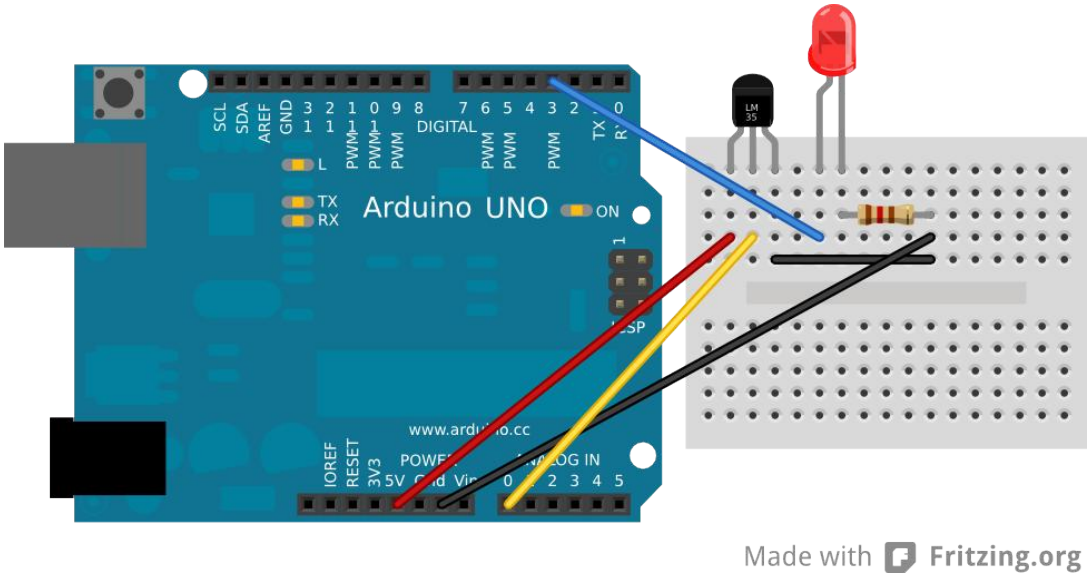

dessous :

ARDUINO	LM35
+5V	+Vs
A0	Vout
GND	GND

NOTE: Si vous connectez accidentellement le LM35 de la mauvaise manière, de capteur de température il devient un remarquable générateur de chaleur - la cicatrice en forme de D sur mon index gauche est un souvenir de ce jour mémorable...

ARDUINO	LED
D3	Anode (fil long)
A0	Cathode (fil court) (Reliée à la résistance de 120Ω)

Exécuter le programme
Saisissez le programme figurant à la page suivante, et sauvegardez-le en le nommant `arduino_firmata_tk.py`.
Connectez votre Arduino, ainsi que le circuit que vous avez câblé, puis lancez le programme comme ceci :

```
$ python arduino_firmata_tk.py
```

Après quelques secondes la fenêtre s'ouvre, quand c'est fait, vous pouvez contrôler la luminosité de la DEL avec le curseur sur l'écran.

Vous remarquerez que la DEL ne semble pas briller plus fort au-delà de 33% ; il n'y a pas de lien direct entre la luminosité et la tension de sortie PWM.

Si vous touchez le LM35, vous verrez l'affichage de la température changer. Les capteurs LM35 sont un peu bruyants, il est possible que les chiffres ne soient pas très stables.

Le code Python réalise 3 tâches principales

1. Initialiser la communication série et configurer les entrées/sorties de l'Arduino.

Deux détails importants à régler ici : mettre en place un thread itérateur pour empêcher le buffer série du Raspberry Pi de déborder lorsqu'il reçoit les données de l'Arduino, et ensuite attendre que l'Arduino envoie des données réelles avant de basculer au programme principal.

2. Définir trois fonctions qui seront appelées par GUI (interface graphique) :

get_temp() - lit la valeur sur la broche A0 de l'Arduino, la convertit en °C, et modifie l'étiquette en haut de la fenêtre.

Si vous êtes familier avec Arduino, pyFirmata retourne une valeur analogique entre 0.0 et 1.0 plutôt qu'un entier compris entre 0 et 1023.

Cette routine se termine en demandant au processus principal Tk, de se relancer après une demi-seconde (500 ms), c'est ainsi que nous rafraichissons l'affichage de la température, indépendamment de la boucle principale.

set_brightness() - convertit la valeur la valeur