

TP 2 : Arduino plateforme : Analog Read,

Le langage C-ARDUINO

```
sketch_apr10a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Ici la configuration et l'initialisation

Ici le traitement

Les entrées analogiques

- La carte ARDUINO possède **6 entrées analogiques** (notées **A0** à **A5**) permettant de mesurer des tensions comprises entre **0V** et **5V** grâce à un convertisseur **ADC** à **N = 10 bits**.
- Donc un pas de quantification (quantum) est égal à:

$$q = \frac{|V_{ref}^+ - V_{ref}^-|}{2^N} = \frac{|5 - 0|}{2^{10}} = \frac{5}{1024} \approx 0.005 \text{ V/unité}$$

Exemple d'application numérique:

- Si la tension à l'entrée **A0** est **3.5 volts** donc la valeur numérique après conversion est :
 - **Vnum = ADC(A0)/q = 700.**
- Si la valeur numérique indiquée après conversion est **500** donc cette valeur correspond à une tension:
 - **Analog = Vnum*q = 2.75 volts.**

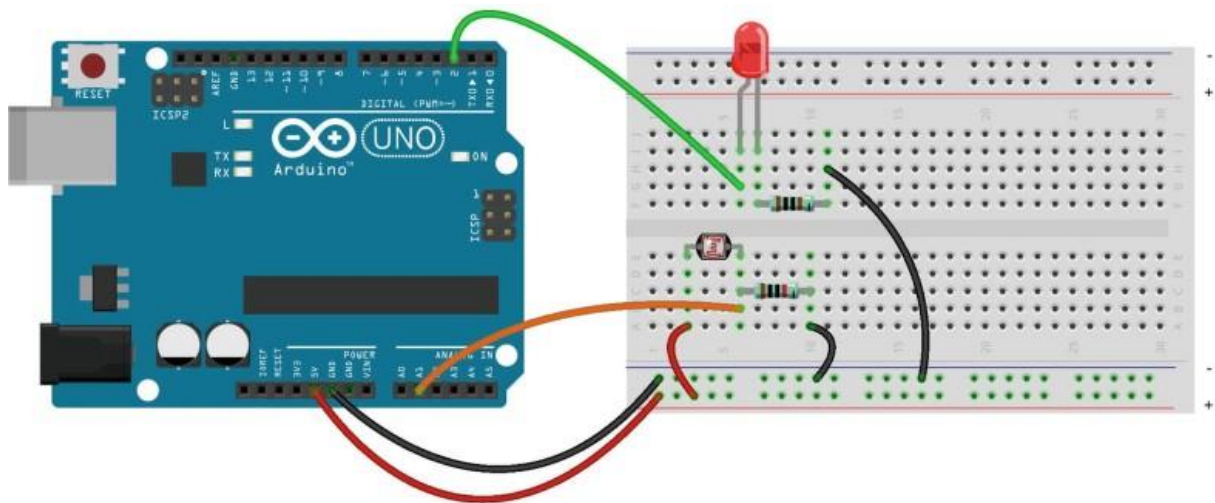
Le C-ARDUINO possède une fonction qui fait la lecture des entrées analogiques (`analogRead`) sur un temps de conversion égal à 100 μ s.

- Syntaxe: `analogRead(pin)`.
- `pin` : est une valeur entre 0 et 5 indique l'entrée analogique qu'on va lire.
- Cette fonction retourne une valeur numérique entre 0 et 1023.
- Exemple: lire l'entrée analogique A4.
- `Analog4 = analogRead(A4)` ;

Exemple : lecture d'une photo_resistance sur l'entrée A1.

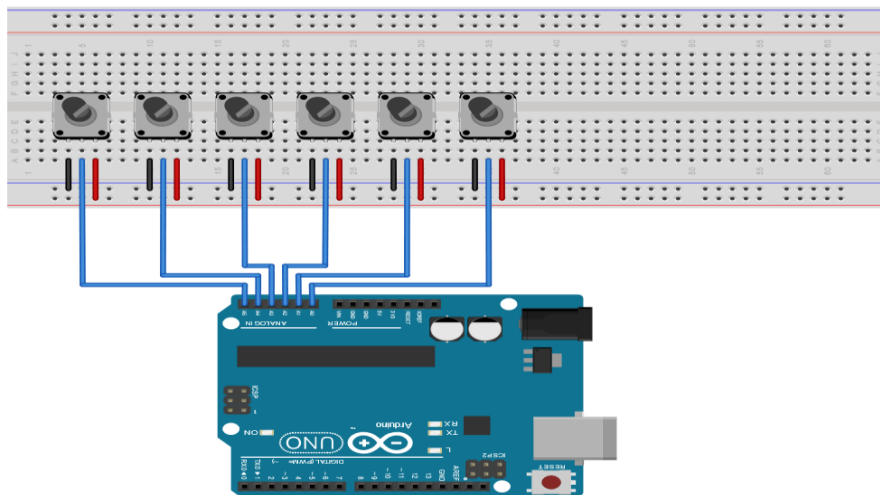
`int k` ;

`k=analogRead(A1);`



fritzing

Exemple 2 : lecture d'un potentiometre ;



A Faire :

Réaliser le système qui indique sur un panneau de LED le suivant:

- Si la tension $U < 1V$ allumer la LED sur pin2.
- Si la tension $1V \leq U < 2V$ allumer la LED sur pin3.
- Si la tension $2V \leq U < 3V$ allumer la LED sur pin4.
- Si la tension $3V \leq U < 4V$ allumer la LED sur pin5.
- Sinon allumer toutes les LED.
- Avec U est reliée à l'entrée analogique A1.

Faire un synoptique, un schéma électronique avec Arduino Uno comme plateforme et écrire un programme de fonctionnement. Quel sont les applications du système.

Note1 : structure Alternative dans C-Arduino

Syntax

```
if (condition1) {  
    // do Thing A  
}  
else if (condition2) {  
    // do Thing B  
}  
else {  
    // do Thing C  
}
```

```
}
```

Example Code

Below is an extract from a code for temperature sensor system

```
if (temperature >= 70) {  
    // Danger! Shut down the system. digitalWrite(12,  
LOW);  
}  
else if (temperature >= 60) { // 60 <= temperature <  
70  
    // Warning! User attention required.  
}  
else { // temperature < 60  
    // Safe! Continue usual tasks. digitalWrite(12;  
HIGH);  
}
```

Exp2 : If then Condition

Int x;

```
if (x > 120) digitalWrite(LEDpin, HIGH);  
  
if (x > 120)  
digitalWrite(LEDpin, HIGH);  
  
if (x > 120) {digitalWrite(LEDpin, HIGH);}  
  
if (x > 120) {  
    digitalWrite(LEDpin1, HIGH);  
    digitalWrite(LEDpin2, HIGH);  
}  
// all are correct
```

Note 2: Comparison Operators:

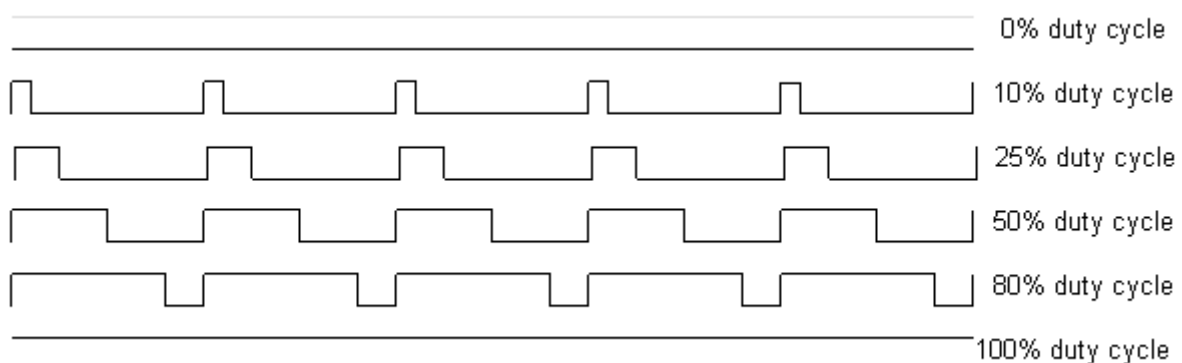
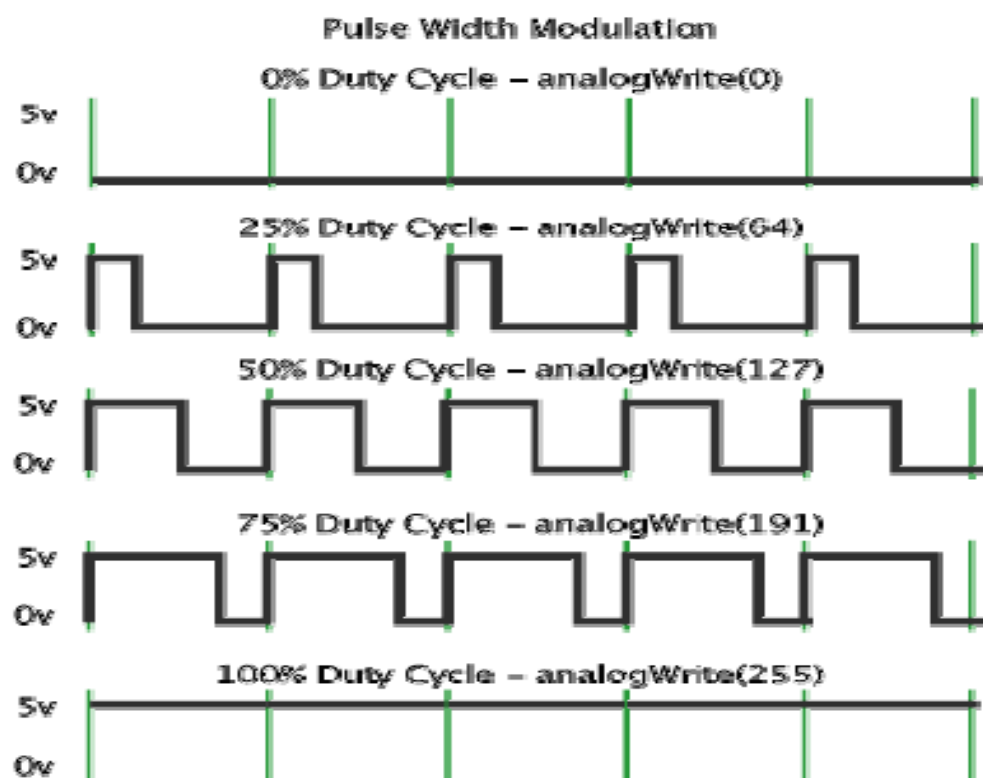
```
x == y (x is equal to y)
x != y (x is not equal to y)
x < y (x is less than y)
x > y (x is greater than y)
x <= y (x is less than or equal to y)
x >= y (x is greater than or equal to y)
```

TP 3 :Les sorties PWM (analogWrite)

Objectif : exploitation des sorties PWM pour control des moteurs ou générer signal analogique

La carte ARDUINO possède quelques lignes fonctionnent en mode PWM (Pulse With Modulation).

- Le signal PWM est un **signal carré** a une **fréquence fixe** avec un **rapport cyclique (DUTY CYCLE)** entre le niveau haut et la période du signal comme indique la figure suivante:



D'après la figure précédente: on utilise la fonction `analogWrite` pour générer un signal PWM.

- D'après le site officiel, la fréquence de signal PWM généré par les pins est:

- Sur une **UNO ATmega8** $F \approx 490$ Hz (pin9, 10, 11) [*Carte obsolète*].

- Sur une **UNO ATmega168/328** $F \approx 490$ Hz (pin3, 9, 10, 11) et $F \approx 980$ Hz (pin 5, 6).

Syntaxe: `analogWrite(pin, value)`.

- pin : est une valeur dans l'ensemble suivant {3, 5, 6, 9, 10, 11} pour la UNO indique la sortie PWM qu'on va utiliser.

- value: est une valeur entre 0 et 255 présente le rapport cyclique de signal PWM, dont 0 présente le rapport 0% et 255 présente le rapport 100% comme indique la figure du diapo 35.

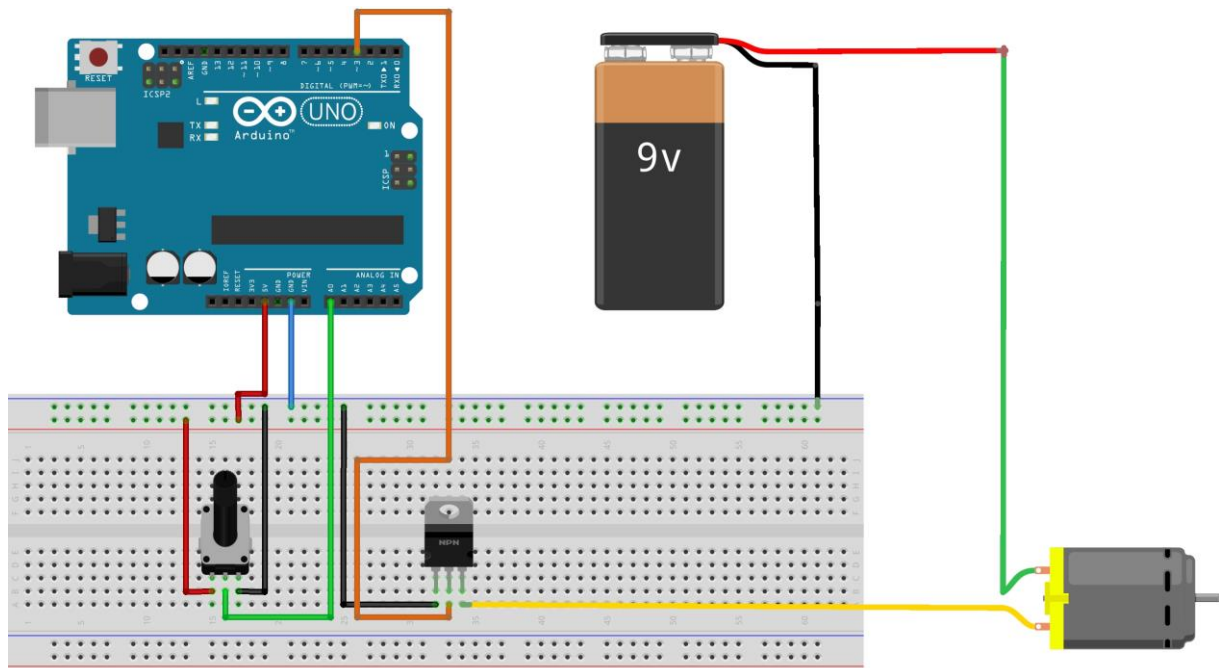
- Exemple: générer un signal PWM de rapport 75% sur la pin11.

100% -----255

75% -----x

$X = 75 * 255 / 100 = 191$

`analogWrite (11, 191) ;`



fritzing

```

/*
duty cycle in arduino is represented by a 8 bit value: 0 -
255
0 --> 0%
127 --> 50%
255 --> 100%
*/

#define pwm 3
#define pot A0

void setup() {
  // put your setup code here, to run once:
  pinMode(pwm,OUTPUT);
  pinMode(pot,INPUT);
  Serial.begin(9600);
  //safety speed reset of the motor
  analogWrite(pwm,0);
}

void loop() {
  // put your main code here, to run repeatedly:
  float val = analogRead(pot);
  float duty = map(val,0,1023,0,255);

```



```
analogWrite(pwm, duty); //here's how to generate PWM
signal from Digital arduino pin
Serial.println(duty);
}
```

A Faire TP 3 :

Exercice 1: Ecrire un programme qui fait le **changement** de la **luminosité** d'une **LED** par le biais d'un signal **PWM** (**pin10**) et l'**entrée** analogique **A0**, dont **0 numérique** indique l'état **OFF**, **1023 numérique** indique l'état **ON** et **511 numérique** indique une luminosité à **50%**.

Exercice 2 : Nous voulons contrôler la **vitesse** et la **direction** d'un moteur DC (pin 9), donner un schéma électronique avec carte Arduino Uno et écrire le programme.

Exercice 2 (***pour champion***) : comment extraire la composante continue du signal PWM (***génération d'un signal analogique à partir d'un signal numérique***) ?

Exercice 3 (***recherche Internet***):Quelles sont les applications du PWM en SE.