

J.L.S. INFORMATIQUE

2, rue Clément ADER

B.P. 50065

57972 YUTZ CEDEX

Tel : 33 (0)3 82 86 00 16

Fax : 33 (0)3 82 86 00 12

URL : www.jls-info.com

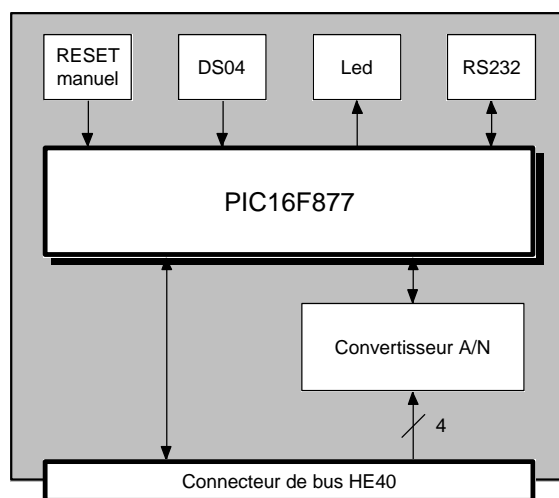
MCU877

Carte universelle à PIC16F877

Caractéristiques générales

La carte **MCU877** est une carte universelle à microcontrôleur 8 bits équipée de 8Ko de FLASH. La carte est au format 100mm x 80mm et dispose d'un convertisseur A/N, d'un connecteur 40 points, d'un bouton-poussoir de RAZ, d'un DIP Switch de configuration et d'un port RS232.

- Microcontrôleur PIC16F877 à 4MHz
- 8Ko de FLASH interne, moniteur embarqué
- 368o de RAM statique interne
- 256o d'EEPROM interne
- UART RS232
- 23 I/O partagées (*Timer, UART, Interrupt...*)
- Dimensions 100mm x 80mm
- Fonctionnement de 0°C à 60°C
- Convertisseur A/N 12 bits sériel, 4 canaux
- DIP Switch de configuration, Led bicolore
- Alimentation +5V programmable, consommation maximum de 21mA

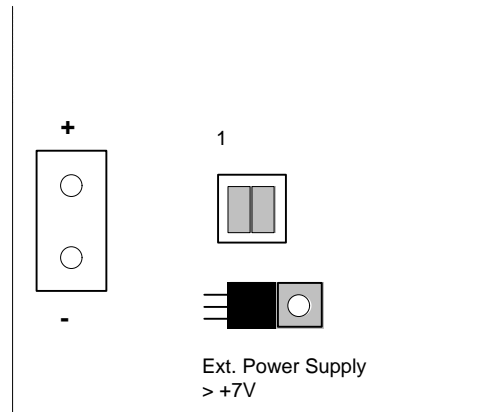
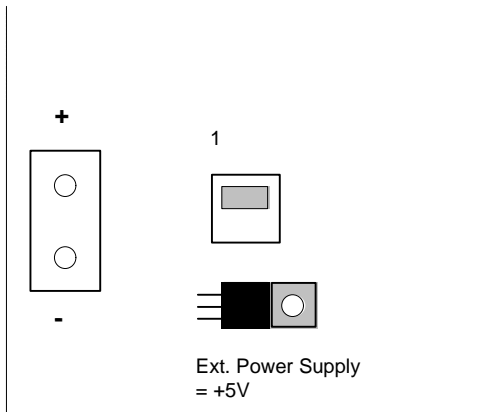


Brochage du connecteur HE40

GND	1	2	GND
AIn2	3	4	AIn3
AIn0	5	6	AIn1
GND	7	8	GND
n.u.	9	10	n.u.
RESET	11	12	RB7 / PGD
RB6 / PGC	13	14	RB5
RB4	15	16	RB3 / PGM
RB2	17	18	RB1
RB0 / INT	19	20	VCC
VCC	21	22	RA4 / T0CKI
RD7 / SP7	23	24	RD6 / PSP6
RD5 / SP5	25	26	RD4 / PSP4
RC0 / T1OSO / T1CKI	27	28	RC7 / RX / DT
RC1 / T1OSI / CCP2	29	30	RC6 / TX / CK
RC2 / CCP1	31	32	RC5 / SDO
RC3 / SCK / SCL	33	34	RC4 / SDI / SDA
RD0 / PSP0	35	36	RD3 / PSP3
RD1 / PSP1	37	38	RD2 / PSP2
VCC	39	40	VCC

Connecteur d'alimentation et configuration

L'alimentation demandée par la carte est de +5Vdc. Néanmoins, pour des applications embarquées ne disposant pas de cette tension, il a été prévu un régulateur à mettre en service :



Codes sources

FLASH

La mémoire de code est située de 0000h à 1FFFh.

Attention, le moniteur est situé dans la plage 1D00h à 1FFFh, ainsi qu'aux adresses 0000h, 0001h et 0002h. *

RAM interne

La lecture et l'écriture de la RAM interne se font par les instructions de transfert suivantes :

```
; equivalences en ram
    cblock 20h           ;ram banque 0
        save1
        save2           ;variables
    endc

...

    movlw 03h
    movwf save1
    movwf save2         ;init save1 et save2 en ram
loop
    movf  save1,w       ;charge
    addlw 45h           ;additionne
    movwf save2         ;stocke
    goto  loop         ;continue

    END
```

Utilisation de la Led bicolore

```
; equivalences
ledv  equ  02h         ;led v rb1
ledr  equ  04h         ;led r rb2

...

    bsf  status,rp0
    bcf  status,rp1    ;banque 1
    movlw b'11111001'
    movwf trisb        ;rb1 & rb2 en sortie
    bcf  status,rp0
    bcf  status,rp1    ;banque 0

...

    movlw ledv
    movwf portb        ;allumage led v
    movlw ledr
    movwf portb        ;allumage led r
    movlw 00h
    movwf portb        ;extinction

...
```

* : voir la rubrique "moniteur de téléchargement embarqué" pour de plus amples informations.

Utilisation du convertisseur A/N et programme de test implanté en FLASH (après passage par le moniteur)

```
LIST P=16F877
INCLUDE <P16F877.INC>
ERRORLEVEL      -302

; variables & equivalences
cblock 20h      ;ram banque 0
    temp      ;mcu
    cpt1      ;compteur 1
    cpt2      ;compteur 2
    cpt_can   ;compteur dclk can
    data_can_i ;data serielle lue can
    data_can_o ;data serielle ecrite can
    datah     ;msb can
    datal     ;lsb can
endc

ledv equ 02h    ;led v rb1
ledr equ 04h    ;led r rb2

p_can equ porte ;pilotage can
din equ 00h     ;re0 : data in
dcs equ 01h     ;re1 : chip select
dclk equ 02h    ;re2 : data clock
dout equ 05h    ;ra5 : data out

; controle can : soc, chx, 12 bits, single ended, no power down
ch0 equ 097h
ch1 equ 0d7h
ch2 equ 0a7h
ch3 equ 0e7h

; code segment
ORG 0003h
goto start

; emet w sur uart
uart_emit
    clrwdt
    btfss pir1,txif
    goto uart_emit
    movwf txreg
    return

; emet w sur uart en hexa
uart_emit_hex
    movwf temp
    swapf temp,w      ;qh
    call writedig
    movf temp,w       ;qb
writedig
    andlw 0fh         ;isole qb
    addlw 0f6h        ;conversion hexa
    btfsc status,c   ;
    addlw 07h         ;lettre
    addlw 3ah         ;chiffre
    goto uart_emit
```

```

; lecture can, channel dans w
convert
    bcf    p_can,dcs    ;cs = 0
    movwf data_can_o    ;controle can
    movlw  08h
    movwf  cpt_can      ;8 pulses dclk can
conv
    rlf    data_can_o,f  ;conversion // -> serie
    btfsc  status,c
    goto   write_1
write_0
    bcf    p_can,din
    goto   pulse1
write_1
    bsf    p_can,din
pulse1
    bsf    p_can,dclk
    bcf    p_can,dclk    ;pulse dclk
    decfsz cpt_can,f
    goto   conv          ;8 bits ecrits ?

    bcf    p_can,din    ;din = 0

    call   lect8bits
    movwf  datah        ;sauve data h can
    call   lect8bits
    movwf  datal        ;sauve data l can
mise_en_forme
    rlf    datal,f
    rlf    datah,f
    andlw  0f0h        ;ajustement data h / l
    movwf  datal
    bsf    p_can,dcs    ;cs = 1
    return

lect8bits
    movlw  08h
    movwf  cpt_can      ;8 pulses dclk
lect
    bsf    p_can,dclk    ;front montant dclk
    btfsc  porta,dout
    goto   lect_h
lect_l
    bcf    status,c
    goto   pulse2
lect_h
    bsf    status,c
pulse2
    bcf    p_can,dclk    ;front descendant dclk
    rlf    data_can_i,f  ;conversion serie -> //
    decfsz cpt_can,f
    goto   lect          ;8 bits lus ?
    movf  data_can_i,w    ;sauve data dans w
    return

```

```

; programme principal
start
    clrwdt
    bsf    status,rp0
    bcf    status,rp1    ;banque 1
    clrf   adcon0
    movlw  06h
    movwf  adcon1        ;can interne off, port a std
    movlw  b'11111111'
    movwf  trisa         ;port a en entree
    movlw  b'00000000'
    movwf  trise        ;re0, re1, re2 en sortie
    movlw  b'11111001'
    movwf  trisb        ;rb1, rb2 en sortie
    bsf    txsta,brgh    ;high speed
    bsf    txsta,txen
    movlw  19h
    movwf  spbrg        ;9600 Bauds
    bcf    status,rp0
    bcf    status,rp1    ;banque 0
    bsf    rcsta,cren
    bsf    rcsta,spen    ;serial port enabled
    bsf    portb,ledv    ;led v on
    bsf    p_can,dcs     ;cs = 1
    bcf    p_can,dclk    ;dclk = 0
    bcf    p_can,din     ;din = 0
loop
    clrwdt
    movlw  ch0
    call   mes_can        ;acq + envoi uart ch0
    movlw  ch1
    call   mes_can        ;acq + envoi uart ch1
    movlw  ch2
    call   mes_can        ;acq + envoi uart ch2
    movlw  ch3
    call   mes_can        ;acq + envoi uart ch3
    goto   loop
mes_can
    call   convert        ;conversion can
    movf   datah,w
    call   uart_emit_hex  ;envoi data h can
    movf   datal,w
    call   uart_emit_hex  ;envoi data l can
    movlw  ' '
    call   uart_emit      ;separateur
    return
END

```

Divers.

La fréquence maximale admissible par le PIC16F877-04P est de **4MHz**. Une fréquence de quartz telle que 3.6864MHz peut être utile pour générer les fréquences de communications sérielles sans erreur. Néanmoins, cela fonctionne parfaitement à 4MHz pour le format pré – programmé 9600, n, 8, 1.

Configuration par DIP Switch

La carte dispose d'un DIP Switch à quatre interrupteurs qui permet de configurer la carte en fonction de ses besoins et du programme qui gère ces switches. Ainsi, on dispose de seize valeurs différentes (0 à F) qui permettent par exemple de choisir une fréquence d'acquisition, un taux de transfert, un nombre d'entrées / sorties, etc. Le DIP Switch est accessible à travers le port A (RA0 à RA3), les interrupteurs tirent les lignes à la masse et celle-ci sont tirées au niveau haut par un réseau SIL.

Correspondance DS04 / Port A :

I1	I2	I3	I4
RA0	RA1	RA2	RA3

Exemple d'utilisation :

I1	I2	I3	I4	RA0	RA1	RA2	RA3	Val*
Off	Off	Off	Off	1	1	1	1	F
On	Off	Off	Off	0	1	1	1	E
Off	On	Off	Off	1	0	1	1	D
On	On	Off	Off	0	0	1	1	C
Off	Off	On	Off	1	1	0	1	B
On	Off	On	Off	0	1	0	1	A
Off	On	On	Off	1	0	0	1	9
On	On	On	Off	0	0	0	1	8
Off	Off	Off	On	1	1	1	0	7
On	Off	Off	On	0	1	1	0	6
Off	On	Off	On	1	0	1	0	5
On	On	Off	On	0	0	1	0	4
Off	Off	On	On	1	1	0	0	3
On	Off	On	On	0	1	0	0	2
Off	On	On	On	1	0	0	0	1
On	On	On	On	0	0	0	0	0

* : valeur lue sur le port A.

```

;equivalences
  cblock 20h
    temp
  endc

...

  bsf    status,rp0
  bcf    status,rp1    ;banque 1
  movlw  06h
  movwf  adcon1        ;entrees porta type logique et arret adc
  movlw  0ffh
  movwf  trisa         ;porta en entree
  bcf    status,rp0
  bcf    status,rp1    ;banque 0

...

  movf   porta,w       ;lecture porta
  movwf  temp          ;sauvegarde en ram
  comf   temp,f        ;inverse les bits a cause du cablage
  movlw  0fh           ;isole quartet bas (d0 ... d3)
  andwf  temp,f        ;resultat dans temp

...

```

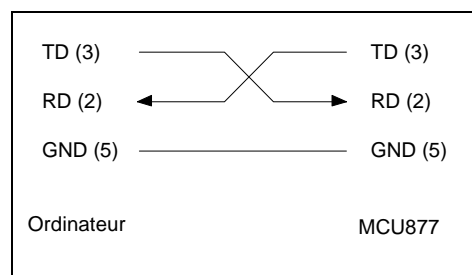
Moniteur de téléchargement embarqué

Le moniteur MON877 a été développé pour télécharger rapidement et simplement un programme *in-situ*, sans être obligé de placer le composant sur un programmeur. Ses fonctions sont donc restreintes :

- lancement automatique du code si l'utilisateur ne prend pas la main au démarrage
- chargement du code utilisateur, avec vérification d'adresses et de validité de données
- lancement manuel du code utilisateur

La liaison entre la carte MCU877 et un ordinateur de développement s'effectue par un port série de type RS232. Les paramètres de communication sont : 9600 bauds, pas de parité, 8 bits de données, 1 bit de stop. Le terminal utilisé pour dialoguer avec le moniteur doit être paramétré avec un délai de fin de ligne de 100ms, afin que le moniteur dispose du temps nécessaire pour programmer la FLASH. Le fichier utilisateur à transférer est le fichier compilé, avec l'extension .HEX.

Câble de liaison ordinateur / MCU877 (prises RS232 mâles 9 points) :



Le code utilisateur doit commencer à l'adresse 0003h, à laquelle branche le moniteur de manière automatique.*

La configuration du microcontrôleur a été programmée de la manière suivante :

- BODEN_ON
- CP_OFF
- WRT_ENABLE_ON
- PWRTE_ON
- WDT_ON
- HS_OSC
- DEBUG_OFF
- CPD_OFF
- LVP_OFF

Le Watchdog est validé, avec un prescaler 1:128. Il convient d'en tenir compte lors de la programmation, grâce à l'instruction CLRWDWT. On trouve donc ses caractéristiques d'activation dans le tableau suivant :

	Min.	Typ.	Max.	Unité
Twdt sans prescaler	7	18	33	ms
Twdt avec prescaler	0.896	2.304	4.224	s

Vcc = +5V, -40°C à +85°C

* : exemple complet dans la rubrique "Utilisation du convertisseur A/N et programme de test implanté en FLASH"

Invite du moniteur et commandes :

```
MON877 : Bootloader V1.0
© JLS Informatique J.S. 2000

PIC>#####          <
Utilisation :
U : chargement
Q : demarrage
PIC>
```

Au démarrage, après affichage des informations commerciales, une barre de progression défile pendant 7s. Pendant cette progression, l'utilisateur prend la main par appui sur une touche, sinon, lorsque le délai est écoulé, le moniteur branche en 0003h et exécute le code correspondant.

Après avoir pris la main sur le moniteur, il affiche les deux commandes disponibles 'U' et 'Q', avec leur description.

```
PIC>u
Attente fichier HEX...
```

On enverra le fichier .HEX correspondant au fichier utilisateur par le terminal (délai de fin de ligne de 100ms)

```
PIC>u
Attente fichier HEX...
#####
0018 mot(s) écrit(s)
PIC>
```

Le PIC se charge de la réception, du contrôle des données et du contrôle des adresses pour programmer la FLASH interne. Pendant cette opération, une barre de progression avance, et à la fin du fichier il indique le nombre de mots écrits.

```
PIC>q
Demarrage programme
```

Le programme utilisateur sera lancé par la commande 'Q'.

Divers.

Les commandes sont traitées aussi bien en majuscule qu'en minuscule.

Un Reset manuel provoquera le lancement du moniteur pour arrêter le programme utilisateur et reprogrammer le PIC.

Le port RS232 reste libre pour l'utilisateur, lorsque le PIC branche sur le code utilisateur.

J.L.S. INFORMATIQUE

2, rue Clément Ader

B.P. 50065

57972 Yutz, Cedex

France

Tél.: 03 82 86 00 16

Fax : 03 82 86 00 12

www.jls-info.com