

SUJET TPs. PIC.

TP 1 et TP 2 Utilisation du logiciel mikroC et du programmeur PicFlash.

TP 3 et TP 4 Caractérisation d'un panneau photovoltaïque, mesure d'irradiation.

Objectifs

- Acquérir les bases pour utiliser un logiciel permettant de programmer un microcontrôleur (PIC) avec un langage de haut niveau (langage C).

Pré-requis :

- base de langage C.

Bibliographie :

- sur le langage C
<http://www-ipst.u-strasbg.fr/pat>
http://lesouriciergris.free.fr/cours_c_dos.html
- sur la programmation des PIC
les programmes exemples fournis dans l'aide du logiciel mikroC
<http://www.abcelectronique.com/bigonoff/> (la bible!, disponible en version papier dans la salle G46.)

Pour tous les TPs, un compte rendu est exigé en fin de séance. On doit trouver dans ce compte rendu :

- Le schéma des connexions réalisées (schéma *propre* réutilisable pour les autres séances).
- Les fonctions utilisées avec leur syntaxe.
- L'organigramme des programmes.
- Le nom des fichiers utilisés et leur emplacement

TP1

Prise en main du système (logiciel, matériel, ...).

En suivant ce qui est fait au vidéo projecteur :

Présentation succincte des E/S du PIC

E/S, quartz, CAN, timer, ...

Prise en main du logiciel mikroC (prétexte : Faire clignoter une LED)
(algorithme, en debug, puis en réel)

- le début : [1st_project_pic_c.pdf](#)
- récupérer et adapter le fichier pour faire clignoter une LED (File Open Program Files / Mikroelektronika / mikroC PRO for PIC/ Examples / Development Systems / EASYPIC6 / Led_blinking / Led_Blinking.c)

```
void main() {  
    PORTB = 0;  
    TRISB = 0;           // Configure pins of PORTB as output  
    while(1) {           // Beginning of a repeat loop  
        PORTB = ~PORTB;  // Toggle diodes on portb  
        Delay_ms(1000);  // Wait for 1 second  
    }  
}
```

Rq : En mode debug, enlever la fonction delay()

- comment trouver les fonctions (Help(F1), mikroC PRO for Pic Libraries, Hardware Libraries ... et aussi Miscellaneous Libraries pour les conversions de type)
- un peu de langage C ([c_syntax.pdf](#) et fin de ces pages)
- communication entre le PC et le PIC (Débuter avec MikroC et le Programmeur PicFlash.pdf)
- utilisation du débogueur en ligne ([compilers_ide.pdf](#))

En autonomie :

Tout fermer, faire un nouveau projet. Ce projet gère un bouton poussoir qui déclenche/arrête un clignotement (sortie numérique : 25 mA max).

Un algorithme et un schéma **propre** des connections (quartz, capa, LED, PIC et résistance) sont exigés avant câblage.

Rendre le système autonome : on alimente le montage via une pile de 9V et un régulateur de tension 7805, on enlève la connexion avec le PC)

TP2

Utilisation d'un afficheur, du convertisseur analogique numérique.

En suivant ce qui est fait au vidéo projecteur :

Présentation du protocole de communication pour utiliser un afficheur

(DOC_AFFICHEUR_LCD.pdf ou DOC_afficheur LCD-162B_argente.pdf en fonction de votre afficheur)

Utilisation des fonctions toutes faites : cf Help(F1) mikroC PRO for Pic Libraries, Hardware Libraries, Lcd Library.

En autonomie :

Pour utiliser l'afficheur, on souhaite utiliser le port B avec les affectations

suivantes : 0,1,5,4,3,2 pour : RS, EN, D7, D6, D5, D4. Comment écrire les

fonctions, comment câbler correctement (code couleur à respecter pour une visualisation facile, noir : masse, rouge : alim, violet :LCD...).

Faire un programme permettant d'afficher votre nom et de faire clignoter une LED.

En suivant ce qui est fait au vidéo projecteur :

Présentation du CAN (broches, caractéristiques, registre de configuration)

Utilisation des fonctions toutes faites (où les trouvez?, lesquelles sont utiles?).

En autonomie :

On utilise un potentiomètre pour créer une tension variable entre 0 et 5 volts. (la visualiser en permanence sur l'oscillo ou au voltmètre). Cette tension est envoyée sur AN0 du PIC. Faire afficher la valeur convertie sur la première ligne de l'afficheur précédée de la chaîne "NUM = "et en parallèle faire clignoter une LED.

Remettre cette valeur en volts pour l'affichage sur la deuxième ligne disponible (précédée de la chaîne ",TENSION = " (utilisation d'une fonction de conversion nombre réelle → chaîne de caractères, attention au format des données pour les calculs.)

_____ TPs 3 et 4 _____

Mesure de courant et de tension. Application à la caractérisation d'un panneau photovoltaïque ou à une mesure d'irradiation.

En autonomie :

Caractérisation d'un panneau photovoltaïque. Adaptez les résultats du TD avec les paramètres de votre panneau et du shunt qui vous a été donné :

- 1/ Mesurez le courant de court-circuit à l'ampèremètre sous ensoleillement maximum.
- 2/ déterminez le gain de l'amplificateur d'instrumentation (AI).
- 3/ En déduire la plage de variation de la tension en sortie de l'AI.
- 4/ Mesurez la tension du circuit ouvert au voltmètre du panneau PV.
- 5/ Déterminez les résistances du pont diviseur.

Mesure d'irradiation.

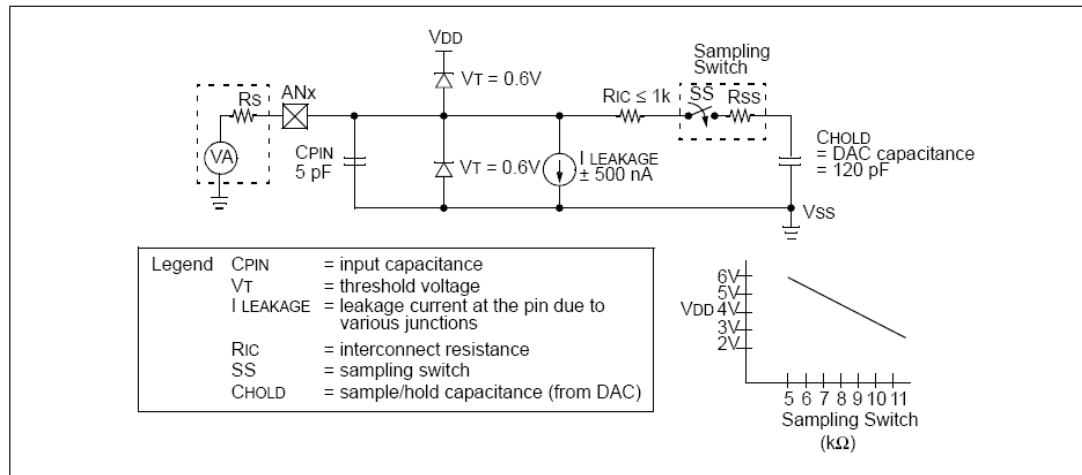
- 1/ Relevez le gain du pyranomètre.
- 2/ déterminez le gain de l'amplificateur d'instrumentation (AI).
- 3/ En déduire la plage de variation de la tension en sortie de l'AI.

... puis, pour tous :

Ecrire l'algorithme. Câbler le montage. Faire afficher les mesures de i_{PV} et v_{PV} (ou celle de la masse, ou celle de l'irradiation) et faire clignoter une LED en même temps. Visualiser à l'oscilloscope le signal envoyé sur la LED, en déduire la cadence d'échantillonnage du programme.

Quand les mesures sont bruitées, il est usuel de faire un moyennage pour améliorer le rapport S/B. Réaliser un programme faisant le moyennage sur 10 ou 20 mesures successives, cela améliore-il le résultat? Que devient la vitesse de rafraichissement de l'affichage?

FIGURE 11-2: ANALOG INPUT MODEL



Modèle analogique d'une entrée du CAN

Caractérisation d'un panneau photovoltaïque.

On connecte le panneau à un rhéostat dont on fait varier la résistance. On part du rhéostat en court-circuit, ce qui donne un premier point. La valeur du rhéostat augmente peu à peu. Il faut d'abord espacer les points relevés en surveillant la tension -le courant ne bouge pratiquement pas- puis dans le coude de la caractéristique on prendra 5 points, enfin dans la dernière partie de la caractéristique, c'est le courant qui bouge alors que la tension varie peu, on veillera donc à espacer les points en surveillant le courant, on finit par la tension de circuit ouvert.

Relever la caractéristique du panneau photovoltaïque et la faire afficher sur un PC. Pour cela 15 points de mesures régulièrement répartis permettent de faire un relevé correct.

Mesure d'irradiation.

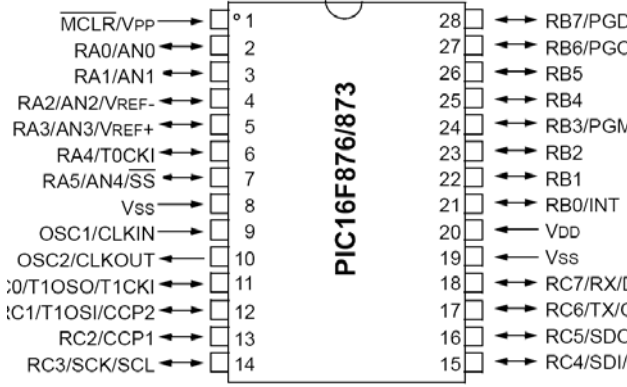
- 1/ Mesurer l'irradiation pour une lampe 100W puis à la même distance pour une lampe 60W.
- 2/ Pour une lampe 100W, Eloignez progressivement le pyranomètre. Relevez et tracez G(distance).
- 3/ Si votre manip fonctionne en autonome, mesurez l'irradiation extérieure.
- 4/ En utilisant les résultats donnés par la caractérisation des panneaux photovoltaïques, déterminez le rendement PV de ces panneaux.

... puis, pour tous : liaison RS232 :

On souhaite envoyer des valeurs du PIC vers le PC, pour cela on va utiliser la liaison série Les fonctions utiles sont les suivantes : cf la librairie **USART Library**.

Au niveau du PC, on peut utiliser LABVIEW, ou plus facilement : l'hyperterminal .
(Programmes Accessoires Communications Hyperterminal)

Faire un programme qui stocke une valeur numérisée chaque fois que l'on appuie sur un bouton et qui envoie ce qu'il a mémorisé (10 valeurs max) sur le port série quand on appuie sur un autre bouton poussoir.

 PIC16F876/873 <i>Brochage du PIC 16F876 (celui du 16F872 est similaire)</i>	• In-Circuit Serial Programming™ (ICSP) via two pins • Single 5V In-Circuit Serial Programming capability • In-Circuit Debugging via two pins • Processor read/write access to program memory • Wide operating voltage range: 2.0V to 5.5V • High Sink/Source Current: 25 mA <i>Ressources du composant.</i>																												
<table border="1"> <thead> <tr> <th>Key Features PICmicro™ Mid-Range Reference Manual (DS33023)</th><th>PIC16F873</th></tr> </thead> <tbody> <tr> <td>Operating Frequency</td><td>DC - 20 MHz</td></tr> <tr> <td>RESETS (and Delays)</td><td>POR, BOR (PWRT, OST)</td></tr> <tr> <td>FLASH Program Memory (14-bit words)</td><td>4K</td></tr> <tr> <td>Data Memory (bytes)</td><td>192</td></tr> <tr> <td>EEPROM Data Memory</td><td>128</td></tr> <tr> <td>Interrupts</td><td>13</td></tr> <tr> <td>I/O Ports</td><td>Ports A,B,C</td></tr> <tr> <td>Timers</td><td>3</td></tr> <tr> <td>Capture/Compare/PWM Modules</td><td>2</td></tr> <tr> <td>Serial Communications</td><td>MSSP, USART</td></tr> <tr> <td>Parallel Communications</td><td>—</td></tr> <tr> <td>10-bit Analog-to-Digital Module</td><td>5 input channels</td></tr> <tr> <td>Instruction Set</td><td>35 instructions</td></tr> </tbody> </table>	Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	Operating Frequency	DC - 20 MHz	RESETS (and Delays)	POR, BOR (PWRT, OST)	FLASH Program Memory (14-bit words)	4K	Data Memory (bytes)	192	EEPROM Data Memory	128	Interrupts	13	I/O Ports	Ports A,B,C	Timers	3	Capture/Compare/PWM Modules	2	Serial Communications	MSSP, USART	Parallel Communications	—	10-bit Analog-to-Digital Module	5 input channels	Instruction Set	35 instructions	Peripheral Features: • Timer0: 8-bit timer/counter with 8-bit prescaler • Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock • Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler • Two Capture, Compare, PWM modules - Capture is 16-bit, max. resolution is 12.5 ns - Compare is 16-bit, max. resolution is 200 ns - PWM max. resolution is 10-bit • 10-bit multi-channel Analog-to-Digital converter • Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Master/Slave) • Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873																												
Operating Frequency	DC - 20 MHz																												
RESETS (and Delays)	POR, BOR (PWRT, OST)																												
FLASH Program Memory (14-bit words)	4K																												
Data Memory (bytes)	192																												
EEPROM Data Memory	128																												
Interrupts	13																												
I/O Ports	Ports A,B,C																												
Timers	3																												
Capture/Compare/PWM Modules	2																												
Serial Communications	MSSP, USART																												
Parallel Communications	—																												
10-bit Analog-to-Digital Module	5 input channels																												
Instruction Set	35 instructions																												

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

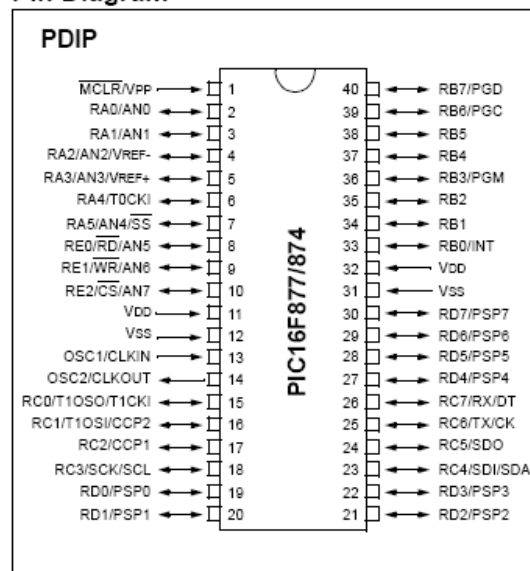
Devices Included in this Data Sheet:

- PIC16F873 • PIC16F876
- PIC16F874 • PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

PIC16F87X

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

PIC16F87X

TABLE 1-1: PIC16F873 AND PIC16F876 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS ⁽³⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	1	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
RA0/AN0	2	2	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0. RA1 can also be analog input1. RA2 can also be analog input2 or negative analog reference voltage. RA3 can also be analog input3 or positive analog reference voltage. RA4 can also be the clock input to the Timer0 module. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
RA1/AN1	3	3	I/O	TTL	
RA2/AN2/VREF-	4	4	I/O	TTL	
RA3/AN3/VREF+	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST	
RA5/SS/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST ⁽¹⁾	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin. RB3 can also be the low voltage programming input. Interrupt-on-change pin. Interrupt-on-change pin. Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock. Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6/PGC	27	27	I/O	TTL/ST ⁽²⁾	
RB7/PGD	28	28	I/O	TTL/ST ⁽²⁾	
RC0/T1OSO/T1CKI	11	11	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or Timer1 clock input. RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I²C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I²C mode). RC5 can also be the SPI Data Out (SPI mode). RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RC1/T1OSI/CCP2	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3/SCK/SCL	14	14	I/O	ST	
RC4/SDI/SDA	15	15	I/O	ST	
RC5/SDO	16	16	I/O	ST	
RC6/TX/CK	17	17	I/O	ST	
RC7/RX/DT	18	18	I/O	ST	
Vss	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
Note 3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

QUELQUES "TRUCS" UTILES

LIRE, MODIFIER des bits individuellement

```
// If RB0 is set, set RC0:  
if (PORTB.F0) PORTC.F0 = 1;  
// Clear TMR0F:  
INTCON.TMR0F = 0;
```

Bitwise Operators Overview

Operator	Operation	Precedence
&	bitwise AND; compares pairs of bits and returns 1 if both bits are 1, otherwise returns 0	8
	bitwise (inclusive) OR; compares pairs of bits and returns 1 if either or both bits are 1, otherwise returns 0	6
^	bitwise exclusive OR (XOR); compares pairs of bits and returns 1 if the bits are complementary, otherwise returns 0	7
~	bitwise complement (unary); inverts each bit	14
<<	bitwise shift left; moves the bits to the left, discards the far left bit and assigns 0 to the right most bit.	11
>>	bitwise shift right; moves the bits to the right, discards the far right bit and if unsigned assigns 0 to the left most bit, otherwise sign extends	11

Relational Operators Overview

Operator	Operation	Precedence
==	equal	9
!=	not equal	9
>	greater than	10
<	less than	10
>=	greater than or equal	10
<=	less than or equal	10

Logical Operators Overview

Operator	Operation	Precedence
&&	logical AND	5
	logical OR	4
!	logical negation	14