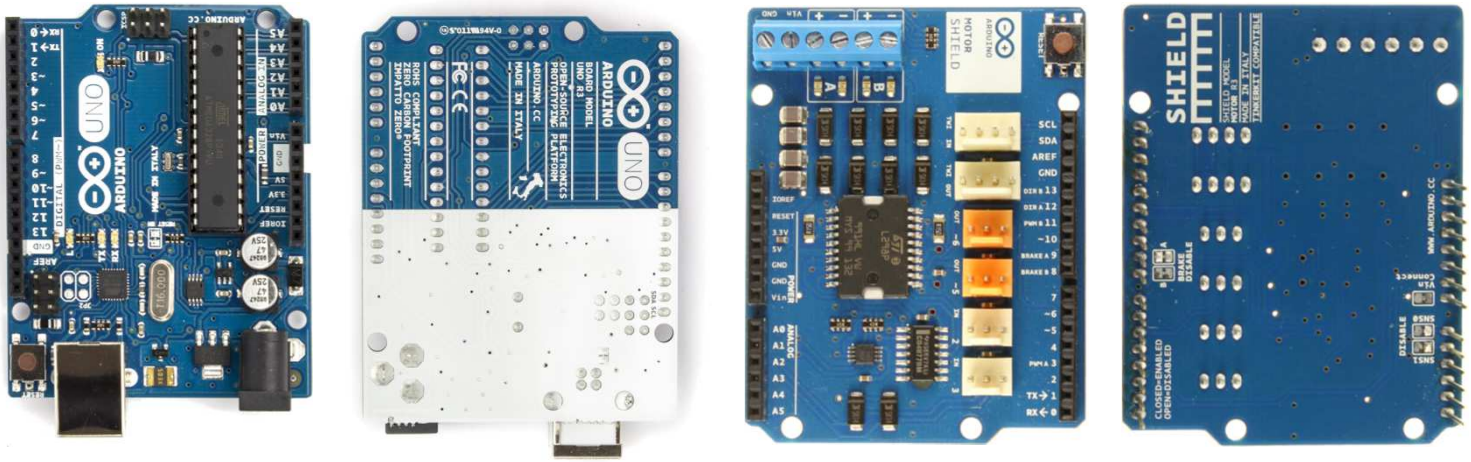
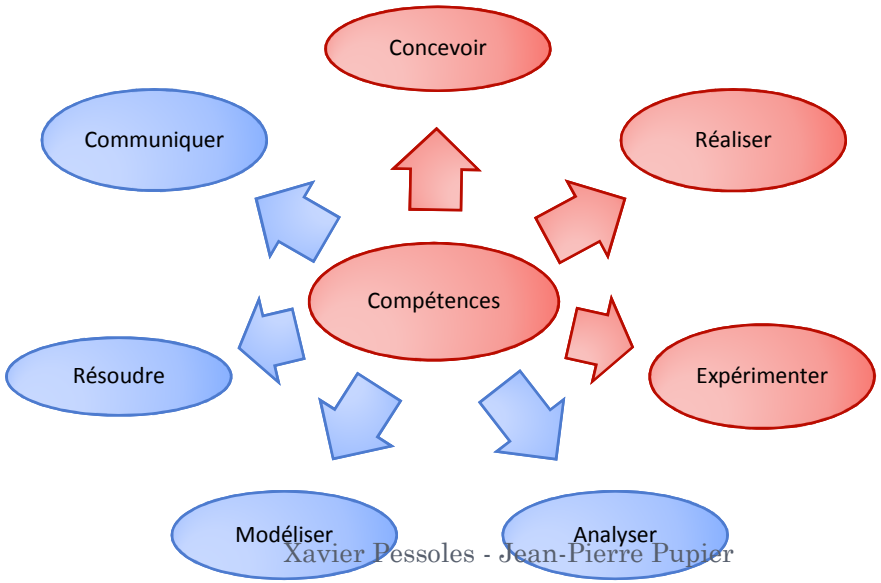


Acquisition et Pilotage – Cartes Arduino et Motor Shield



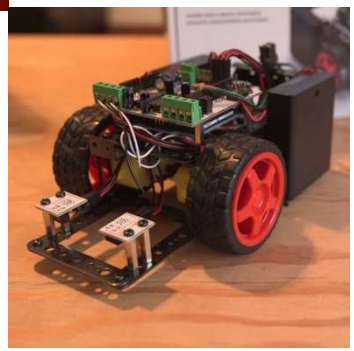
Xavier Pessoles - Jean-Pierre Pupier

Rouvière PTSI

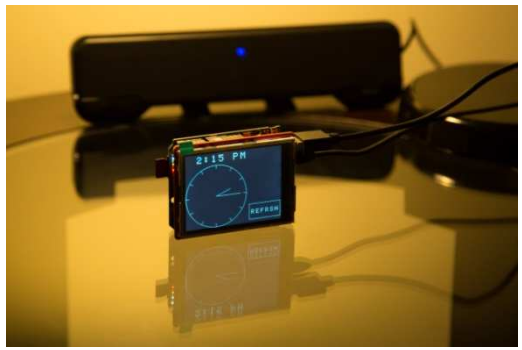
UNE CARTE ARDUINO POUR QUOI FAIRE …

- … pour réaliser des (mini ?) projets

Harpe laser
http://makezine.com/projects/laser-harp/



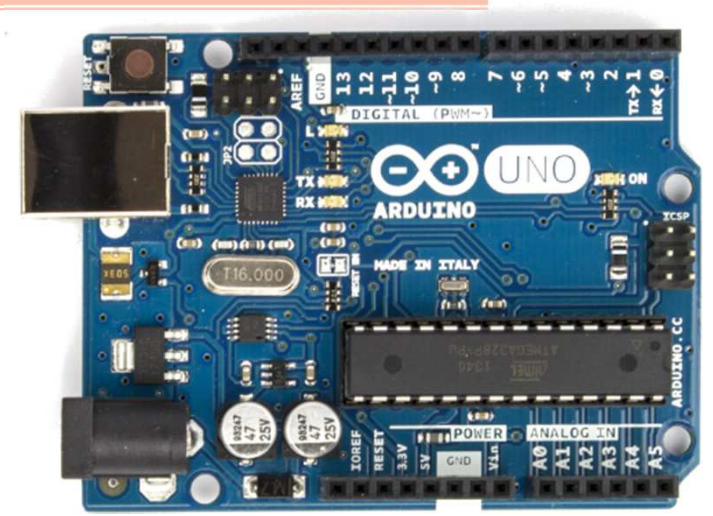
Joute robotique
http://makezine.com/video/ready-set-joust/

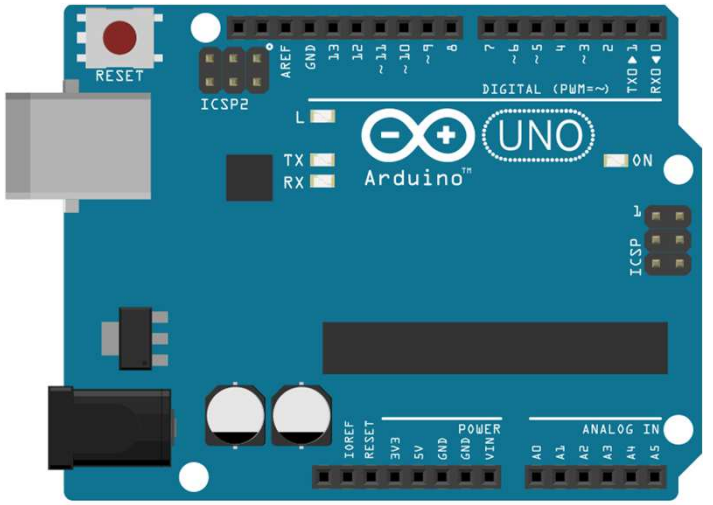


Réveil
http://makezine.com/video/never-forget-to-set-an-alarm-because-this-alarm-clock-sets-itself/

La carte Arduino UNO

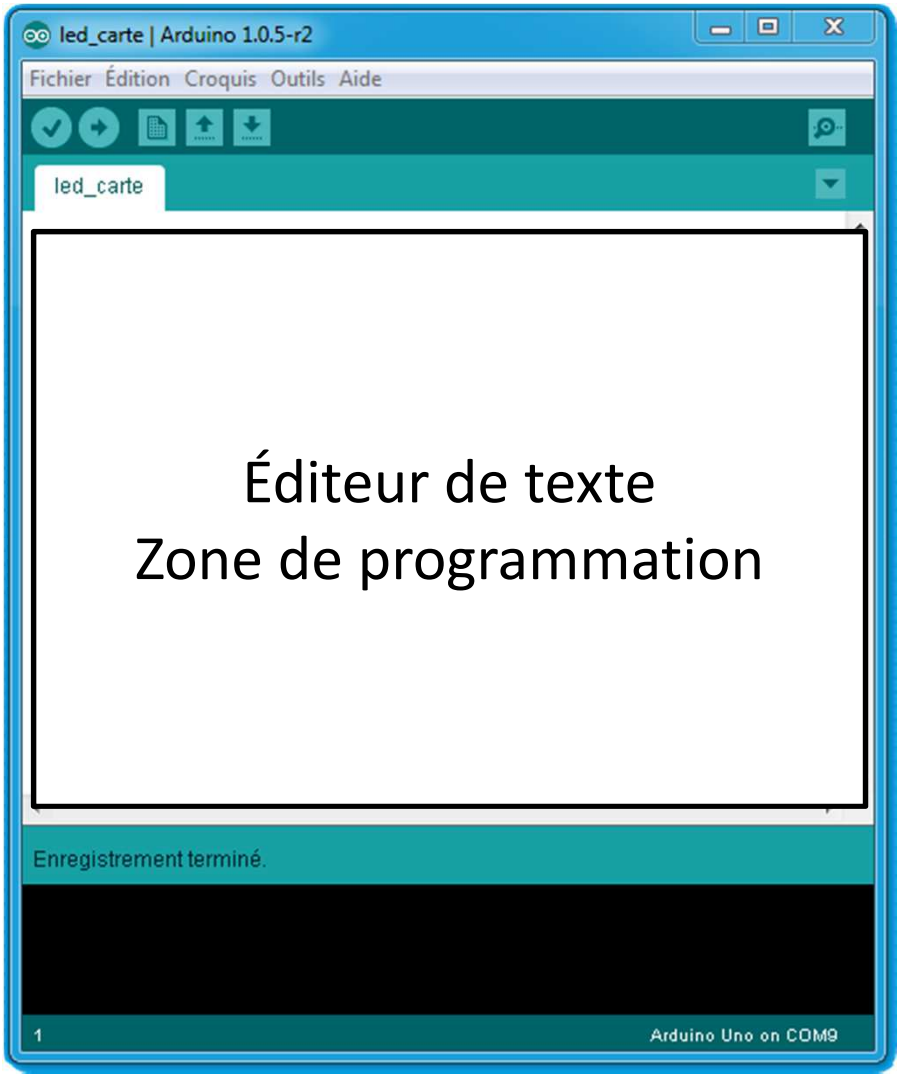
- Microcontroller : ATmega328
- Operating Voltage : 5V
- Input Voltage (recommended) : 7-12V
- Input Voltage (limits) : 6-20V
- Digital I/O Pins : 14 (of which 6 provide PWM output)
- Analog Input Pins : 6
- DC Current per I/O Pin : 40 mA
- DC Current for 3.3V Pin : 50 mA
- Flash Memory : 32 KB (ATmega328) of which 0.5 KB used by bootloader
- SRAM : 2 KB (ATmega328)
- EEPROM : 1 KB (ATmega328)
- Clock Speed : 16 MHz
- Length : 68.6 mm
- Width : 53.4 mm
- Weight : 25 g

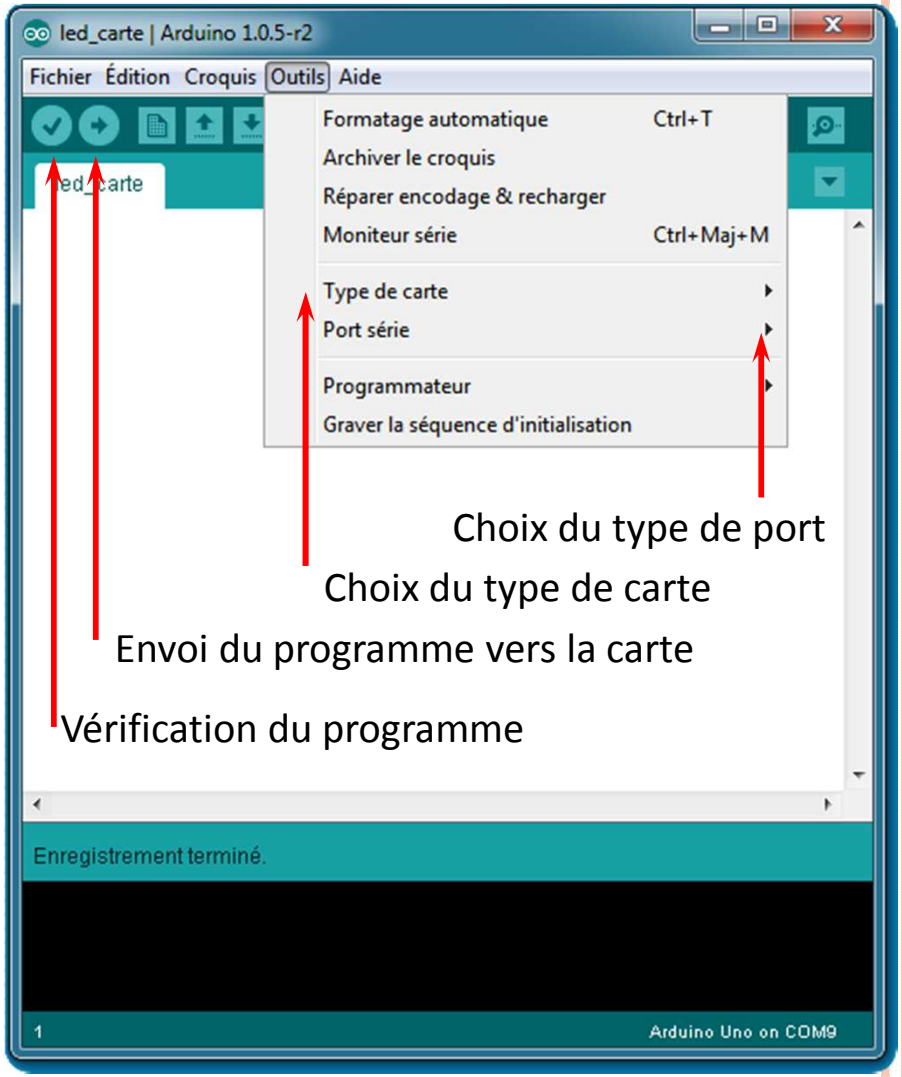




fritzing

L'INTERFACE LOGICIELLE





Première réalisation
Clignotement d'une led

Xavier Pessoles - Jean-Pierre Pupier

Allumer et éteindre une Led par période de 1 seconde

- A la différence de Python, il faut déclarer les variables et leur type.
- Les lignes doivent se terminer par des
- Void désigne la déclaration d'une fonction
- Pour réaliser un commentaire il faut faire précéder le commentaire de //

`int led = 13;`	Déclaration de la variable entière led et affectation du nombre 13 à la variable led. 13 correspond à une sortie de la carte Arduino possédant une led.

`void setup() {`
`  pinMode(led, OUTPUT);`
`}`
	Setup : configuration du matériel. Lancée après avoir appuyé sur Reset.
	La broche led est déclarée comme une sortie

`void loop() {`
`  digitalWrite(led, HIGH);`
`  delay(1000);`
`  digitalWrite(led, LOW);`
`  delay(1000);`
`}`
	La boucle loop est lancée un nombre infini de fois.
	
	Cette séquence permet d'allumer et d'éteindre la led toutes les secondes.

Travail à réaliser – Clignotement d'une LED

- A l'aide de l'image page 3 et de la carte situer la led
- Saisir le code suivant sur le logiciel Arduino
- Lancer la vérification
- Implanter le programme sur la carte
- Vérifier son bon fonctionnement

- Modifier le programme pour déterminer la fréquence correspondant à la persistance rétinienne.

Travail à réaliser – Clignotement d'une LED Affichage sur la console série

- La console série permet d'afficher sur l'écran de l'ordinateur des informations à destination de l'utilisateur
- Modifier le programme précédent de la manière suivante

```
int led = 13;
void setup() {
    pinMode(led, OUTPUT);
    Serial.begin(57600);
}
void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  Serial.print("JOUR \n");
  digitalWrite(led, LOW);
  delay(1000);
  Serial.print("NUIT \n");
}
```

- Modifier le code précédent en utilisant le code ci-contre
- Pour afficher les messages :
  - Menu Outils
  - Moniteur série

Acquérir les informations d'un potentiomètre rotatif

Xavier Pessoles - Jean-Pierre Pupier

REMARQUE

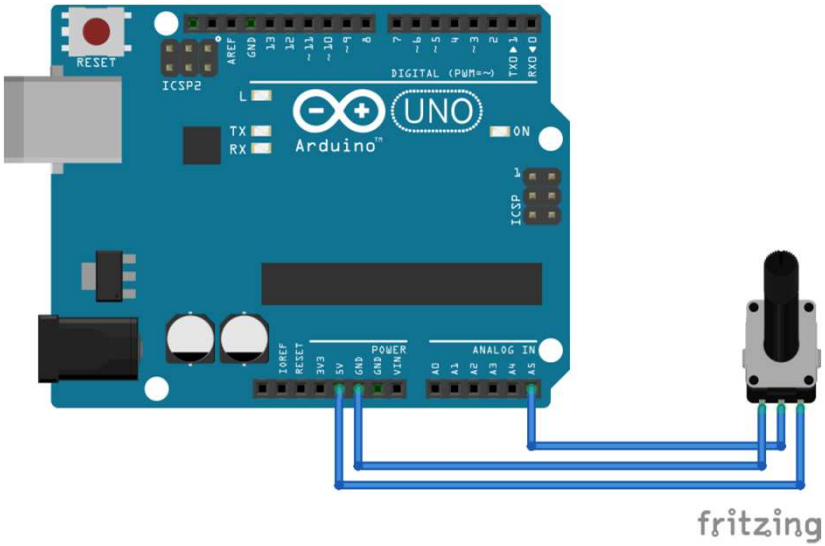
DEBRANCHER LA CARTE ARDUINO DU PORT USB AVANT TOUT BRANCHEMENT (POTENTIOMETRE, BATTERIE, CARTE SHIELD…)

Spécifications du Potentiomètre

- Omeg PC20BU-47K-Lin
  - Type de potentiomètre : axial, monotour
  - Résistance : 47 kΩ
  - Puissance = 0,4 kW
  - Tolérance : ±20%
  - Diamètre de l'axe : 6 mm
  - Caractéristique : linéaire
  - Montage : THT
  - Encombrement du corps : Diamètre 20mm x 10mm
  - Matériau de la piste : plastique
  - Type d'usinage de l'axe : lisse
  - Longueur de l'axe : 43 mm
  - Longueur du filetage : 7 mm
  - Trame des pistes : 5,08 mm
  - Course mécanique : 300°
  - Caractéristique des potentiomètres : mono
  - Tension de travail maxi : 500 VDC

Acquisition potentiomètre
Matériel et câblage

- Une carte arduino UNO
- Un potentiomètre rotatif



Travail à réaliser

- Réaliser le câblage
- Écrire et tester le programme

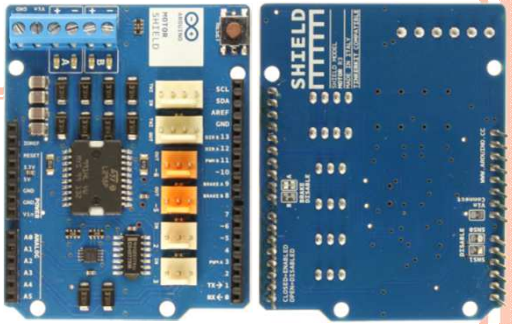
```
int IOcapteur = A5;    // Déclaration de l'entrée du potentiomètre
int valeur = 0;        // Déclaration de la valeur lue

void setup() {
  pinMode(IOcapteur, INTPUT);
  Serial.begin(57600);      // ouverture du port série
}

void loop() {
  valeur = analogRead(IOcapteur); // Lecture et affectation de la valeur
  Serial.print(valeur); // Affichage de la valeur
  Serial.println();
}
```

Pilotage d'un moteur

DEBRANCHER LE PORT USB

Xavier Pessoles - Jean-Pierre Pupier

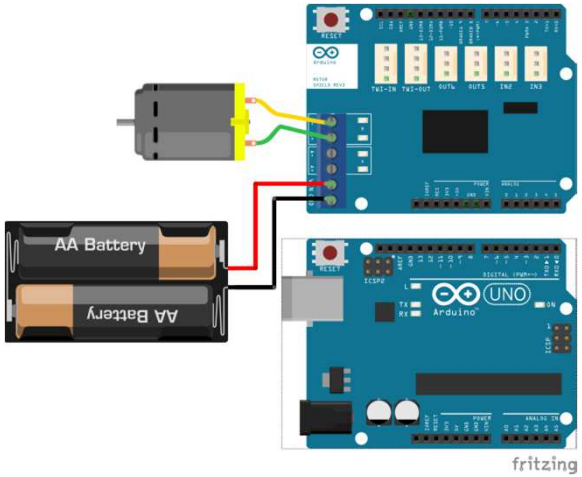
Spécifications du moteur et de la carte Motor Shield



- Matériel nécessaire :
  - Carte arduino UNO
  - Carte Motor Shield
  - Un moteur électrique
  - Batterie (attention, il n'y en a pas assez pour tous, mais elle n'est pas indispensable)
- Motor Shield
  - Operating Voltage : 5V to 12V
  - Motor controller : L298P, Drives 2 DC motors or 1 stepper motor
  - Max current : 2A per channel or 4A max (with external power supply)
  - Current sensing : 1.65V/A
  - Free running stop and brake function
- Moteur électrique à courant continu

Pilotage d'un moteur
Câblage et Programme

- Réaliser le câblage ci-contre. Les deux cartes doivent être branchées l'une sur l'autre.
- Pour piloter la sortie B du moteur, il faut utiliser la sortie PWM B à savoir la sortie 11.
  - Pour piloter la sortie PWM, il faut écrire un nombre de 0 à 255 qui sera proportionnel à la tension d'alimentation.
  - Exemple, si la batterie est de 6V et qu'on envoie le nombre 64 sur le PWM, le moteur sera alimenté à $6 \cdot \frac{64}{255} \simeq 1{,}5\ V$




```
int pwmPin = 11;    // LED connected to digital pin 11
void setup()  {
   pinMode(pwmPin, OUTPUT);
}
void loop()  {
  // fade in from min to max in increments of 5 points:
  for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=25) {
    // sets the value (range from 0 to 255):
    analogWrite(pwmPin, fadeValue);
    // wait to see the dimming effect
    delay(500);
  }
  // fade out from max to min in increments of 5 points:
  for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=25) {
    // sets the value (range from 0 to 255):
    analogWrite(pwmPin, fadeValue);
    // wait to see the dimming effect
    delay(500);
  }
}
```

Pilotage d'un moteur
Travail à réaliser

- Câbler et saisir le programme.
- Expliquer le déroulement du programme.
- Tester le programme

Pilotage d'un moteur en boucle ouverte

DEBRANCHER LE PORT USB

Xavier Pessoles - Jean-Pierre Pupier

Travail à réaliser

- A l'aide du travail réalisé trouver une solution permettant de faire en sorte que la vitesse du moteur soit fonction de l'angle du potentiomètre.
- Détecter la commande de seuil du moteur et ajuster votre programme.