

# MICRO-CONTROLEURS : PROBLÈMES ET SOLUTIONS

Les applications à base de microcontrôleurs amènent leur lot de problèmes qui

trouvent en général assez facilement réponse. Nous avons essayé de regrouper

les problèmes fréquemment rencontrés et de donner les solutions afférentes.

Ce qui est en général bon pour la famille 8051 l'est pour d'autres...

**P.\* J'ai une magnifique alimentation de laboratoire connectée à ma carte microprocesseur. Je mets en route le montage avec l'interrupteur de l'alimentation, et ça ne marche pas !**

S.\* Mettez un interrupteur entre l'alimentation et la carte microprocesseur ou mettez un reset manuel pour démarer le montage. Les alimentations de laboratoire ont souvent des temps de mise en route importants, or les microprocesseurs ont horreur des alimentations qui s'établissent trop lentement et font n'importe quoi dans ces cas là. De plus si vous voulez forcer le reset en éteignant puis en rallumant votre alimentation, il est probable que la tension d'alimentation n'aura pas eu le temps de chuter pour resetter le microprocesseur.

**P. J'ai un émulateur de ROM (ou un émulateur) connecté à ma carte microprocesseur, quand je mets en route l'alimentation du système, le reset ne se fait pas ou se fait mal.**

S. Etes-vous sûr que l'émulateur de ROM n'alimente pas votre montage quand l'alimentation est coupée ? L'émulateur de ROM a sa propre alimentation qui peut alimenter votre montage via les lignes de données de la ROM simulée.

Lorsque vous coupez l'alimentation de votre montage, l'émulateur de ROM l'alimente suffisamment par les lignes de données pour que le microprocesseur continue de fonctionner. Au rétablissement de l'alimentation de votre montage, le reset ne fonctionnera pas. Certains émulateurs de ROM disposent de lignes accessoires qui peuvent être pilotées depuis le calculateur auquel elles sont reliées. Utilisez une de ces lignes pour commander le reset, ou installez un reset manuel sur votre montage.

**P. Je viens de câbler ma carte 80C5X, ça ne marche pas.**

S. Reliez vous EA au bon potentiel ? Si vous avez une ROM externe, EA doit être à la masse (c'est toujours le cas avec les microcontrôleurs romless). Si

vous fonctionnez avec la ROM interne du microprocesseur (microprocesseur à fenêtre ou OTP), EA est relié au +5V. S. L'oscillateur de votre microcontrôleur fonctionne t-il ? Si vous avez des fils de plus de deux centimètres entre le quartz et les connexions du microprocesseur, ou entre le quartz et les capacités de l'oscillateur, ou encore entre les capacités et la masse du microprocesseur, RECABLEZ. Peut-être vous êtes-vous dit : j'ai un quartz de 1 MHz, ça marchera mieux qu'avec un quartz de 6 ou 12 MHz... Faux, remettez tout de suite le bon quartz. Cela n'oscille toujours pas ? Essayez de mettre une résistance de 1 MΩ à 10 MΩ entre les pattes du quartz. Vérifiez la valeur des capacités aux bornes du quartz.

**P. L'alimentation est OK, l'oscillateur fonctionne, il n'y a pas de courts-circuits ni dans les lignes de données, ni dans les lignes d'adresses, le programme ne fonctionne pas.**

S. Essayez un programme ultra simple (deux instructions, une pour incrémenter un port et l'autre pour boucler, par exemple). S'il ne fonctionne pas, vérifiez les points précédents, et essayez de changer les composants ! Si le programme simple fonctionne, vérifiez votre programme. Mettez des instructions qui manipulent des lignes de port pour déterminer à quel endroit le programme cesse de s'exécuter normalement. Voici les erreurs classiques :

**P. Le programme s'exécute bien jusqu'au retour du premier sous-programme.**

S. Le pointeur de pile S.P. n'est pas initialisé. A chaque appel de sous-programme, l'adresse de retour est empilée dans la pile système. Si le pointeur de pile n'est pas initialisé, au retour du premier sous-programme, l'adresse de retour peut avoir une valeur incorrecte qui plante le programme. Il faut initialiser le S.P. avant tout appel de sous-programme et avant l'autorisation des interruptions.

S. Une erreur classique est de faire un

sous-programme qui initialise les registres internes du microprocesseur et la mémoire. Etes-vous sûr de ne pas remettre à zéro l'adresse de retour dans la pile, dans votre souci de grand nettoyage ? Ou d'initialiser le S.P. justement à l'intérieur de ce sous programme ?

**P. Le programme s'exécute bien jusqu'à l'autorisation des interruptions.**

S. Le programme gérant l'interruption se trouve t-il bien à l'endroit correspondant à l'interruption activée ? Se termine t-il bien par RTI ? L'indicateur qui a activé l'interruption est-il bien désactivé dans l'interruption ? Certaines interruptions sont réactivées automatiquement quand elles sont servies, en revanche pour d'autres, il est nécessaire de désactiver manuellement les indicateurs qui les ont générées (RI, TI, etc). Dans le cas contraire, le programme d'interruption sort, et, après avoir exécuté l'instruction courante, entre de nouveau dans le programme d'interruption. Le programme principal s'exécute alors à une vitesse très lente.

S. Vérifiez qu'il reste suffisamment de place dans la pile. Pour vérifier la place dans la pile, initialisez la mémoire avec une valeur connue (0, 0x55, ou ce que vous voulez), puis faites un petit sous-programme qui scrute la pile depuis l'adresse la plus haute (0xFF) jusqu'à rencontrer une valeur différente de la valeur d'initialisation. En appelant ce sous-programme de temps en temps dans votre programme, vous pourrez ainsi suivre l'évolution de l'utilisation de la pile système.

**P. Mon programme fonctionnait normalement, puis, tout à coup, à la suite d'une modification mineure, le programme ne fonctionne plus du tout. Pourquoi ?**

S. Vérifiez si la taille de votre programme a franchi une barrière correspondant à une puissance de 2, par exemple 1024 octets, 2048 ou 4096 octets, etc. Si c'est le cas, peut-être

P.\* = problème, S.\* = solution.



avez vous la ligne d'adresse correspondante court-circuitée à la masse. Cette ligne d'adresse sera la ligne A10 si vous franchissez la taille de 1024 octets, la ligne A11 si c'est 2048 octets, etc.

S. Peut-être créez-vous une nouvelle variable, ou faites-vous un appel de sous-programme supplémentaire qui aura eu pour conséquence de diminuer la place libre pour le stack (pile système). N'ayant plus assez de mémoire pour stocker les adresses de retour, le programme ne fonctionne plus (cf. question précédente).

**P. Je commande un transistor monté en émetteur commun avec un port P1 ou P3. Cela ne fonctionne pas correctement, ou bien, ça fonctionne pendant un certain temps puis ça ne fonctionne plus...**

S. Si vous commandez un transistor avec un port de 80C51, lisez bien la documentation. Quand on envoie un niveau logique haut sur le port, celui-ci n'est capable de délivrer que quelques micro-ampères. Aussi, si votre résistance de base est inférieure à quelques dizaines de kΩ, point de salut. Donc, si vous utilisez le port en sortie, utilisez toujours une résistance de rappel vers le +5V de quelques kΩ (à moins que vous ne vous connectiez sur un dispositif à haute impédance d'entrée).

De plus, lors de la commutation de l'état bas à l'état haut, le port dispose d'un transistor supplémentaire connecté temporairement au +5V pour accélérer les temps de montée des signaux. Cet apport transitoire de courant suffit souvent à faire commuter le transistor, ce qui donne l'apparence d'un fonctionnement normal (pas très long-temps). Un autre problème apparaît si vous lisez l'état du port pour connaître l'état du niveau logique que vous avez précédemment envoyé. Si l'impédance par rapport à la masse de ce qui est connecté au port est trop faible, la valeur lue sur le port sera toujours zéro, alors que la commande fonctionnera.

**P. J'utilise un microcontrôleur avec un watch-dog pour contrôler le bon déroulement du programme. Le programme se « plante » parfois et le watch-dog ne génère pas de reset comme il devrait le faire.**

R. Peut-être utilisez-vous une interruption périodique pour relancer le watch-dog. Si c'est le cas, sachez qu'il est probable que le programme principal soit « planté » alors que le programme d'interruption se déroule normalement. Dans ce cas, le watch-dog, relancé dans le programme d'interruption, ne provoquera pas de reset pour redémarrer le programme dans de bonnes conditions. Une autre mauvaise utilisation du watch-dog est de parsemer le programme d'appels à la fonction de recharge du watch-dog. Si le watch-dog est relancé dans de nombreux endroits du programme, il est probable qu'en cas de crash de celui-ci, il tombe sur une zone du programme où le watch-dog est relancé, et donc que celui-ci soit inefficace.

**P. J'utilise un convertisseur analogique/digital (ou un microcontrô-**

; Gestion d'une interface série, réception bufferisée, émission sans buffer  
; La longueur de la file tampon doit être une puissance de 2 (2, 4, 8, 16...)  
; RD\_RX et WR\_RX sont les pointeurs de lecture et d'écriture dans la file.

| SEGMENT_ | DATA               | SEGMENT DATA                                    |
|----------|--------------------|-------------------------------------------------|
| SEGMENT_ | BIT                | SEGMENT BIT                                     |
| SEGMENT_ | STACK              | SEGMENT IDATA                                   |
| L_FILE   | EQU 4              | ; longueur de la file (2, 4, 8, 16,...)         |
| WR_RX:   | RSEG DS 1          | ; pointeur d'écriture [0..L_FILE-1]             |
| RD_RX:   | DS 1               | ; pointeur de lecture [0..L_FILE-1]             |
| FILE_R:  | DS RSEG L_FILE     | ; file de réception                             |
| TTI:     | DBIT 1             | ; flag de remplacement de TI                    |
|          | RSEG SEGMENT_STACK |                                                 |
|          | DS 1               | ; uniquement pour positionner le stack à la fin |

; définition des vecteurs de reset et d'interruption  
CSEG AT 0 ; vecteur de reset en 0  
AJMP START  
CSEG AT 23H ; vecteur d'émission / réception série  
AJMP RSIT

; PROGRAMME D'INTERRUPTION SERIE  
; Si l'interruption provient de la réception d'un caractère, met celui-ci dans  
; une file circulaire. Si il provient de la fin de transmission d'un caractère,  
; positionne le flag de remplacement TTI

|        |                      |                                                |
|--------|----------------------|------------------------------------------------|
| RSIT:  | JNB TI,RSITO         | ; si pas transmission, => réception            |
|        | CLR TI               | ; remet à 0 le flag transmission               |
|        | SETB TTI             | ; et positionne le flag secondaire             |
|        | JNB RI,RSIT1         | ; si pas réception, sort                       |
| RSITO: | CLR RI               | ; remet à 0 le flag reception                  |
|        | PUSH PSW             | ; sauve les registres                          |
|        | PUSH ACC             |                                                |
|        | MOV A, R0            | ; sauve R0 sans tenir compte de sa banque      |
|        | PUSH ACC             |                                                |
|        | MOV A, WR_RX         | ; pointeur d'écriture                          |
|        | ADD A, #FILE_R       | ; file de reception                            |
|        | MOV R0, A            | ; R0 pointe sur l'emplacement de la donnée     |
|        | MOV A, SBUF          | ; prend la donnée recue                        |
|        | MOV @R0, A           | ; transfère dans la file                       |
|        | INC WR_RX            | ; incrément le pointeur d'écriture             |
|        | ANL WR_RX, #L_FILE-1 | ; si dépasse la longueur de la file, boucle... |
|        | POP ACC              | ; restaure les registres utilisés              |
|        | MOV R0, A            |                                                |
|        | POP ACC              |                                                |
|        | POP PSW              |                                                |
| RSIT1: | RETI                 |                                                |

; Routines de lecture / écriture dans l'interface série

; écriture du caractère A vers l'interface série  
RSPUT: JNB TTI,RSPUT ; attend la fin de la transmission  
CLR TTI ; remet à 0 le flag de remplacement de TI  
MOV SBUF, A ; et envoie la donnée  
RET

; lecture d'un caractère depuis la file, résultat dans A  
RSGET: MOV A, RD\_RX ; pointeur de lecture  
XRL A, WR\_RX ; compare au pointeur d'écriture  
JZ RSGET ; si pas de caractère, attend  
MOV A, RD\_RX ; prend le pointeur de lecture  
ADD A, #FILE\_R ; additionne à la position de la file  
MOV R0, A ; R0 pointe maintenant sur l'octet à lire  
MOV A, @R0 ; lit l'octet dans la file  
INC RD\_RX ; pointe sur la valeur suivante  
ANL RD\_RX, #L\_FILE-1 ; passe de L\_FILE-1 à 0 si dépasse  
RET

; initialise l'interface série (le timer 1 est utilisé pour le gén. de bauds)  
RSINI: MOV SCON, #050H ; 8 bits RxTx autorisés, timer 1 : baud  
ANL TMOD, #00FH ; ne touche pas à la programmation du timer 0  
ORL TMOD, #020H ; timer 1 en autoreload  
MOV TH1, #0FDH ; 0E8H: 1200Bd @ 11.0592 MHz (0xFD pour 9600Bd)  
SETB TR1 ; timer 1 actif (dans TCON)  
MOV RD\_RX, WR\_RX ; vide la file de réception  
SETB ES ; valide l'interruption série (active si EA)  
SETB TTI ; initialise le flag secondaire TI  
RET

; PROGRAMME PRINCIPAL  
; Re-émet simplement les caractères reçus par interruption

|             |                          |                                       |
|-------------|--------------------------|---------------------------------------|
| START:      | MOV SP, #SEGMENT_STACK-1 | ; stack : première place libre        |
| ACALL       | RSINI                    | ; initialise l'interface série        |
| SETB EA     |                          | ; autorise les interruptions validées |
| LOOP:       | ACALL RSGET              |                                       |
| ACALL RSPUT |                          | ; prend le caractère suivant          |
| AJMP LOOP   |                          | ; et le re-émet                       |
| END         |                          |                                       |

■ Listing 1

leur en comportant un) connecté à un multiplexeur pour acquérir plusieurs entrées et les valeurs que j'obtiens n'ont qu'un lointain rapport avec ce à quoi je m'attendais.

S. Etes-vous sûr que vous n'envoyez pas sur une des entrées une tension supérieure ou inférieure à la tension d'alimentation (ou de référence) ? Attention, les entrées A/D ne sont pas aussi accommodantes que les entrées digitales.

Généralement, elles ne supportent que 0,2 V au-dessus et au-dessous de la tension d'alimentation ; pas question donc de les protéger avec une 1N4148 ou équivalent. La plupart des convertisseurs commencent à perdre les pédales avec des surtensions sur les entrées de 0,6 V. Aussi, pour protéger les entrées, utilisez des diodes schottky avec un seuil de 0,3 V (en fait ça marche encore avec un seuil de 0,4 V) ou alors mettez avant le convertisseur un ampli opérationnel CMOS de type rail à rail alimenté entre la masse analogique et la tension de référence du convertisseur (un amplificateur rail à rail est un amplificateur dont la tension de sortie peut atteindre la tension d'alimentation).

Vous serez sûr alors d'une part de ne pas dépasser la tension d'alimentation, et d'autre part d'attaquer le convertisseur avec une impédance suffisamment faible. (cf. question suivante).

P. J'utilise un convertisseur Analogique/digital disposant de plusieurs

entrées, et j'ai l'impression qu'il y a de la diaphonie entre les voies (une voie interfère avec une autre).

S. Vous avez peut-être protégé les entrées de votre convertisseur avec deux diodes connectées à l'alimentation, précédées par une résistance. Cette résistance constitue avec la capacité interne du convertisseur un réseau RC. Lors de l'acquisition d'une voie, cette capacité se charge à la tension d'entrée sélectionnée.

Si on commute sur une autre voie, cette capacité se décharge dans la résistance de protection de l'autre voie jusqu'à atteindre la tension d'entrée que l'on veut mesurer. Si la résistance est trop élevée, cette tension n'est pas atteinte, et les voies interagissent entre elles.

**P. Je dois gérer une interface série en réception par interruption (avec un buffer), en revanche, je préférerais la gérer directement sans buffer pour ce qui concerne l'émission, or il n'y a qu'un seul flag pour autoriser l'interruption, que ce soit en réception ou en émission. Comment faire ?**

S. Si on gère l'interruption réception, il faut gérer aussi l'émission. La manière la plus simple de gérer l'émission dans ce cas est de créer un flag auxiliaire qui sera la recopie de TI (cf. listing 1)

**P. J'ai une carte micro-contrôleur connectée à un téléviseur ou à un Minitel (ou à quelque chose qui est**

rélié au secteur). Celle-ci fonctionne mal.

R. Les masses de certains appareils sont flottantes et sont parfois reliés au secteur par des capacités. Ces capacités vieillissant, leurs fuites augmentent, et il peut arriver (souvent) que des courants importants traversent le montage comportant le microprocesseur si celui-ci est relié par ailleurs à une masse (ou connecté à un appareil lui-même connecté à la masse). Ces courants peuvent provoquer des différences de potentiel qui perturberont le fonctionnement des circuits logiques.

A la mise en route d'un Minitel, il pourra même arriver que l'on détruisse les circuits d'interface. La solution est d'utiliser des opto-coupleurs qui permettront d'isoler complètement une partie du montage.

Si vous testez avec un oscilloscope, n'oubliez pas de couper la masse (secteur) de celui-ci. Si vous ne pouvez pas isoler les appareils, étudiez bien le cheminement des masses de manière à diminuer les courants qui circulent à l'intérieur de la zone sensible qui comprend le microprocesseur et ces périphériques. Les destructions éventuelles surviennent généralement à la mise sous tension de l'appareil douteux (Minitel ou téléviseur). Si possible, laissez ceux-ci toujours sous tension.

J.-L. VERN

## UNE GAMME COMPLETE D'OUTILS DE DEVELOPPEMENT

OUTILS DE GÉNÉRATION DE CODE

- Editeur de liens étendu (1)
- Compilateur C-ANSI (1)
- Simulateur intégral
- Emulateurs haute performance (3)
- Kits enseignement

OUTILS DE MISE AU POINT

- Editeur macro Assembleur
- Noyau temps réel
- Carte de développement

80  
51

DISTRIBUTEUR  
OMNITECH-SERTRONIQUE  
AGENCES  
BORDEAUX : 56 34 46 00  
LE MANS : 43 86 74 74  
LILLE : 20 33 21 97  
LYON : 72 73 11 87  
NANTES : 40 49 90 90  
PARIS : (1) 46 13 07 80  
ROUEN : 35 88 00 38

(1) Optimisé pour le 8051 • Nombreux pragmas et modèles de compilation • Bibliothèques ANSI • Flottants simple et double précision.

(2) Pagination de l'espace code jusqu'à 1 Mo.

(3) Supporte plus de 40 composants • Versions interne et externe • 40 MHz • Transparence totale • 512 ko de Ram d'émulation code.

RAISONNANCE  
ZI Rue des Sources  
38920 CROLLES  
FRANCE  
Tél. : 76 08 18 16  
Fax : 76 08 09 97

**SIMULATION ELECTRONIQUE  
INTERACTIVE**

• Entrée de schémas

• Oscilloscope logiciel

- Noyau de calcul interactif IsSpice4
- Plus de 5000 composants en bibliothèque

- plus: balayage de paramètres analyse de monte Carlo optimisation affichage de tensions en temps réel ...

**EXCEM**

12 Chemin des Hauts de Clairefontaine 78580 Maule

tél: 34 75 13 65 fax: 34 75 13 66

Documentation et disquettes de démonstration sur demande.