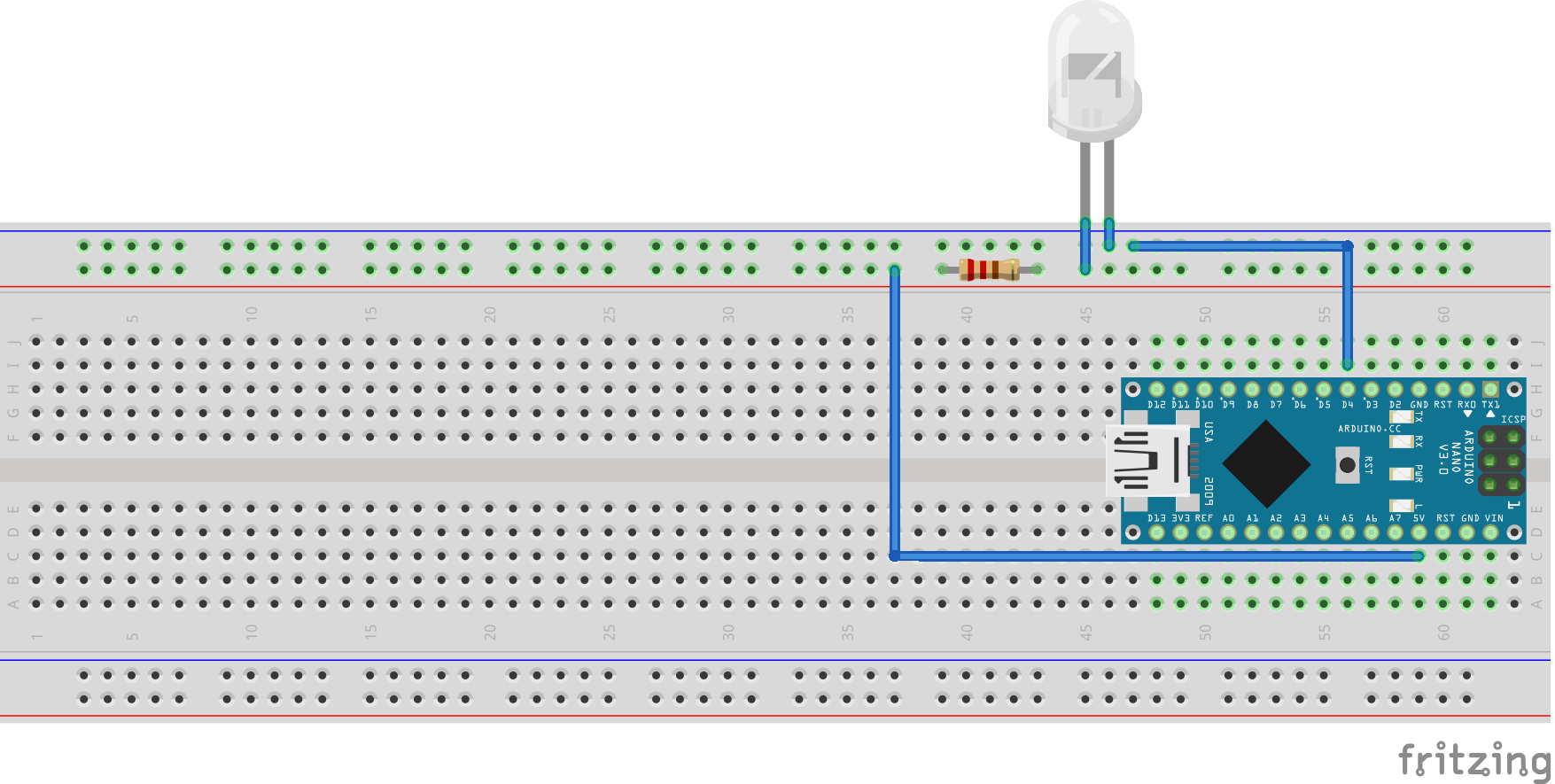
# Tâche 1 : Économie d’énergie par mise en veille.

Nom du projet : Low\_Power\_With\_Led

Lorsque l’arduino a fini l’intégralité de ses tâches, il doit passer en mode veille pour économiser le maximum d’énergie pour augmenter l’autonomie de la batterie.

## Architecture mise en place:

* 1 Led
* 1 Résistance 220 ohm
* 1 Arduino nano
* 2 Cables



## Librairies à utiliser:

Pour la mise en veille, la librairie LowPower de Rocket Scream est utilisée (<http://www.rocketscream.com/blog/2011/07/04/lightweight-low-power-arduino-library/>)

Car elle permet facilement de mettre en veille les différents modules de l’arduino nano pour ce type de puce (ATmega328P).

## Programmation:



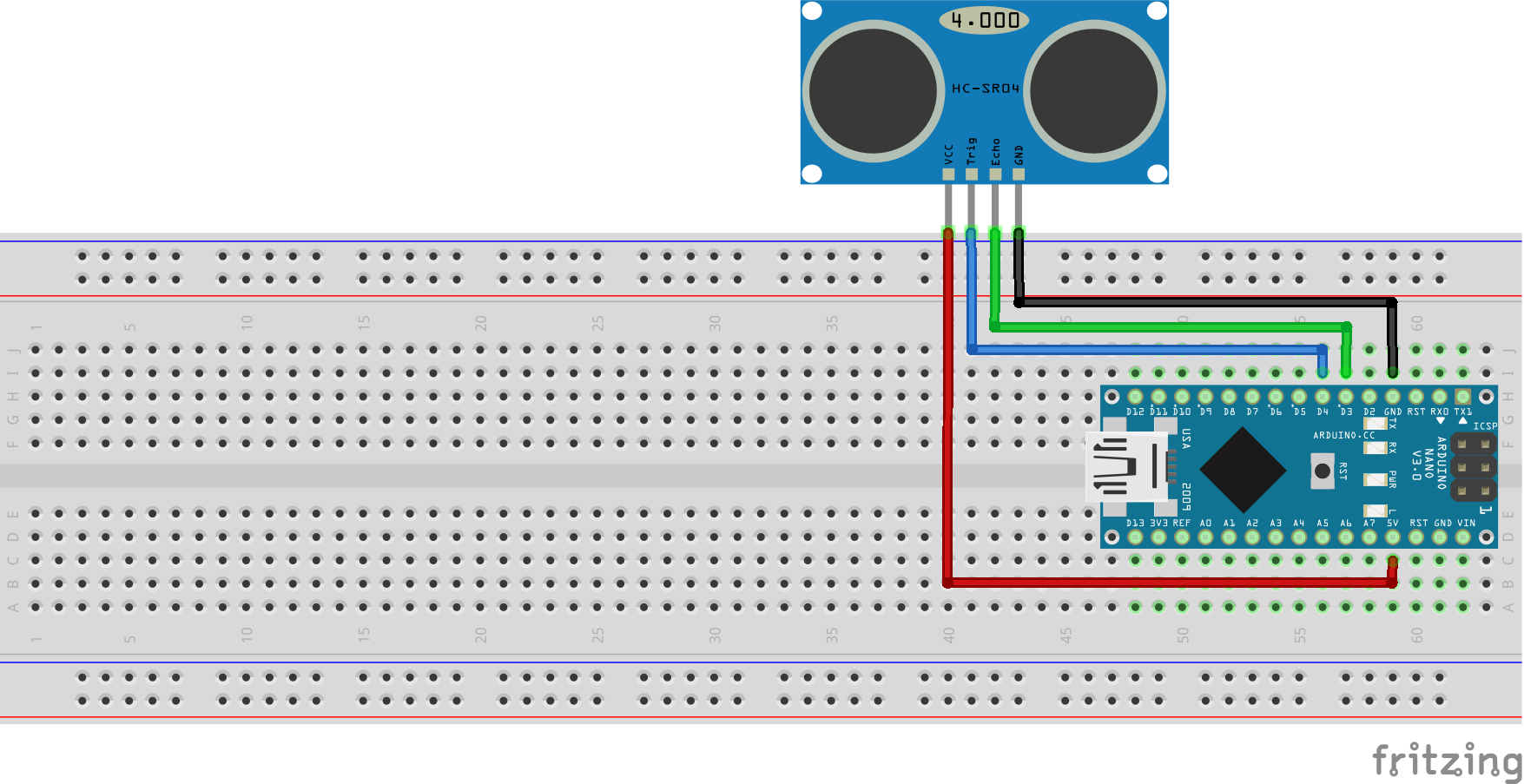
# Tâche 2 : Mise en place du capteur de proximité.

Nom du projet : ProximitySensor

Nous avons ajouté sur l’arduino nano le capteur de proximité.

## Architecture mise en place:

* 1 Capteur de proximité HC-SR04
* 1 Arduino nano
* 4 Cables



## Librairies à utiliser:

Nous avons créé nous même une bibliothèque permettant d’utiliser ce capteur.

## Programmation:

### Librairie perso:





### Programme principal:



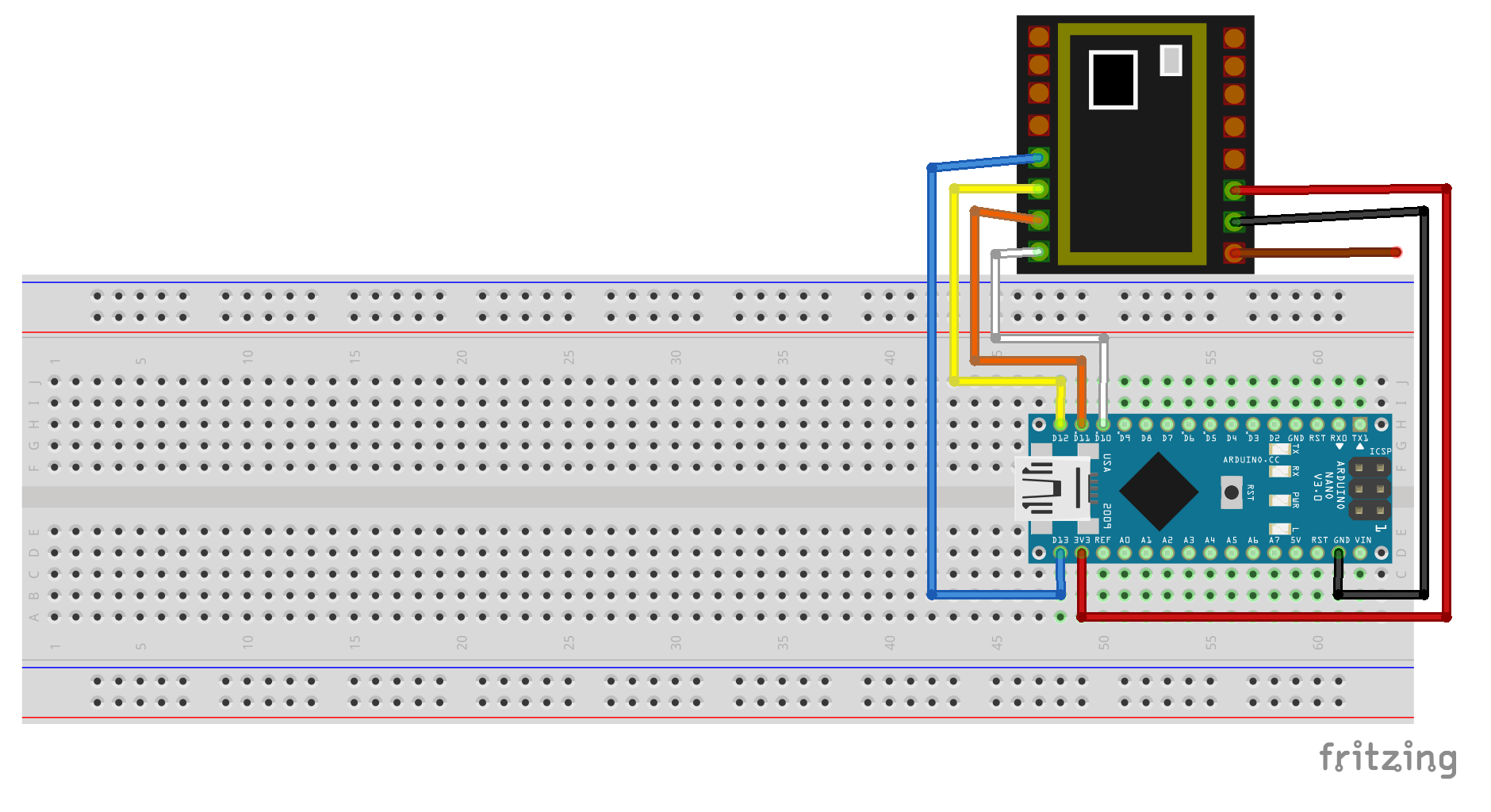
# Tâche 3 : Émission de données en Lora

Nom du projet : LoraEmit

Après récupération des données, l’arduino les envoi à l’aide du module LoRa.

## Architecture mise en place:

* 1 Puce LoRa
* 1 Antenne
* 1 Arduino nano
* 6 Cables



## Librairies à utiliser:

Pour l'émission, nous nous servons du code créé par Mr Pham sur son GitHub : <https://github.com/CongducPham/LowCostLoRaGw>

## Programmation:

Le code est disponible sur le GitHub de Mr Pham : <https://github.com/CongducPham/LowCostLoRaGw>

# Tâche 4 : Intégration des modules précedents.

Nom du projet : LowPowerAndEmitSensors

L’arduino récupère les informations du capteur de proximité et du capteur de température/hydrometrie et les envoi au travers de la puce Lora. L’arduino se met en Low power jusqu’à la fin de la temporisation.

## Architecture mise en place:

## Programmation: