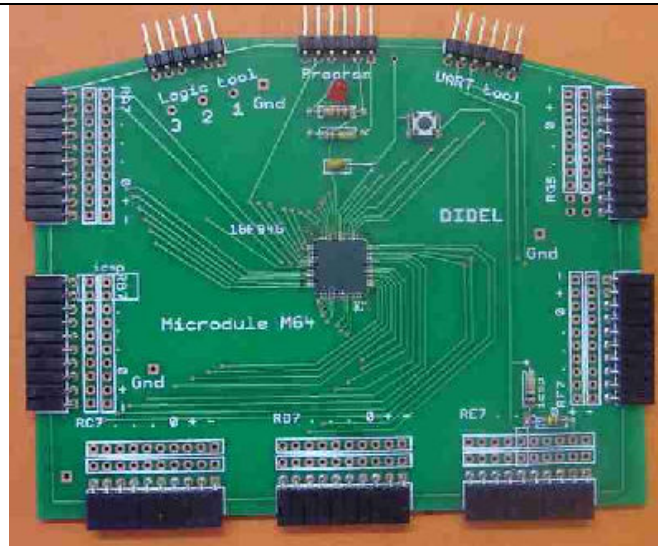


M64 Microchip Pic 16F946

La carte permet de développer des applications pour des petits processeurs en permettant d'afficher des variables, compteurs, pointeurs en temps réel.



The board permits to develop real time applications for small processors. Variables, counters, pointers can be displayed with a minimum of interaction with the process.

Contenu du kit – kit content 25.-

- 1 x M64 PCB
- 1 x 16F946 soldered
- 1 x 100nF capacitor
- 1 x 100k resistors
- 1 x 4k7 resistor
- 1 x 2k2 resistor
- 1 x 3mm LED red
- 1 x 1N34 Ge diode
- 1 x push button
- 3 x ground pin
- 7 x FC07-8
- 1 x MC07-40
- 2 x F05-40

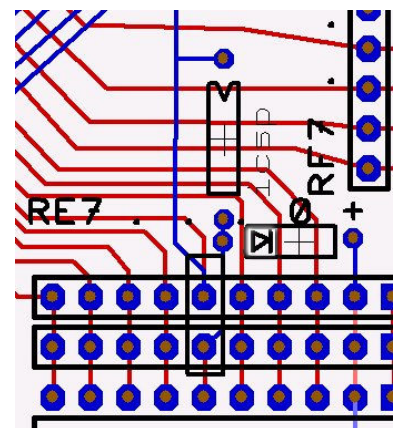
On the proto serie, several wires were missing. Add the 4 still missing wires if you use the logic tool.



Schema

Les ports A B C D E F sont complets (8 bits). Le portG a 6 bits.

La ligne RE3 est la seule à poser problème, car elle est utilisée par la programmation (Mclr) et reçoit du 12V. Pour ne pas gêner le Pickit2, une résistance série de 4.7 kOhm est mise en série entre le processeur et le connecteur de sortie. Pour protéger l'application contre le 12V, une diode limite la tension à 5.5V. La résistance limite le courant si RE3 est une sortie. Un cavalier peut être insérer dans ce cas (à l'emplacement du rectangle), mais il faut l'enlever à chaque programmation.



Initialisation des ports (voir PicT946.asm)

```
    Clr    PortA
    Set    Status:#RP0
    Move   #16'07,W
    Move   W,CmCon0
    Clr    AnSel
;      Move #2'01110000,W; 8 Mhz
;      Move W,OscCon      ; banque 1
    Move   #0,W          ; sorties
    Move   W,PortA
    Move   W,PortB
    Move   W,PortC
    Move   W,PortD
    Move   W,PortE
    Set    Status:#RP1
    Move   W,TrisF      ; banque 3
    Move   W,TrisG
    Clr    PortF
    Clr    PortG
    Clr    Status:#RP0
    Clr    LCDCon      ; banque 2
    Clr    Status:#RP1

    Clr    PortA
    Clr    PortB
    Clr    PortC
    Clr    PortD
    Clr    PortE
```

Avec cette initialisation, tous les bits sont en sortie et la fréquence est de 4 MHz.

Les macros facilitent l'insertion et l'enlèvement des affichages de contrôle. Elle sont englobées dans des ".If Test" pour ne pas avoir besoin de les enlever dans le processeur final.

Pour avoir le même timing dans le processeur final, il faut mettre des Nop ou Jump APC+1 (2 cycles) après le .Else, comme ci-dessous

```
.Macro CopyA
.If Test
    Move 1,W
    Move W,PortA
.Else
    Jump APC+1
.Endif
.Macro CopyF
.If Test
    Set Status:#RP1
    Set Status:#RP0
    Move %1,W
    Move W,PortF
    Clr Status:#RP1
    Clr Status:#RP0
.Else
    Jump APC+1
    Jump APC+1
    Jump APC+1
.Endif
; test macro
Test = 1
    CopyFG Cx1,Cx2
```

Macros

```
Fichier M64Macros.asi dans www.didel.com/pic/T946.zip
\b; CopyA Var ou CopyA #Val
.Macro CopyA
.If Test
    Move 1,W
    Move W,PortA
.Endif
.Endmacro
\b; CopyB Var ou CopyB #Val
.Macro CopyB
.If Test
    Move 1,W
    Move W,PortB
.Endif
.Endmacro
\b; CopyC Var ou CopyC #Val
.Macro CopyC
.If Test
    Move 1,W
    Move W,PortC
.Endif
.Endmacro
\b; CopyD Var ou CopyD #Val
.Macro CopyD
.If Test
    Move 1,W
    Move W,PortD
.Endif
.Endmacro
\b; CopyE Var ou CopyE #Val
.Macro CopyE
.If Test
    Move 1,W
    Move W,PortE
.Endif
.Endmacro
\b; CopyF Var ou CopyF #Val
.Macro CopyF
.If Test
    Set Status:#RP1
    Set Status:#RP0
    Move %1,W
    Move W,PortF
    Clr Status:#RP1
    Clr Status:#RP0
.Endif
.Endmacro
\b; CopyG Var ou CopyG #Val 6 bits
.Macro CopyG
.If Test
    Set Status:#RP1
    Set Status:#RP0
    Move %1,W
    Move W,PortG
    Clr Status:#RP1
    Clr Status:#RP0
.Endif
.Endmacro
\b; CopyFG Var ou CopyFG #Val 6 bits
.Macro CopyFG
.If Test
    Set Status:#RP1
    Set Status:#RP0
    Move %1,W
    Move W,PortF
    Move %2,W
    Move W,PortG
    Clr Status:#RP1
    Clr Status:#RP0
.Endif
.Endmacro
```