BASE DE LA STRUCTURE

```
/* Etape 1 - déclaration des variables et des bibliothèques en début de programme. */

void setup() { // Etape 2 -
/* initialisation et configuration des entrées/sorties. Se lance seulement au début du programme. */
}

void loop() { // Etape 3 -
// partie principale. Se lance en boucle infiniment.
}
```

CRÉER UNE SOUS-PARTIE

```
void maFonction(){
/* mettre après « void loop() », cette partie de code ne bouclera pas */ }

maFonction();
/* renvoi à « void maFonction() » */
```

ENTRÉES/SORTIES

CONFIGURATION ENTREE/SORTIE (digitale seulement)

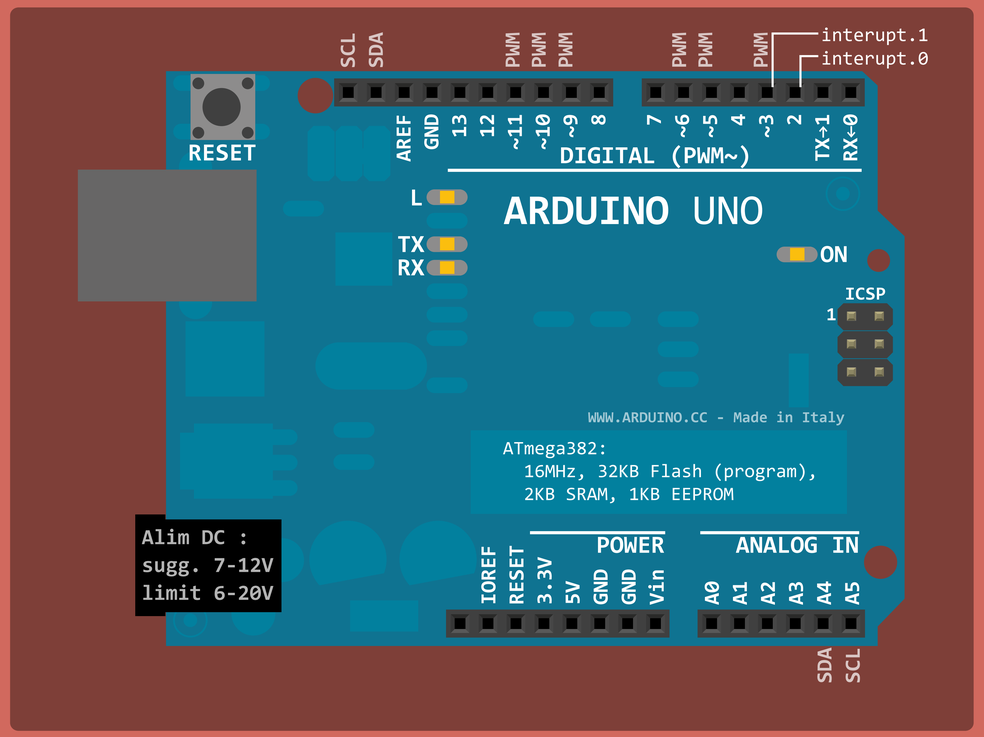
```
pinMode
(pin,          // num de la pin
INPUT          // pour une entrée
OUTPUT         // pour une sortie
INPUT_PULLUP   // pour entrée pullup
)
```

LIRE une entrée

```
... digitale // de la pin 0 à 13
var = digitalRead(pin)
... analogique // de la pin A0 à A5
var = analogRead(pin)
```

ECRIRE sur une sortie

```
... digitale // de la pin 0 à 13
digitalWrite(pin, HIGH/ou/LOW)
... analogique
// les pins 3 5 6 9 10 11
analogWrite(pin, valeur)
```



BIBLIOTHÈQUES

Charger une bibliothèque

```
#include<nomDeLaBibliothèque.h>
// inclure une bibliothèque
```

Exemple de bibliothèque (Servo)

```
#include<servo.h>
// inclure la bibliothèque servo
Servo monServo; // déclarer le servo
monServo.attach(pin);
// attacher mon servo à une pin
monServo.write(angle);
// indiquer un angle dans mon servo
```

INTERRUPTIONS

```
/* Pin2 = interuption 0 et
Pin3 = interuption 1 */
attachInterupt(1/ou/0, // pin2 ou 3
maFonction,// sous partie à appeler
LOW          // mode d'activation bas
FALLING      // mode relache bouton
RISING       // mode appui bouton
CHANGE       // mode changement d'état
)
```

VARIABLES, TABLEAUX, DONNÉES

Type de données

```
boolean   // Variable binaire ——————————→ 0 ou 1, false ou true
int       // Nombre entier (positif et négatif) ——————→ de -32768 à 32767
char      // caractère ASCII de 0 à 255 ——————→ ex : -128 - 127, 'a' '$' etc.
long      // entier de grandes valeurs ——————→ de -2147483648 à - 2147483647
float     // Nombre à virgule ——————————→ ex : 2,02 ou 58,0002
```

Tableau de caractères

```
char tabl[8] = "Arduino";
  // Nombre de caractères +1
char tabl[8] =
  {'A','r','d','u','i','n','o','\0'};
  // avec '\0' pour finir la chaine
```

Convertir un type de donnée

```
char(variable)   byte(variable)
int(variable)    float(variable)
long(variable)
```

Tableau de nombres

```
int mesInt[] = {40, 4, 58, 13, 6};
// ou
int mesInt[5];   // tableau de 5 ints

mesInts[0] = 42;
/* Assigner 42
à la 1ère entrée du tableau */

// ici, mesInts[3] est égal à 13
```

OPÉRATEURS

Opérateurs élémentaires

```
=   assigner une valeur      %   modulo
+   addition                 -   soustraction
/   division                 *   multiplication
```

Opérateurs composés

```
++  incrémenter    --  décrémenter
+=  addition composée
-=  soustraction composée
```

FAIRE DES COMMENTAIRES

```
Commentaire d'une ligne // je commente en une ligne comme ceci
Commentaire multiligne  /* je peux aussi commenter
                        en plusieurs linges comme cela */
```

CONDITIONS ET BOUCLES

Boucles

```
while(condition) // ex : var < 9
{ // répéter tant que c'est vrai }

for(init; condition; increment)
// ex: for(int i = 0; i <= X; i++)
{ // faire ceci «X» fois }
```

Conditions

```
if(condition) // ex : var = 2
{ // faire ceci une fois
}
else // sinon
{ //sinon faire cela une fois
}

switch(variable)
{ case 1 :
// si variable égale à 1 faire ceci
break;
case 2 :
// si variable égale à 2 faire cela
break;
}

break;      // quitter la boucle
```

Opérateurs de conditions

```
== égale à          != différent de
< moins que         > plus que
<= moins/égal à     >= plus/égal à
&& et               || ou
```

FONCTIONS UTILES

Mathématique

```
min(x, y)   max(x, y)
// calcule le min/max entre x et y
constrain(x, mini, maxi)
// contraindre x entre mini et maxi
sqrt(x)     pow(x, exposant)
// Racine carré et puissance de x
map(var de départ, val_basse_départ,
   val_haute_départ,
   val_basse_arrivée,
   val_haute_arrivée)
// Mise à l'échelle d'une variable
```

Hasard

```
randomSeed(A0)   // dans void setup
var =random(max) // de 0 à max-1
var =random(min, max)  //max exclut
```

Temps

```
delay(X);              // en millisec
variable = millis() /* temps depuis
lequel arduino est allumé */
```

Entrée/Sortie avancé

```
tone(pin, freq_Hz) // joue un son
tone(pin, freq_Hz, duration_ms)
noTone(pin)        // couper le son
variable = pulseIn(pin,
HIGH/ou/LOW, temps de calcule)
//calculer une pulsation

// bien d'autres fonctions existent
```

COMMUNICATION

```
Serial.begin(9600/ou/11500/ou/etc.);
/* démarre une liaison série sur une fréquence (mettre dans void setup) */
```

Communication vers la console série

```
Serial.print(«aaa»); // écrit aaa
Serial.print(var); // montre la var.
Serial.println(var); /* idem puis va à la ligne */
```

Communication entres objets

```
Serial.avalible();
/* donne le nombre de caractères arrivant sur port serie */
Serial.read(); // lire ce caractère
Serial.write(45);
// envoi 45 au port serie
Serial.write(«aaa»);
/* envoi le tableau de caractère aaa au port serie */
Serial.flush(); // vide le port serie
```

LAB FAB EDU LAB — UNIVERSITÉ RENNES 2

ARDUINO la Feuille de triche DE BASE

Visitez https://arduino.cc/fr/Reference/ Pour plus d'informations sur l'ensemble des fonctions Arduino.
Cette ***Cheat sheet*** est une fiche d'aide **en français** pour les utilsateurs d'Arduino à partir du niveau débutant !