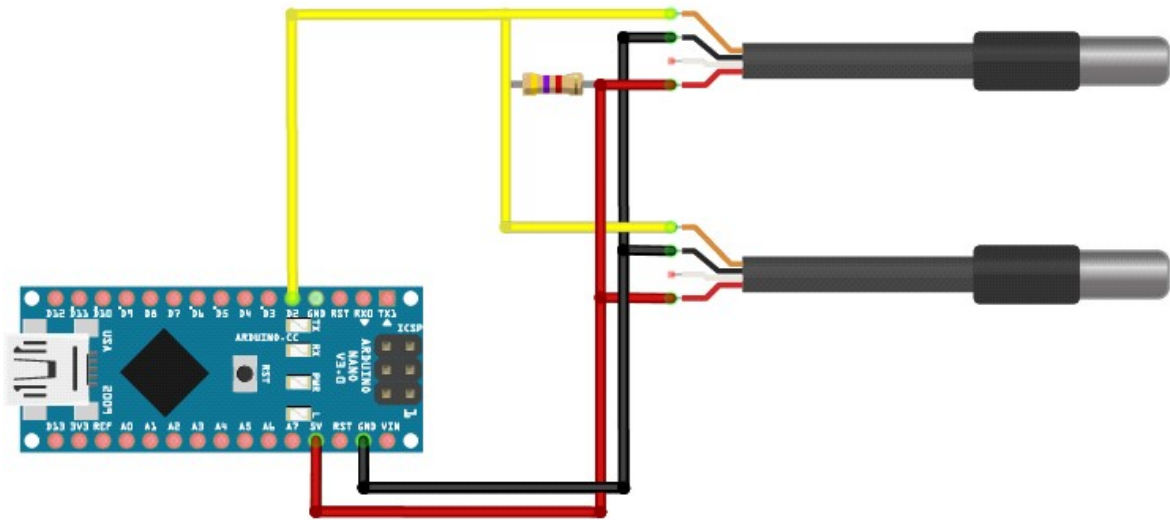


Dithermo sensor

[Plan de câblage / *Wiring diagram*](#)



fritzing

[Liste du matériel requis / *List of required equipment:*](#)

- x1 carte arduino nano v3.0
- x2 sonde DS18B20
- x1 résistance (*resistor*) 4,7kΩ
- x1 cable USB

[Informations complémentaires / *Further informations :*](#)

Les fils rouges et noirs servent à l'alimentation de la sonde (rouge sur + et noir pour la masse). L'alimentation est en 5V. Le fil jaune (sur le pin D2) sert pour les données.

Chaque sonde DS18B20 a une adresse différente par construction.

The red and black wires are used to supply the probe (red on + and black for ground).

The power supply is 5V. The yellow wire connected to digital pin 2 (D2 pin) is used for data.

Each DS18B20 sensor has a different address by construction.

dithermo.ino

```
#include <OneWire.h> //Librairie du bus OneWire
#include <DallasTemperature.h> //Librairie du capteur

OneWire oneWire(2); //Bus One Wire sur la pin 2 de l'arduino
DallasTemperature sensors(&oneWire); //Utilisation du bus Onewire pour les capteurs
DeviceAddress sensorDeviceAddress; //Vérifie la compatibilité des capteurs avec la librairie
int incomingByte = 0;
boolean value=false;
unsigned long lasttime;
const unsigned int ONE_SECOND=1000;
float buff0[100];
float buff1[100];
int nb_temp = 0;

void setup(void){
  Serial.begin(115200); //Permet la communication en serial
  sensors.begin(); //Activation des capteurs
  sensors.getAddress(sensorDeviceAddress, 0); //Demande l'adresse du capteur à l'index 0 du bus
  sensors.getAddress(sensorDeviceAddress, 1); //Demande l'adresse du capteur à l'index 0 du bus
  sensors.setResolution(sensorDeviceAddress, 12); //Résolutions possibles: 9,10,11,12
}

void loop(void){
  while (!value){
    Serial.println(7);
    delay(50);
    while (Serial.available()){
      incomingByte = Serial.read();
      if (incomingByte!=-1){
        value = true;
        lasttime=millis();
      }
    }
  }
  if (millis()-lasttime >= ONE_SECOND - 25) {
    float result0 = 0, result1 = 0;
    for (int i = 0; i < nb_temp; i++){
      result0 += buff0[i];
      result1 += buff1[i];
    }

    Serial.print(result0/nb_temp); //Récupération de la température en celsius du capteur n°0
    Serial.print(",");
    Serial.println(result1/nb_temp); //Récupération de la température en celsius du capteur n°1
    nb_temp = 0;
    lasttime = millis();
  }
  sensors.requestTemperatures(); //Demande la température aux capteurs
  buff0[nb_temp%100] = sensors.getTempCByIndex(0);
  buff1[nb_temp++%100] = sensors.getTempCByIndex(1);
}
```