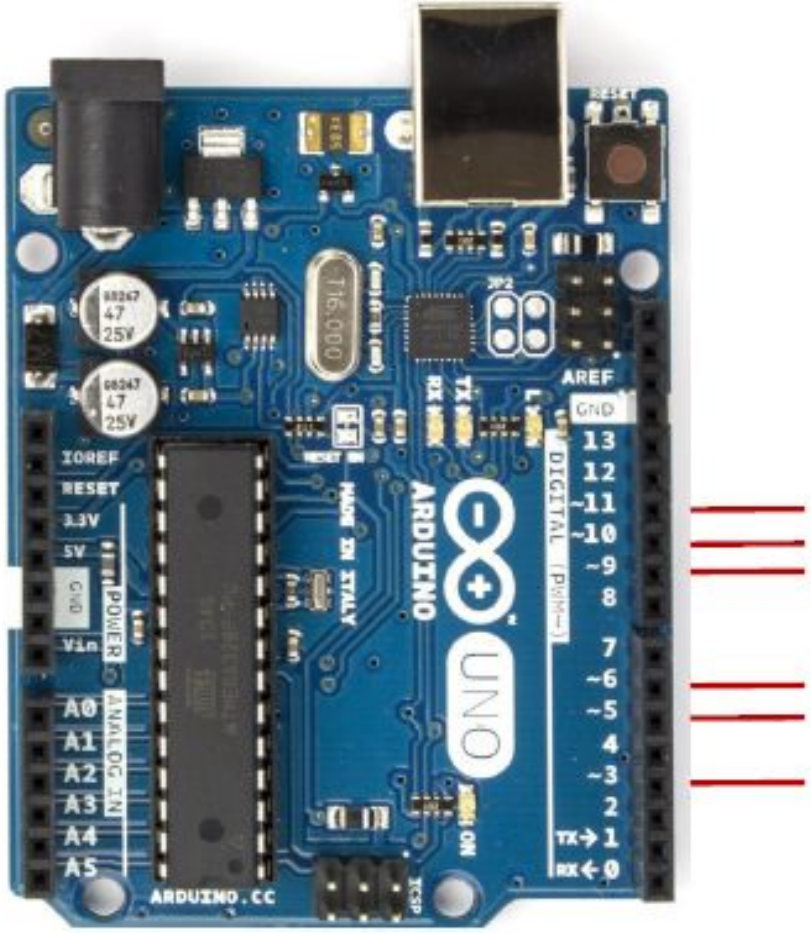
- Dans certains cas il est possible de faire comme si on avait une tension variable en utilisant un signal carré, de rapport cyclique variable



Pulse Width Modulation (PWM)

Sorties compatibles PWM de l'arduino

- Les I/O qui permettent d'utiliser les PWM sont précédées du signe « ~ »
- Il s'agit des I/O 3, 5, 6, 9, 10 et 11
- La fréquence d'horloge du PWM est de 490 Hz sauf pour l'I/O n°5 qui a une fréquence de 980 Hz.



Pulse Width Modulation (PWM)

Les nouvelles fonctions

- **analogWrite(X, Y)** : c'est la fonction dédiée au PWM de l'arduino. Il faut spécifier l'I/O X ainsi que la valeur du rapport cyclique Y. Y doit avoir une valeur comprise entre 0 et 255. Y = 0 : la sortie est toujours à 0. Y = 255 : la sortie est toujours à 5 V.

```
analogWrite(led_rouge, 100);
```

- **for (X=a; X<b; X++) {YYYYYY}** : cela s'appelle une boucle FOR et permet d'exécuter plusieurs fois les opérations YYYYYY. Pour compter les boucles, on utilise la variable X que l'on initialise à la valeur « a » et que l'on incrémente de 1 à chaque début de boucle (X++). On effectue la boucle tant que X est inférieure à « b »

```
for (i=0; i<=255; i++) {
    analogWrite(led_rouge, i);
    delay(10); }
```

Pulse Width Modulation (PWM)

Les nouvelles fonctions

- **millis()** : la carte arduino est dotée d'un chronomètre qui enregistre le temps écoulé depuis l'allumage de la carte. Le temps est obtenu en milliseconde et le temps maximum enregistrable est 50 jours soit 4.32 10^9 ms. Si le temps maximum n'est pas suffisant, il est possible de connecter à l'arduino une horloge extérieure alimentée par une pile.


```
chrono=millis(); //on mémorise la valeur du chronomètre
.....

if ((millis()-chrono)>500) { // durant plus de 500
        etat--; // on enlève 1 à la variable « état »
        delay(20);} // on règle la sortie de la boucle if
```

Pulse Width Modulation (PWM)

Les nouvelles fonctions

- **long** : en raison de la longueur du chiffre renvoyé par la fonction millis(), il faut stocker cette valeur dans une variable de type long. Cette variable est de type entier est va de –2 147 483 648 à 2 147 483 647.

- **unsigned** : pour arriver aux 50 jours, on enlève le signe de la variable long. Dans ce cas unsigned long va de 0 à 4 294 967 295.

Pulse Width Modulation (PWM)

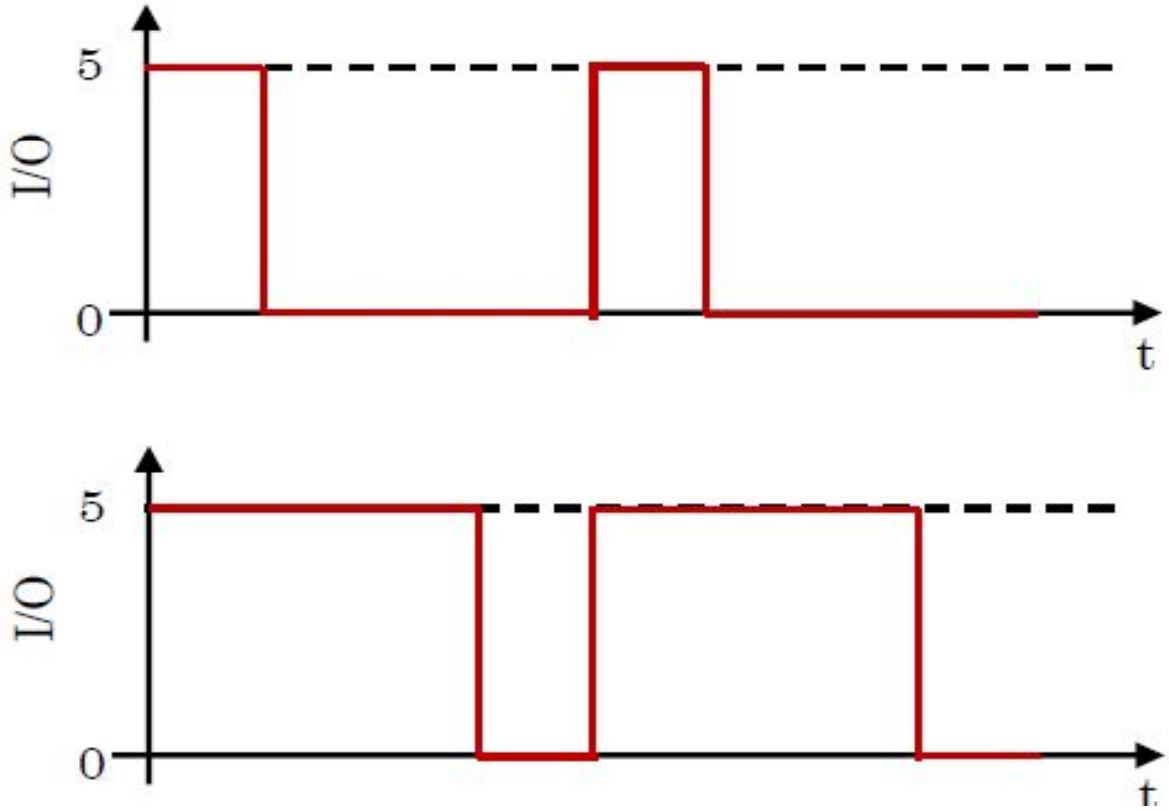
Exemple : luminosité d'une LED

- Pour modifier la luminosité d'une LED il faut soit modifier la valeur de la résistance mise en série avec la LED soit modifier la tension d'alimentation.
- Ces deux solutions ne sont pas envisageables simplement avec l'arduino
- Une solution consiste à utiliser la persistance rétinienne (utilisée par la Télévision). En effet si on allume et on éteint la LED très rapidement, on la verra toujours éclairée. Si on augmente le temps durant lequel la LED est allumée, on verra la LED plus brillante.

Pulse Width Modulation (PWM)

Exemple : luminosité d'une LED

- La variation du rapport cyclique du signal module la luminosité de la LED



Pulse Width Modulation (PWM)

Exemple d'un programme pour une LED avec PWM

```
int ledPin = 9;      // LED connected to digital pin 9

int analogPin = 3;   // potentiometer connected to analog pin 3

int val = 0;         // variable to store the read value

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

void loop()
{
  val = analogRead(analogPin);   // read the input pin
  analogWrite(ledPin, val / 4);  // analogRead values go from 0 to 1023, analogWrite values from
0 to 255
}
```