

Utilisation des E/S en C et Makefile

Utilisation de WiringPi

Développez une application en C sur rPi permettant de contrôler les GPIO du rPi. L'application doit s'appeler `rpi-gpio` et doit prendre les paramètres suivants:

```
rpi-gpio bcm_pin on|off
```

Pour cette application, utiliser la librairie WiringPi pour interfacer les E/S. Pour la compilation, utiliser le compilateur gcc. Vous devez donc installer WiringPi ainsi que GCC pour pouvoir faire fonctionner votre programme.

Vous pouvez appeler le fichier source `rpi-gpio-wpi.c`

Utilisation de la librairie standard

Refaire le même programme, mais en utilisant uniquement les fonctions de la librairie standard. Ceci veut dire que vous pouvez accéder aux E/S avec les fonctions `open()`, `read()`, `write()` et `close()`. Le format reste le même.

Vous pouvez appeler le fichier source `rpi-gpio-libc.c`.

Optimisation de votre projet et Makefile

Une fois que vous avez codé les deux possibilités, vous allez pouvoir séparer votre code pour minimiser les doublons. Créez quatre nouveaux fichiers:

- `main.c`
- `gpio.h`
- `gpio-wpi.c`
- `gpio-libc.c`

Dans `gpio.h`, déclarez la fonction `set-gpio()`. La fonction prend deux paramètres: numéro de pin et son état. Ensuite, dans `gpio-wpi.c` et `gpio-libc.c` vous allez implémenter la même fonction, mais en utilisant WiringPi et libc respectivement. Dans `main.c`, vous ne pouvez appeler que la fonction `set-gpio()`.

Makefile

Il est possible maintenant de pouvoir compiler le programme soit en utilisant *WiringPi*, soit en utilisant la librairie standard C. Concevez votre Makefile pour pouvoir fournir le paramètre `wpi` pour compiler avec *WiringPi* ou `libc` pour compiler avec la librairie standard C. Les deux possibilités sont:

```
make wpi
make libc
```