

---

CPE Lyon - 4IRC - S7 - Année 2019/20  
Architecture et Protocoles Réseaux pour l'IoT  
**TP 2 - Interaction avec capteurs et acteurs depuis un  
micro-contrôleur**

---

Ce TP a pour but deux objectifs :

1. Connecter différents dispositifs aux interfaces d'un micro-contrôleur pour obtenir ses données ou interagir avec l'environnement.
2. Ce TP va vous permettre d'aller plus loin dans l'exploration des micro-contrôleurs en découvrant l'implémentation d'un contrôleur.

Dans ce TP vous allez tester plusieurs capteurs/acteurs à partir de votre micro-contrôleur RF Sub-1GHz, dans ce but vous allez continuer à explorer la documentation de ce module ainsi que les codes d'exemple présents dans le dépôt GIT (<https://github.com/CPELyon/modules-techno-innovmodules>).

*L'ordre des exercices proposés ici n'est pas strict, le nombre d'objets de chaque type étant limité, on ne peut pas travailler tous au même temps avec le même type de capteur.*

### Créer une nouvelle application

Pour chacun des exercices suivants vous allez créer une application. En fin de séance vous allez rendre sur la plateforme *e-campus* un seul fichier compressé avec chacune des applications que vous avez créées.

Pour créer une application, le plus simple est de copier une application existante (par exemple `apps/base/i2c_temp`) dans un nouveau dossier (par exemple `apps/base/feu`)

Le Makefile est ainsi déjà créé, et vous n'avez que le contenu du fichier "main.c" à modifier (vous pouvez le renommer sans incidence sur le Makefile).

Pensez aussi à changer le contenu du fichier README.

Pour plus d'informations, référez-vous à la documentation de l'API développée pour les modules : [http://wiki.techno-innov.fr/index.php/Technique/Logiciel/API/New\\_App](http://wiki.techno-innov.fr/index.php/Technique/Logiciel/API/New_App)

### Exercice 1. Feu de circulation

*Pour cet exercice vous allez récupérer 3 LEDs (rouge, orange, verte), 1 résistance, et 1 breadboard, voir la Figure 1.*

D'abord vous allez faire un montage des LEDs et la résistance dans votre breadboard. Pour ceux qui n'ont pas encore travaillé avec un breadboard, vous pouvez consulter la guide sparkfun sur <https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard>.

Le montage des LEDs et résistance vous pouvez le faire à votre préférence, une suggestion de montage est montrée dans la figure 2

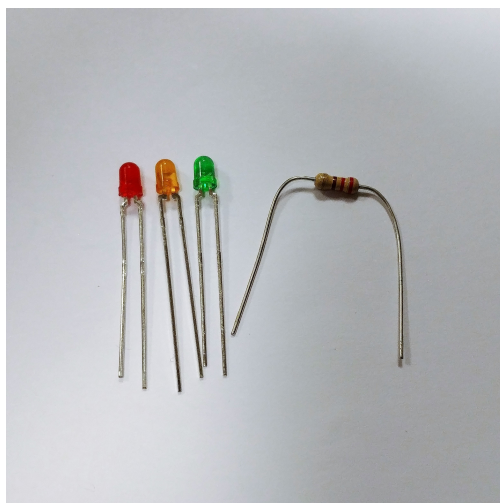


FIGURE 1 – 1 led rouge, 1 led orange, 1 led verte et ses résistances

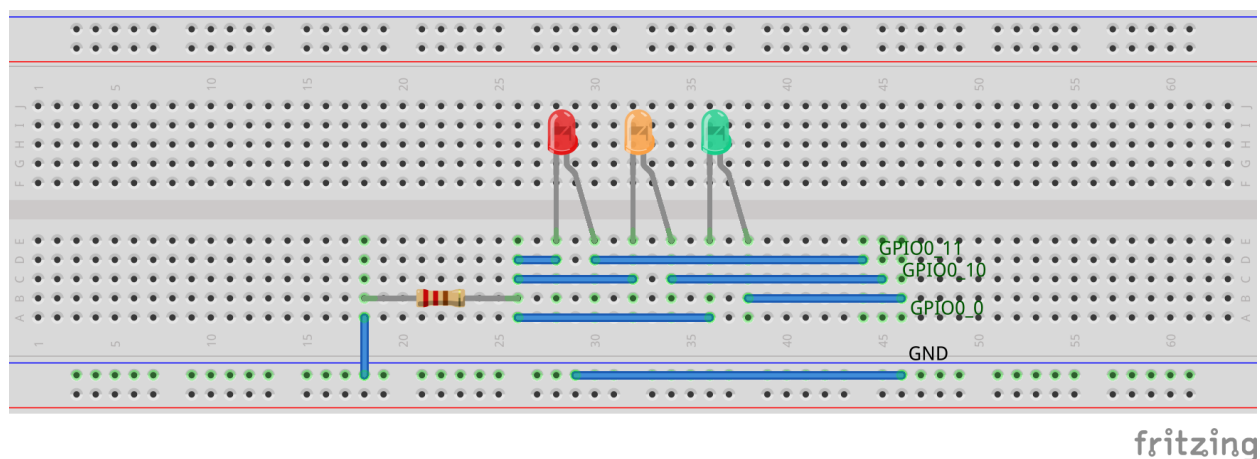


FIGURE 2 – Exemple montage feu

Du point de vue de notre micro-contrôleur, allumer ou éteindre une Led revient à changer l'état de la sortie correspondante. Pendant le TP 1, vous avez utilisé les sorties GPIO connectés en interne aux LEDs du module (LPC\_GPIO\_0\_28 et LPC\_GPIO\_0\_29)

Pour cet exercice, vous allez programmer le comportement d'un feu de circulation. En utilisant des temporisateurs comme la fonction `msleep(t)` en C, le but de cet exercice sera de simuler le changement de l'état d'allumages des LEDs. Pensez bien à garder l'ordre d'allumage, vous pouvez utiliser la structure `switch/case`.

## Exercice 2. LED RGB Neopixel

Pour cet exercice vous allez récupérer le matériel demandé dans la Figure 3.

A partir des exemples présents dans le code du module (dispo sur git), créez le support d'une led RGB de type Neopixel dans votre nouvelle application, ce type de LED permet de connecter plusieurs LEDs en chaîne. La LED doit être connectée de cette façon :

- Pin + à la sortie +3.3V du micro-contrôleur

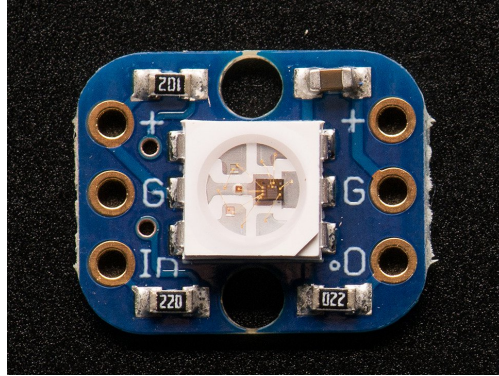


FIGURE 3 – 1 Led RGB

- Pin **G** à la sortie **GND** du micro-contrôleur
- Pin **I** à une sortie **GPIO**, dans le code donné ça sera le **GPIO0\_19**.

Pour connecter plusieurs LEDs en chaîne vous devez connecter le pin **0** de la première LED au pin **I** de la LED suivante et les pins **+** et **G** entre les deux LEDs.

Le code pour gérer ce type de LED est donné dans l'exemple `/apps/base/ledstrip/`, prenez le temps de bien comprendre le code et le tester avec une LED. Modifiez ce code pour itérer entre les couleurs bleu, blanc, rouge à une durée de 250ms. Puis si vous avez le temps, demandez à vos encadrants une deuxième/troisième LED selon la disponibilité de matériel et tester avec plusieurs LEDs.

### Exercice 3. Capteurs météo

Pour cet exercice vous allez récupérer le matériel demandé dans la Figure 4. Cette puce regroupe plusieurs capteurs environnementaux que vous allez devoir interroger pour obtenir leurs valeurs. Ces valeurs seront renvoyés par le port série et lues avec `minicom`.

Pour ce faire, vous disposez du code contrôleur sur le git suivant : <https://github.com/CPELyon/modules-techno-innov>. Attention, ce git est différent du git techno-innov!!!. Le code est dans le sous-dossier `/apps/rf_sub1G/sensors/`.

Le module *sensors* a une interface I<sup>2</sup>C pour pouvoir accéder à ses données, la distribution des pins est présenté dans la figure 4.

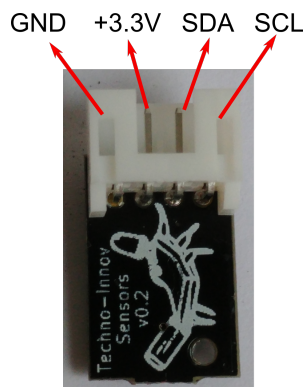


FIGURE 4 – Capteur météo *sensors*

## Exercice 4. Écran mono-couleur

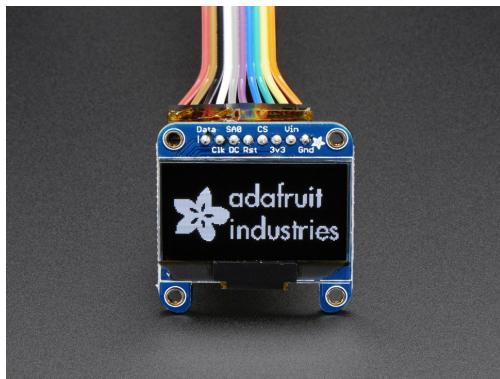


FIGURE 5 – 1 écran mono-couleur

Pour cet exercice vous allez récupérer le matériel demandé dans la Figure 5.

Dans cette exercice le but est de pouvoir afficher sur l'écran la valeur de la température obtenue à partir du capteur intégré au module RF Sub-1GHz ainsi que les noms des intégrants de votre binôme.

Avant d'afficher directement les informations, assurez vous de comprendre comment afficher des informations sur cet écran. Pour ce faire, vous disposez du code contrôleur d'un écran OLED Adafruit sur le git : <https://github.com/CPELyon/modules-techno-innov>. Le code est dans le sous-dossier /apps/rf\_sub1G/oled/ est vous devez l'adapter selon les connecteurs de votre micro-contrôleur.

Dans le code donné, on utilise l'interface I<sup>2</sup>C pour pouvoir envoyer les informations à afficher, on positionne les données en indiquant la ligne à utiliser, la distribution des pins est présenté dans la figure 6.

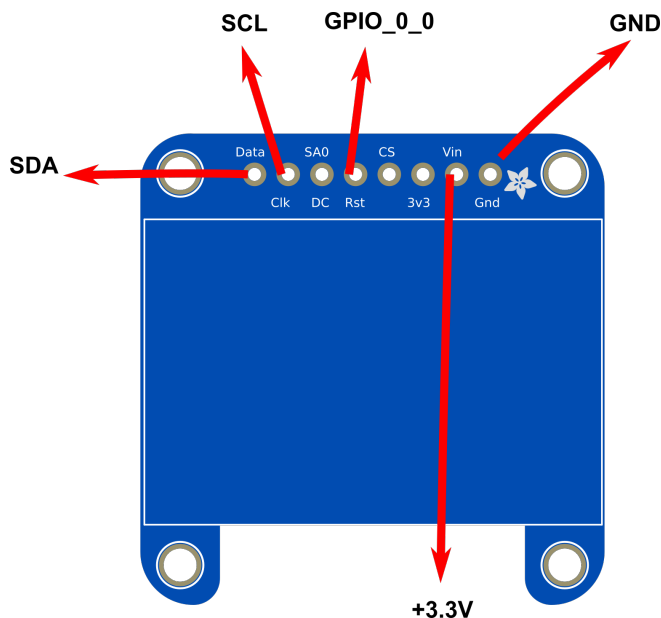


FIGURE 6 – Disposition des pins du module *OLED*

Pour finir, affichez aussi la température obtenue à partir du capteur *sensors* pour la comparer à celle intégrée au micro-contrôleur.

## Liens outils

Lien vers d'autres outils pour programmer les LPC :

- mxli : <http://www.windscooting.com/softy/mxli.html>
- lpc2lisp : <http://sourceforge.net/projects/lpc2lisp/>
- nxpprog : <http://sourceforge.net/projects/nxpprog/>
- GLPC (GUI pour lpc2lisp) : <http://sourceforge.net/projects/glpc/>
- Neopixel : <https://learn.adafruit.com/adafruit-neopixel-uberguide/the-magic-of-neopixels>
- Écran mono-couleur : <https://www.adafruit.com/product/938>