

attendre une demi-seconde avant chaque parcours de la boucle. Ceci permet à d'autres applications de tourner pendant le temps de sommeil, mais signifie qu'il existe un temps moyen d'un quart de seconde avant que tout changement sur l'entrée ne soit pris en compte. Délai plus court, réponse plus rapide, CPU plus chargé... un choix cornélien!

À mesure que le nombre d'entrées augmente, et que le nombre de réponses à ces entrées devient plus important et plus varié, il est souvent nécessaire de gérer les tâches avec des priorités différentes. Les interruptions sont le moyen de connecter rapidement une entrée comme "Il y a un gouffre juste devant la voiture" avec la réponse "Stop!".

## Un autre programme en Python

[http://ryniker.ods.org/raspberrypi/MagPi/interrupt\\_test23.py](http://ryniker.ods.org/raspberrypi/MagPi/interrupt_test23.py) montre comment gérer les interruptions. Ce programme configure la broche 23 en entrée, configure le fichier edge en "both" (les deux), de sorte que les interruptions se produisent sur un front montant ("rising") ou descendant ("falling"), ouvre ensuite le fichier value de la broche. L'appel de `select.poll()` crée un objet de polling (balayage des entrées) "po", ensuite `po.register()` ajoute le fichier value de la broche du GPIO en tant que source qui peut répondre à une requête de `po.poll()`. Ce programme utilise cette seule source d'interruption, mais d'autres broches du GPIO, et de nombreuses autres sources, peuvent être enregistrées dans l'objet de polling. Par exemple, un pipe connecté à un autre processus pourrait être une source d'interruption, ou encore un socket qui reçoit des données sur le réseau depuis un système distant.

Le second paramètre de `po.register` spécifie laquelle des 3 conditions sera reconnue comme interruption. La valeur `select.POLLPRI` spécifie que seule "donnée prioritaire à lire" provoquera une interruption. Les autres conditions possibles - "données disponibles" et "prêt pour une sortie" - sont toujours vraies pour une broche de GPIO, donc une opération de polling lorsque l'une d'elles est activée sera réalisée immédiatement. Si d'autres sources d'interruptions sont enregistrées avec `po`, elles peuvent utiliser ces conditions.

Parfois, l'absence d'un signal attendu peut être importante. L'appel à `po.poll(60000)`

attendra une interruption, mais seulement pendant 60 secondes (60.000 millisecondes), avant de retourner une liste vide de signaux d'interruption, indiquant qu'il a atteint la limite fixée (time out).

Le noyau maintient le fichier value d'une broche du GPIO avec 2 valeurs : un caractère 0 ou un 1 qui représente la valeur actuelle sur la broche, et un caractère LF (Line feed, '\n', 0x0A). `f.seek(0)` remet la position courante au début du fichier, de façon à ce que la valeur du premier caractère puisse être lue à nouveau.

## Extension du GPIO

Seules quelques broches du GPIO sont accessibles sur le Raspberry Pi, mais plusieurs personnes ont montré que des circuits intégrés peu chers comme le MCP23017 peuvent utiliser le bus I2C pour augmenter ce nombre. Un schéma comme <http://shop.ciseco.co.uk/k002-slice-of-pi-o/> peut être utilisé jusqu'à 8 fois pour ajouter 128 broches d'entrées/sorties numériques au Raspberry Pi. Utiliser la configuration drain-ouvert de la sortie interruption du MCP23017 pour relier les interruptions de plusieurs circuits à une seule broche du GPIO. Une résistance de tirage au +3V3 met l'entrée à 1, jusqu'à ce qu'un des circuits connectés la mette à 0. Quand une interruption survient, le gestionnaire d'interruption doit lire les valeurs de tous les dispositifs générant des interruptions pour savoir lequel a activé le signal d'interruption (il peut y en avoir plusieurs), lancer les programmes de traitement appropriés, puis supprimer toutes les demandes d'interruptions (donc l'entrée du GPIO repasse à l'état haut) pour autoriser le déclenchement d'une nouvelle interruption.

## Résumé des URL

Schéma du Raspberry Pi :

<http://www.raspberrypi.org/wp-content/uploads/2012/04/Raspberry-Pi-Schematics-R1.0.pdf>

Les programmes :

<http://ryniker.ods.org/raspberrypi/MagPi/gpio23-max.py>

<http://ryniker.ods.org/raspberrypi/MagPi/23-maxa.c>

Extension des entrées/sorties :

<http://shop.ciseco.co.uk/k002-slice-of-pi-o/>

**Auteur : Richard Ryniker**