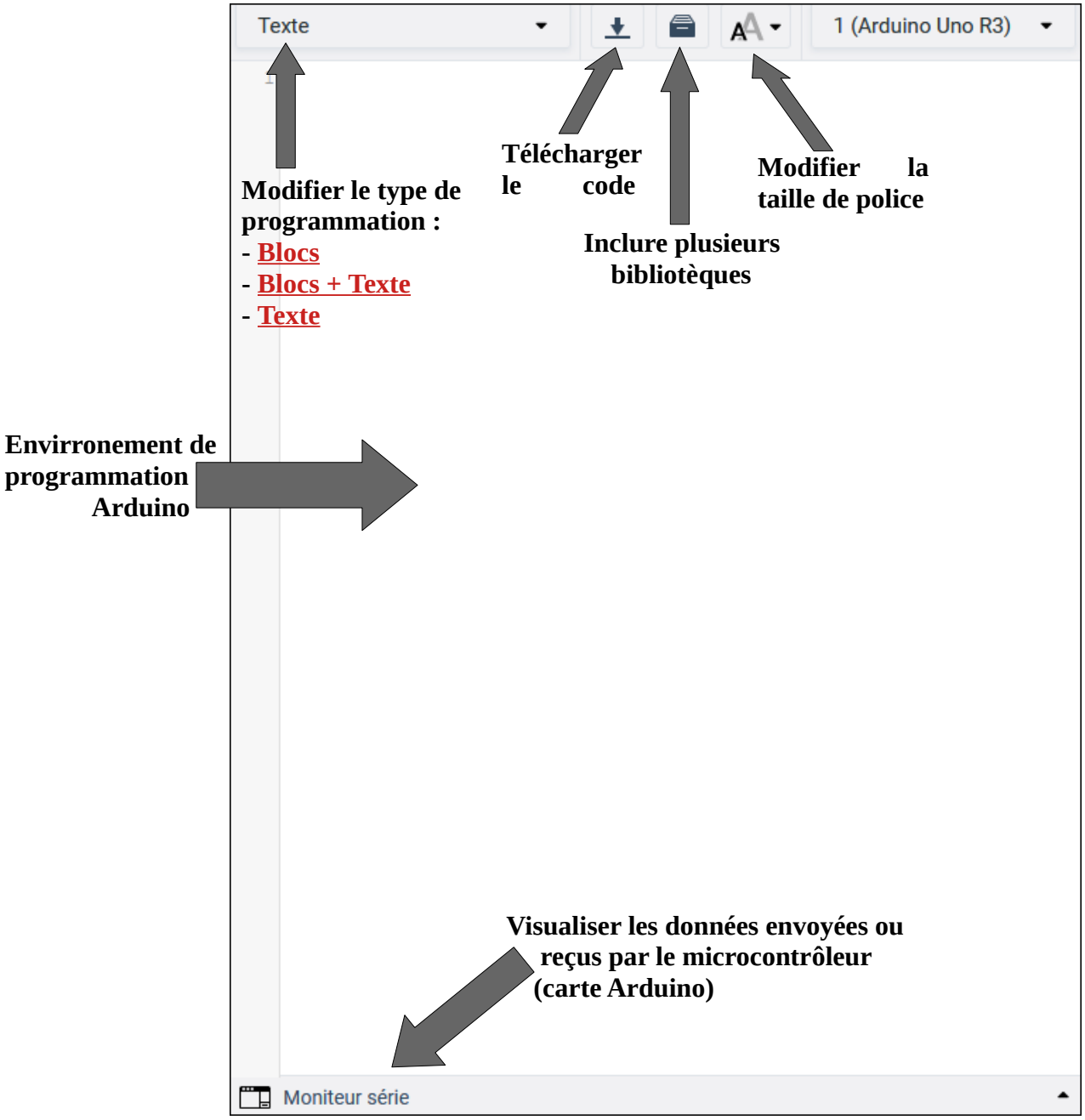
La programmation sur Thinkercad

Partie I : Définition

La programmation sur Thinkercad consiste à utiliser un environnement de développement en ligne pour créer des programmes qui contrôlent des circuits électroniques simulés.

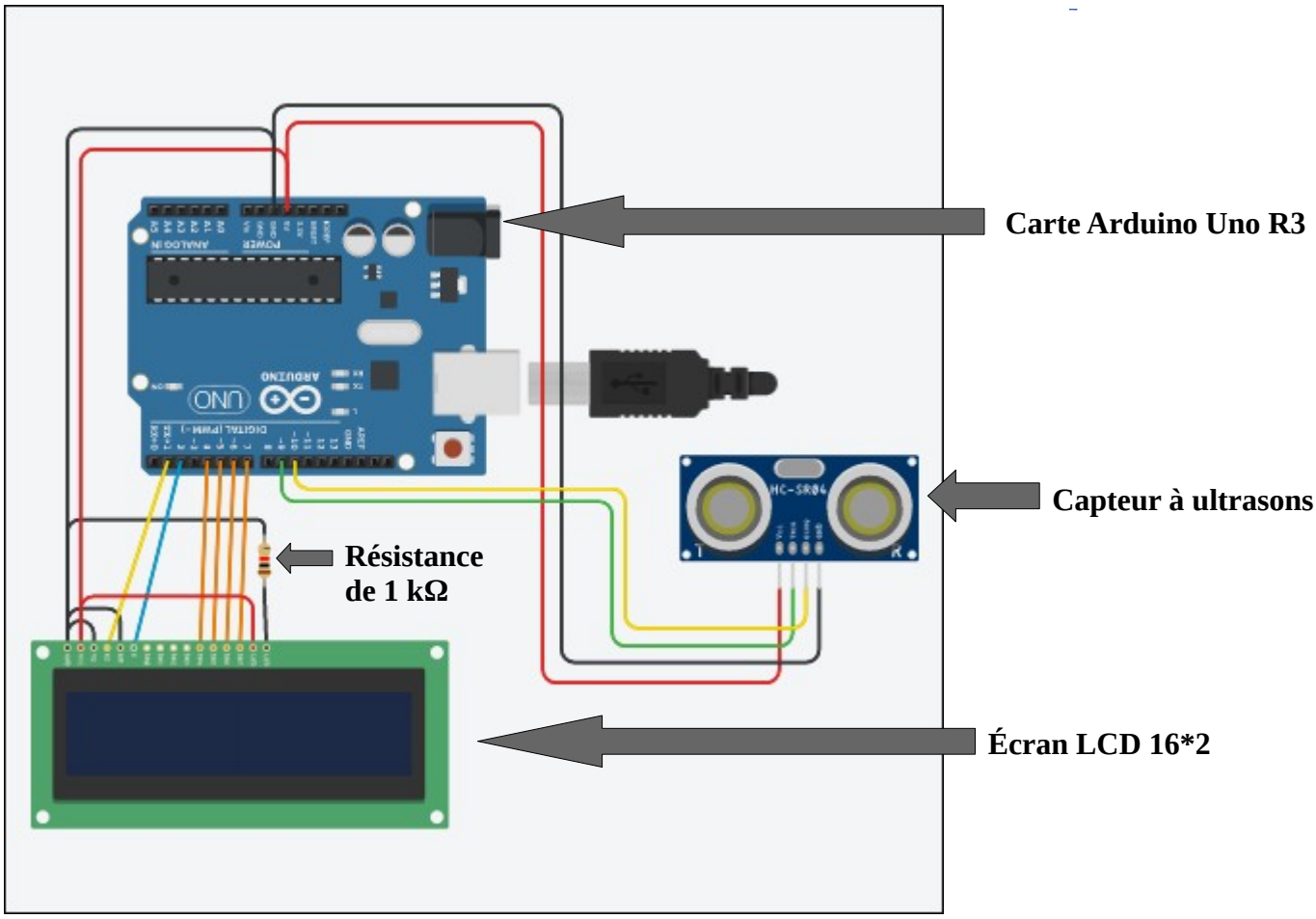
Partie II : Présentation de l'interface de programmation



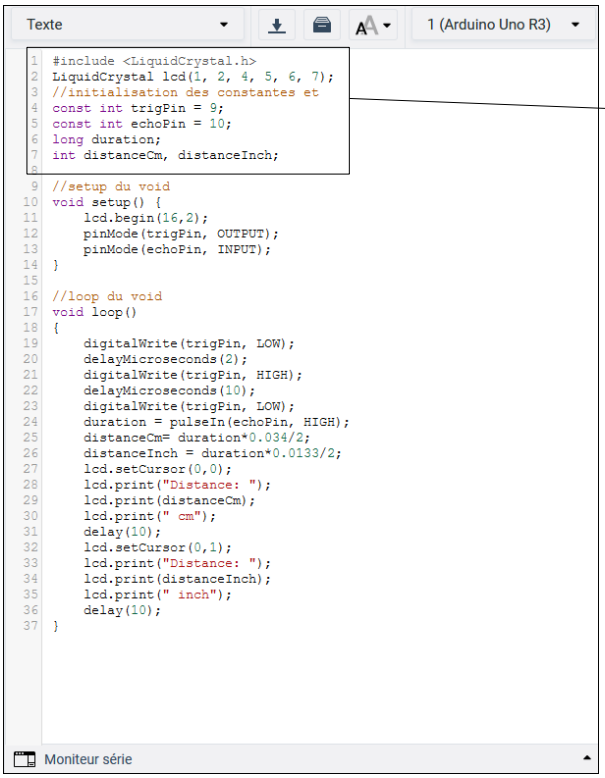
Partie III : Exemples de programmes de capteurs

1) Le capteur à ultrasons

a) Le circuit électrique



b) Le programme associé




```
#include <LiquidCrystal.h>
LiquidCrystal lcd(1, 2, 4, 5, 6, 7);
//initialisation des constantes et
const int trigPin = 9;
const int echoPin = 10;
long duration;
int distanceCm, distanceInch;

//setup du void
void setup() {
    lcd.begin(16,2);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
}

//loop du void
void loop()
{
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distanceCm= duration*0.034/2;
    distanceInch = duration*0.0133/2;
    lcd.setCursor(0,0);
    lcd.print("Distance: ");
    lcd.print(distanceCm);
    lcd.print(" cm");
    delay(10);
    lcd.setCursor(0,1);
    lcd.print("Distance: ");
    lcd.print(distanceInch);
    lcd.print(" inch");
    delay(10);
}
```

```
1  #include <LiquidCrystal.h>
2  LiquidCrystal lcd(1, 2, 4, 5, 6, 7);
3  //initialisation des constantes et
4  const int trigPin = 9;
5  const int echoPin = 10;
6  long duration;
7  int distanceCm, distanceInch;
```

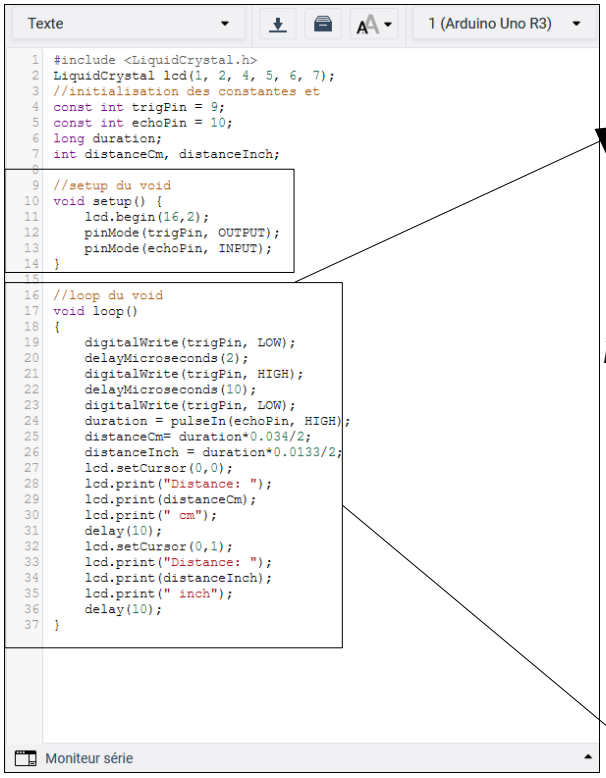
A- *Les lignes **1** et **2** servent à définir le type d'écran* **(*)** *ainsi que la bibliothèque utilisé.*

B- *Les lignes **4** et **5** servent à définir les constantes, en particulier ici le déclencheur (trig = trigger) et l'écho du capteur à ultrasons.*

C- *La ligne **6** sert à définir le stockage de nombres entiers sur 4 octets (= 32 bits), donc de -2 147 483 648 à +2 147 483 647).*

D- *La ligne 7 va définir en interne (int = internal) la distance dans une ou deux unités de longueur (comme le centimètre ou le inches (le pouce)).*

(*) - LCD = Liquid Crystal Display ; en francais, pour décrire un écran comme celui-ci, on dirait « écran ECL » qui veux donc dire Écran à Cristaux Liquide




```
 9  //setup du void
10  void setup() {
11      lcd.begin(16,2);
12      pinMode(trigPin, OUTPUT);
13      pinMode(echoPin, INPUT);
14  }
```

A- *La ligne 10 définit de manière général l'écran LCD et ses caractéristiques et les entrées et les sorties du « pinMode ».*

B- *La ligne 11 définit quand à elle, les dimensions de l'écran LCD, soit 16 caractères en longueur et 2 en largeur.*

C- *La ligne 12 et 13 sert à définir le « PinMode » du déclencheur (trigger) et de l'écho (écholocalisation[1]).*

```
16  //loop du void
17  void loop()
18  {
19      digitalWrite(trigPin, LOW);
20      delayMicroseconds(2);
21      digitalWrite(trigPin, HIGH);
22      delayMicroseconds(10);
23      digitalWrite(trigPin, LOW);
24      duration = pulseIn(echoPin, HIGH);
25      distanceCm= duration*0.034/2;
26      distanceInch = duration*0.0133/2;
27      lcd.setCursor(0,0);
28      lcd.print("Distance: ");
29      lcd.print(distanceCm);
30      lcd.print(" cm");
31      delay(10);
32      lcd.setCursor(0,1);
33      lcd.print("Distance: ");
34      lcd.print(distanceInch);
35      lcd.print(" inch");
36      delay(10);
37  }
```

A- *La ligne 17 contient toutes les fonctions à écrire sur le printer (ici l'écran LCD) et recueillit par les différents capteurs d'un système.*

B- *Les lignes 19 et 23 servent à écrire une valeur sur un Output (ex : un écran LCD).*[2]

C- *Les lignes 25 et 26 correspond à la relation distance = (vitesse du son*[3] *x temps) / 2.*

D- *Les ligne 27 à 36 correspondent aux print, c'est à dire à l'écriture sur un écran LCD, de une valeur de la distance trouver par le capteur, le « delay » correspond quand à lui, au délais entre chaque mesure du capteur, il vaut 10 millisecondes.*

---

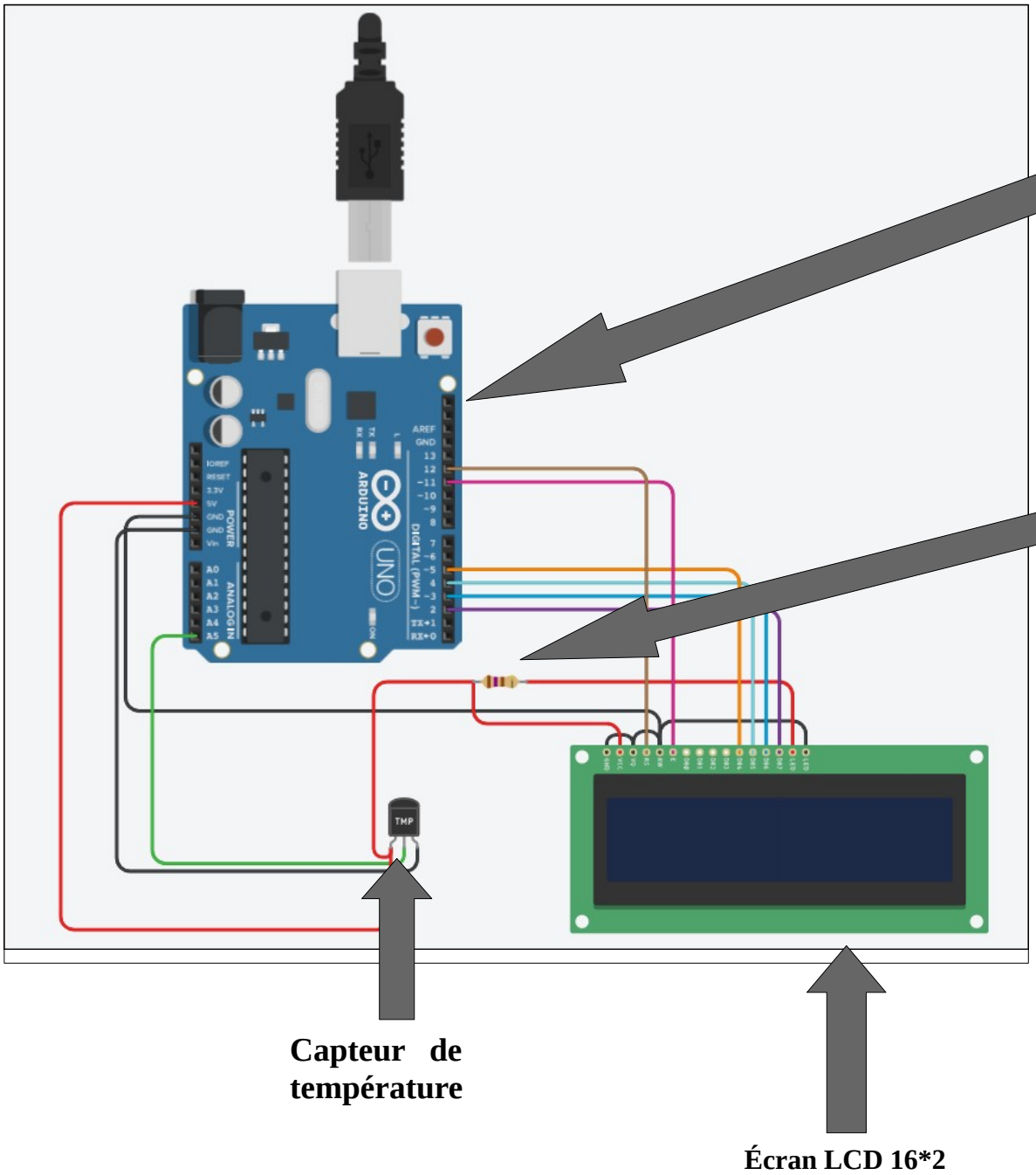
1 — L'écholocalisation est un procédé de localisation par les ondes sonores et plus généralement par ultrasons. Le système envoi un signal continue sur un objet et et suivant le temps que ce dernier va mettre pour revenir à son point de départ lorsqu'il aura toucher cet objet, on va pouvoir calculer à quelle distance il se situe de l'émetteur à ultrasons.

2 — Si le Pin est configurer sur l'OUTPUT, alors le voltage correspond à 5V (ou à 3,3V si c'est une carte de 3,3V) et sera noté HIGH et LOW pour 0V. Si le Pin est configurer sur l'INPUT, digitalWrite() sera activé (HIGH) ou désactivé (LOW).

3 La vitesse du son est de 340m/s ; Nous ne verrons pas le calcul pour la distance en inches.

2) Le capteur de température

a) Le circuit électrique

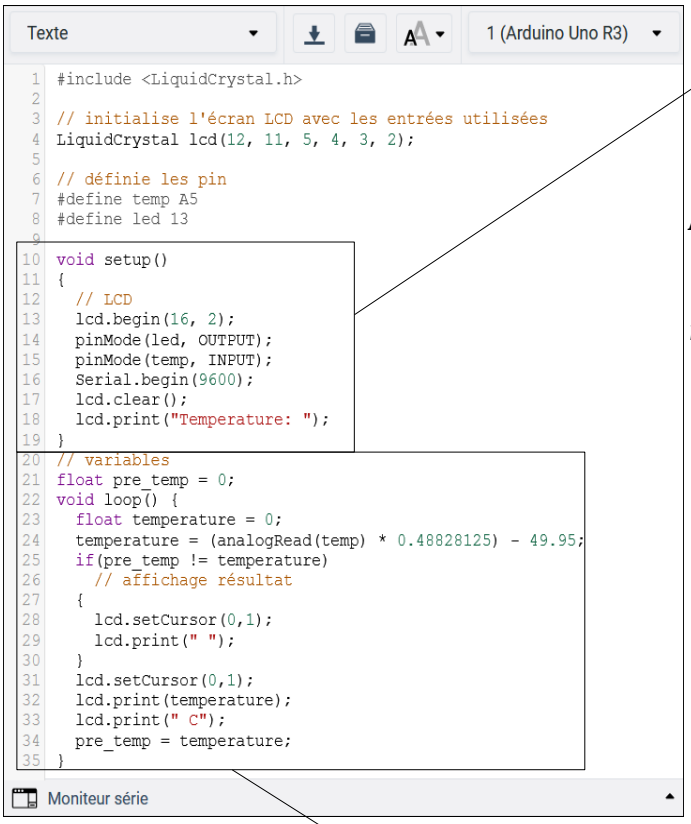


A noter que le calcul de la température étant complexe, nous ne l'aborderons pas ici.

b) Le programme associé

```
 9  void setup()
10  {
11    // LCD
12    lcd.begin(16, 2);
13    pinMode(temp, INPUT);
14    Serial.begin(9600);
15    lcd.clear();
16    lcd.print("Temperature: ");
17  }
```

A- *Ce bloc de code permet d'initialiser l'écran LCD en indiquant son PinMode.*
La ligne 15 permet de nettoyer l'écran de toutes inscriptions pour ensuite y afficher comme représenter ci-dessus, "Temperature: [variable de la temperature trouver par le capteur]"




```
18  // variables
19  float pre_temp = 0;
20
21  void loop() {
22    float temperature = 0;
23    temperature = (analogRead(temp) * 0.48828125) - 49.95;
24    if(pre_temp != temperature)
25      // affichage résultat
26    {
27      lcd.setCursor(0,1);
28      lcd.print(" ");
29    }
30    lcd.setCursor(0,1);
31    lcd.print(temperature);
32    lcd.print(" C");
33    pre_temp = temperature;
34  }
```

A- *La ligne 19 définit un float, c'est à dire une variable qui permet de socker des nombres décimaux tels que 3,14; 9,99 etc.*
B- *Les ligne 27 à 33 correspondent à l'affichage de la température finale après calcul ligne 23. (AnalogRead(temp) permet de lire la broche analogique de la température et de réaliser le calcule.*