

TP1 Capteurs / Actionneurs

- Tom Bevan
- Vinh Faucher
- Florian Latapie
- TP1 Capteurs / Actionneurs
 - 1. Gestion des périphériques
 - * R1 Lister l'ensemble des périphériques
 - * R2 Remplissez la table d'allumage des afficheurs
 - Programmation des interruptions
 - * R3 Rappeler les avantages et inconvénients de chaque méthode
 - * R4 A quoi sert les préfixes suivants, souvent utilisés en embarqué : volatile sur la variable d'état des boutons poussoirs, static sur la routine d'interruption ?

1. Gestion des périphériques

R1 Lister l'ensemble des périphériques

Périphérique	Adresse
hex0	0x8005080
hex1	0x8005070
hex2	0x8005020
hex3	0x8005040
hex4	0x8005050
hex5	0x8005060
jtag_uart	0x80050b0
key	0x8005030
leds	0x80050a0
onchip_memory2_0	0x8002000
sdram	0x4000000
switches	0x8005090
timer	0x8005000

R2 Remplissez la table d'allumage des afficheurs

Chiffre décimal	Configuration binaire	Configuration hexadécimale
0	100 0000	0x40
1	111 1001	0x79
2	010 0100	0x24
3	011 0000	0x30
4	001 1001	0x19
5	001 0010	0x12
6	000 0010	0x02
7	111 1000	0x78
8	000 0000	0x00
9	001 0000	0x10

Chiffre décimal	Configuration binaire	Configuration hexadécimale
A	000 1000	0x08
B	000 0011	0x03
C	100 0110	0x46
D	010 0001	0x21
E	000 0110	0x06
F	000 1110	0x0E

Programmation des interruptions

R3 Rappeler les avantages et inconvénients de chaque méthode

Méthode de lecture de périphérique

- **Polling** : Les avantages du polling sont qu'elle est simple à mettre en place et à gérer, et qu'elle ne nécessite pas de protocole de communication spécifique. Elle permet également de surveiller régulièrement l'état d'un périphérique ou d'une ressource, ce qui peut être utile pour détecter les erreurs ou les anomalies. Les inconvénients du polling sont qu'elle peut entraîner un surcharge de trafic réseau et une utilisation inutile de la bande passante, surtout si elle est utilisée fréquemment et avec des intervalles courts. Elle peut également augmenter la latence et la latence des opérations, car elle nécessite un délai de réponse pour chaque demande de polling. En outre, polling peut ne pas détecter les événements en temps réel et peut manquer des événements importants s'ils se produisent entre les pollings.
- **Interruptions** : Les avantages des interruptions sont qu'elles permettent une détection en temps réel des événements, ce qui réduit la latence et la latence des opérations. Les interruptions permettent également de réduire la charge de traitement sur le processeur en ne nécessitant une attention qu'en cas de besoin. Les interruptions peuvent également réduire la consommation d'énergie en ne nécessitant une surveillance active que lorsque cela est nécessaire. Les inconvénients des interruptions sont qu'elles peuvent rendre le code plus difficile à déboguer et à maintenir, car elles peuvent causer des effets de bord imprévus lorsque plusieurs interruptions sont gérées simultanément. Les interruptions peuvent également causer des conflits d'accès aux ressources partagées, ce qui peut entraîner des erreurs de synchronisation. Enfin, les interruptions peuvent causer des erreurs de matériel et de logiciel si elles ne sont pas gérées correctement.

R4 A quoi sert les préfixes suivants, souvent utilisés en embarqué : volatile sur la variable d'état des boutons poussoirs, static sur le routine d'interruption ?

- **Volatile** : Le mot-clé "volatile" est utilisé pour indiquer au compilateur qu'une variable peut être modifiée par des sources extérieures au programme, telles que des interruptions ou des accès à la mémoire partagée. En utilisant "volatile" sur une variable qui représente l'état des boutons poussoirs permet de s'assurer que le compilateur ne supprimera pas des lectures inutiles de cette variable, car il sait que cette variable peut changer à tout moment. Cela permet également de s'assurer que les modifications apportées à cette variable par des interruptions ou des accès à la mémoire partagée sont prises en compte de manière fiable. |
- **Static** : Le mot-clé "static" est utilisé pour limiter la visibilité d'une routine ou d'une variable à un seul fichier de code source. En utilisant "static" sur une routine d'interruption, cela permet de s'assurer qu'elle ne sera pas accessible depuis d'autres fichiers de code source, ce qui peut aider à maintenir la sécurité et la stabilité du système. Cela peut également aider à éviter les conflits de nom avec des routines d'interruption définies dans d'autres fichiers de code source. En utilisant "static" sur une routine d'interruption, cela permet également de limiter l'espace mémoire utilisé pour stocker les informations relatives à cette routine, cela permet de conserver de la mémoire pour d'autres usages. |

Temps (ms)	Temps Essai 1	Temps Essai 2	Temps Essai 3	Temps Essais 4	Temps Essai 5
Vinh	0.379	0.378	0.383	0.374	0.467
Florian	0.400	0.437	0.415	0.470	0.479
Tom	0.444	0.416	0.381	0.528	0.468