

PIC*Génial* – Microcontrôleur PIC 16F84/16F870

Introduction à la programmation avec le PIC*Génial*

Incompatibilité 16F84–16F870

Le port A du 16F870 est initialisé en mode lecture analogique. La simple définition de la direction ne suffit pas à assigner les bits 0,1,2,3,5 en entrée (et en sortie il peut y avoir des effets bizarres).

Ceci concerne le poussoir SW2 à droite, qui est relié à RA4. Le poussoir SW3 à gauche sur RA4 est assigné à un timer et pas à un port analogique, et réagit de la même façon sur le 16F84 et le 16F870.

Le programme PicgT page 8 n'agit donc que sur la LED liée à RB3. Le programme PicGT0 page 12 doit être modifié en écrivant bSwitch = 4, ou en ajoutant les instructions d'initialisations ci-dessous. Le programme Picgp1 doit être complété.

Les 5 instructions à ajouter dans l'initialisation du portA sont les suivantes (les pseudos .Ref 16F870 et .Ins 16F870M.asi importent les fichiers correspondants):

```
Clr      PortA
Bank0to1 ; Set Status:#RP0 dans 16F870M.asi
Move    #16'06,W
Move    W,AdCon1 ; Dans 16F870.ref
Bank1to0 ; Clr Status:#RP0
Move    #DirA,W
Move    W,TrisA
```

Erreurs de circuit dans PicgExe870

Les Leds sur RB et le connecteur ont été inversés. Il faut donc soit retourner la plaque pour avoir les poids faibles à droite, soit appeler la routine de permutation de bits suivante:

```
Routine PermuteW Permute les 8 bits dans W (non testé)
; 23 us, 23 bytes (si routine des 9 instr id, 31 us, 15 bytes)
in: W
out: W
mod: C1,C2
PermuteW:
Move    W,C1
Swap    C1,W      ; 8..5
Move    W,C2
RRC     C2
RLC     C1
RRC     C2
RLC     C1
RRC     C2
RLC     C1
RRC     C2
RLC     C1
Swap    C1,W      ; 8..5
Move    W,C2
RRC     C2
RLC     C1
RRC     C2
RLC     C1
RRC     C2
RLC     C1
Move    C1,W
Ret
```