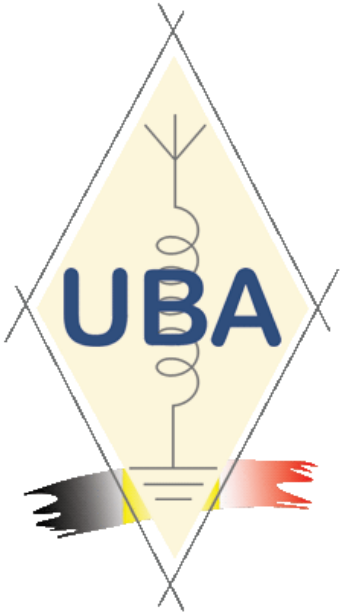
Electronique numérique

Introduction aux composants programmables : les microcontrôleurs



ON7WR – 21 Octobre 2011 – ON4AI

Préliminaires

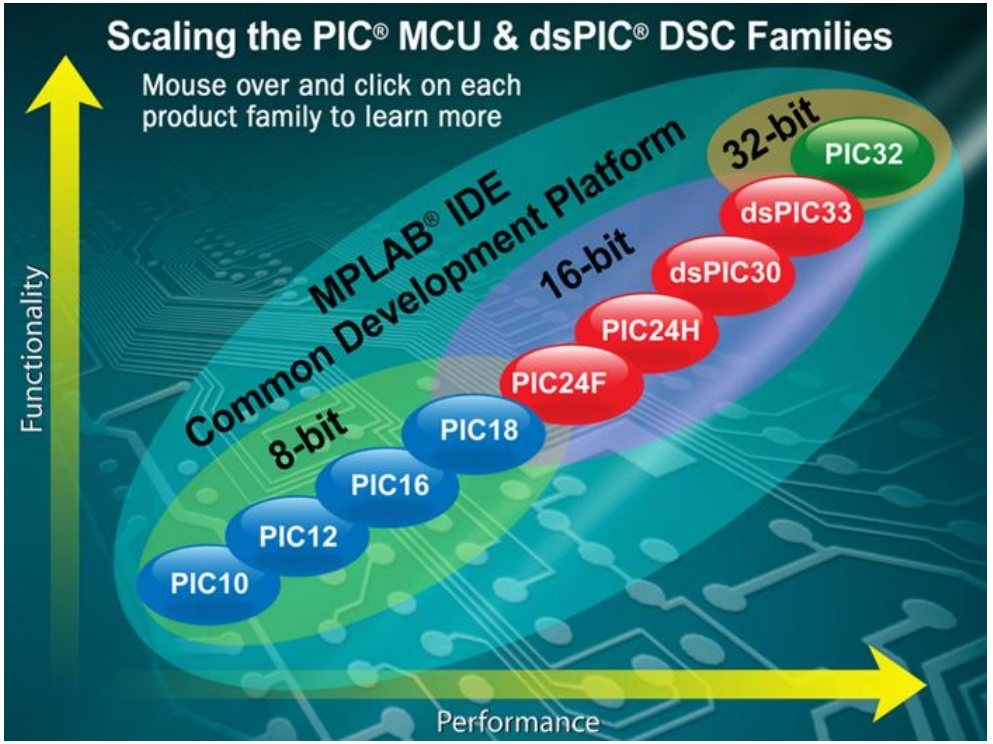
- Information >< expertise → Document plus que résumé
- Objectifs :
  - Introduction ;
  - Donner envie d'aller plus loin ;
  - Sources d'information ;
  - Trucs/Astuces pour démarrer ;
  - Programmation en C >< ASM (!).

Définitions

- Distinction entre µP et µC
  - µP : Processeur → Bus d'adresse + Bus de données
  - µC : Processeur + RAM + ROM + I/O
- → Microcontrôleurs PIC (Microchip)

Microcontrôleurs PIC

- 8 bits : « Standards », 16F et 18F
- 16 bits : 24F et dsPIC (Filtres numérique, DSP)
- 32 bits : Pic 32



PIC 18F…

- Intérêts :
  - Intégration : ADC/DAC, USB, I/O, …
  - Oscillateur interne
  - Bus : SPI, I^2C, CAN, Ethernet

PIC 18F … Datasheet, le minimum

- Brochage : ICSP (In-Circuit Serial Programming)
  - 1 VPP
  - 2 VDD : +3.3/5 VDC
  - 3 GND : VSS
  - 4 PGD
  - 5 PGC
  - 6 PGM (autre méthode)

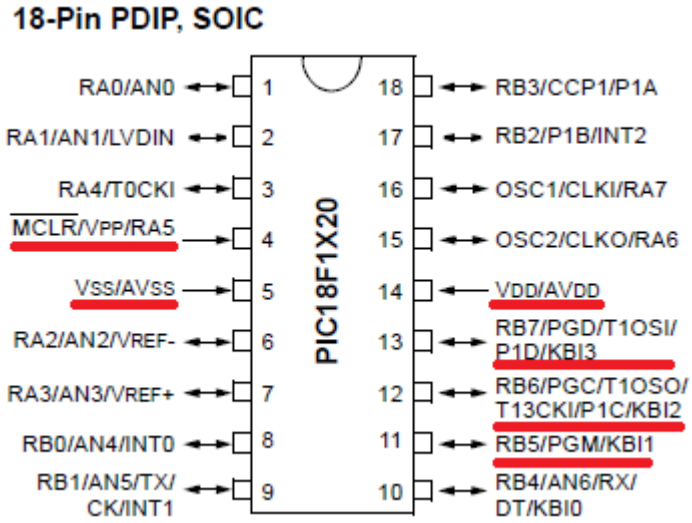
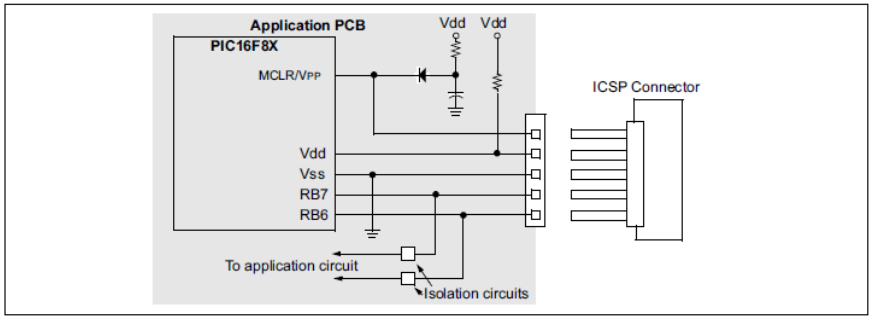


FIGURE 1: TYPICAL APPLICATION CIRCUIT



PIC 18F … Datasheet, le minimum

- Brochage : I/O
  - Port: ½ octet / 1 octet
  - Bits utilisables indépendamment
  - En entrée ou sortie

TABLE 10-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit 0	ST	Input/output port pin or analog input.
RA1/AN1/LVDIN	bit 1	ST	Input/output port pin, analog input or Low-Voltage Detect input.
RA2/AN2/VREF-	bit 2	ST	Input/output port pin, analog input or VREF-.
RA3/AN3/VREF+	bit 3	ST	Input/output port pin, analog input or VREF+.
RA4/T0CKI	bit 4	ST	Input/output port pin or external clock input for Timer0. Output is open-drain type.
$\overline{MCLR}$/VPP/RA5	bit 5	ST	Master Clear input or programming voltage input (if $\overline{MCLR}$ is enabled); input only port pin or programming voltage input (if $\overline{MCLR}$ is disabled).
OSC2/CLKO/RA6	bit 6	ST	OSC2, clock output or I/O pin.
OSC1/CLKI/RA7	bit 7	ST	OSC1, clock input or I/O pin.

TABLE 10-3: PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/AN4/INT0	bit 0	TTL(1)/ST(2)	Input/output port pin, analog input or external interrupt input 0.
RB1/AN5/TX/CK/INT1	bit 1	TTL(1)/ST(2)	Input/output port pin, analog input, Enhanced USART Asynchronous Transmit, Addressable USART Synchronous Clock or external interrupt input 1.
RB2/P1B/INT2	bit 2	TTL(1)/ST(2)	Input/output port pin or external interrupt input 2. Internal software programmable weak pull-up.
RB3/CCP1/P1A	bit 3	TTL(1)/ST(3)	Input/output port pin or Capture1 input/Compare1 output/ PWM output. Internal software programmable weak pull-up.
RB4/AN6/RX/DT/KBI0	bit 4	TTL(1)/ST(4)	Input/output port pin (with interrupt-on-change), analog input, Enhanced USART Asynchronous Receive or Addressable USART Synchronous Data.
RB5/PGM/KBI1	bit 5	TTL(1)/ST(6)	Input/output port pin (with interrupt-on-change). Internal software programmable weak pull-up. Low-Voltage ICSP enable pin.
RB6/PGC/T1OSO/T13CKI/ P1C/KBI2	bit 6	TTL(1)/ST(6,8)	Input/output port pin (with interrupt-on-change), Timer1/ Timer3 clock input or Timer1oscillator output. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD/T1OSI/P1D/KBI3	bit 7	TTL(1)/ST(6)	Input/output port pin (with interrupt-on-change) or Timer1 oscillator input. Internal software programmable weak pull-up. Serial programming data.

PIC 18F … Datasheet, le minimum

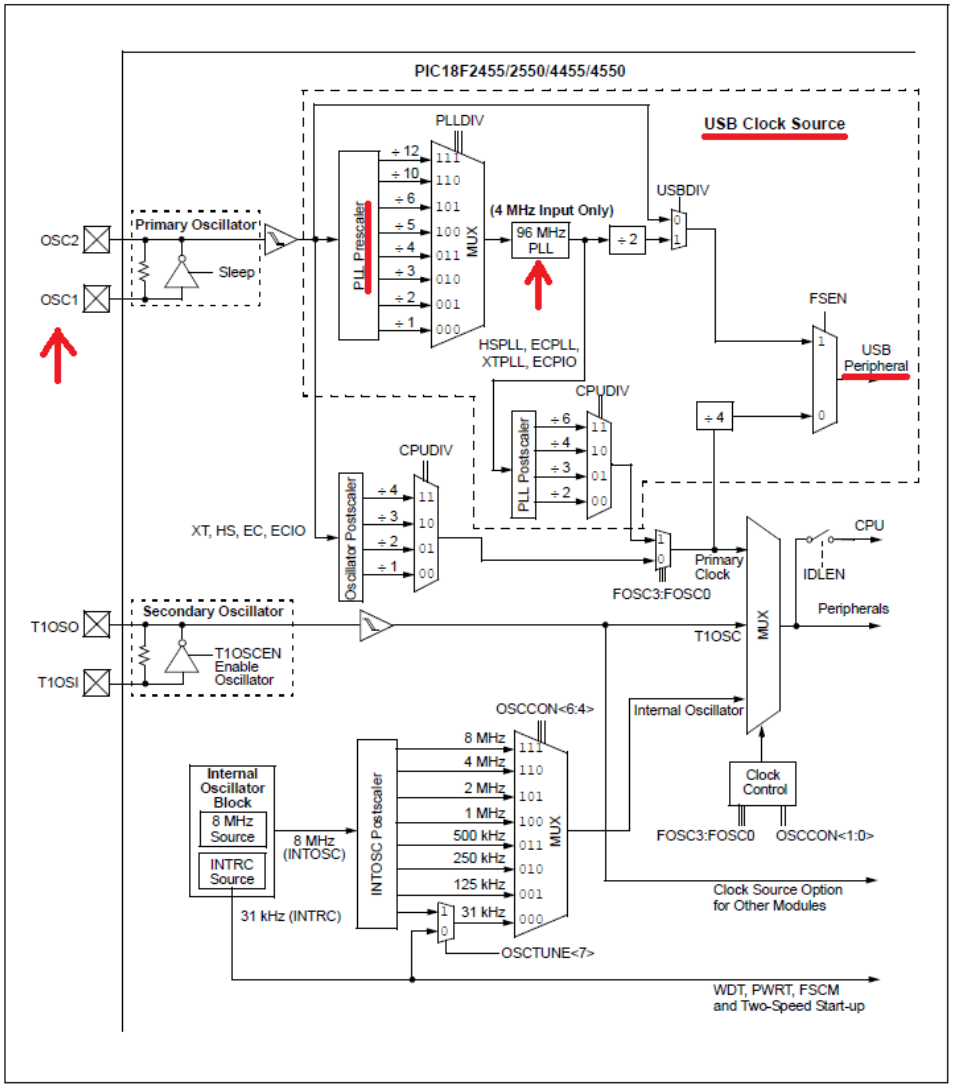
- Registres : deux types, « variables » internes et « configuration » du comportement du µC
  - Oscillateur : Interne, externe, vitesse ;
  - Timers ;
  - Interruptions ;
  - …

PIC 18F … datasheet, le minimum

- Schéma d'implantation interne : aide à la compréhension.

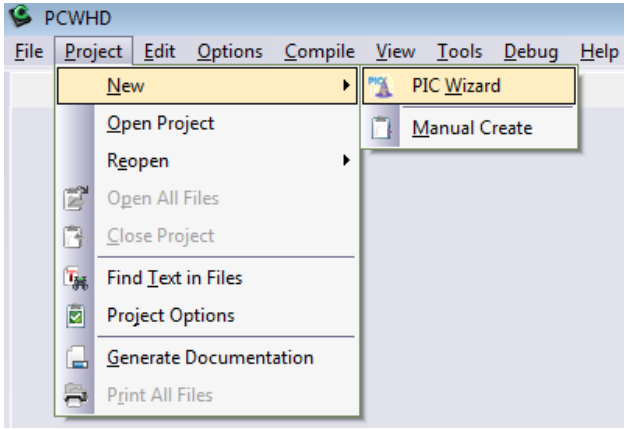
  Ex. Valeur oscillateur pour USB? (18F2550)

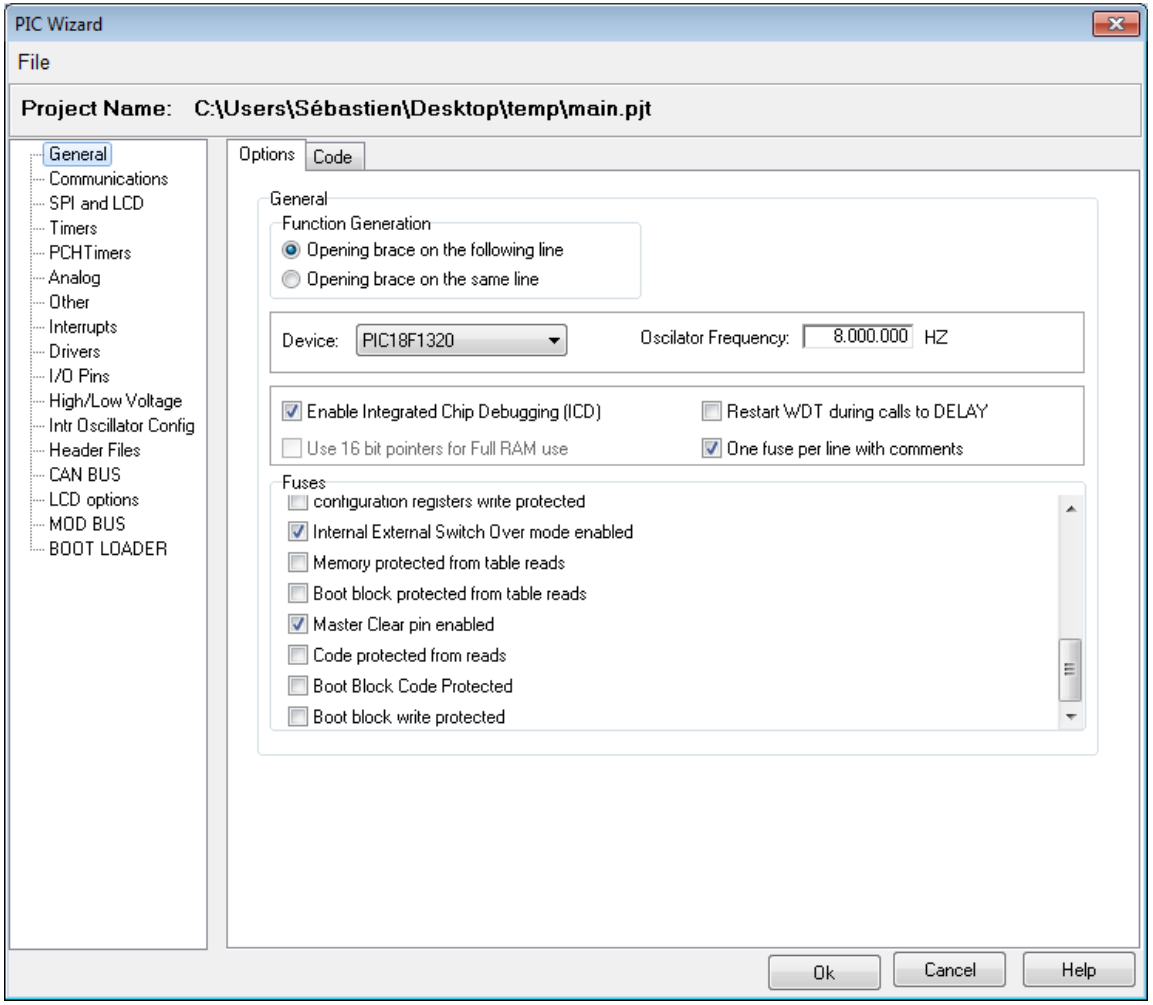
FIGURE 2-1: PIC18F2455/2550/4455/4550 CLOCK DIAGRAM



Compilateur CCS C

- Les registres, notion de « fusible »: Astuce, PCW.EXE, Génération facilitée





Compilateur CCS C

- Fichier généré par PCW : « main.h »

```
#device ICD=TRUE
#device adc=16

#FUSES NOWDT                    //No Watch Dog Timer
#FUSES WDT128                   //Watch Dog Timer uses 1:128 Postscale
#FUSES INTRC                    //Internal RC Osc
#FUSES FCMEN                    //Fail-safe clock monitor enabled
#FUSES BROWNOUT                 //Reset when brownout detected
#FUSES BORV27                   //Brownout reset at 2.7V
#FUSES NOPUT                    //No Power Up Timer
#FUSES NOCPD                    //No EE protection
#FUSES STVREN                   //Stack full/underflow will cause reset
#FUSES DEBUG                    //Debug mode for use with ICD
#FUSES LVP                      //Low Voltage Programming on B3(PIC16) or B5(PIC18)
#FUSES NOWRT                    //Program memory not write protected
#FUSES NOWRTD                   //Data EEPROM not write protected
#FUSES NOWRTC                   //configuration not registers write protected
#FUSES IESO                     //Internal External Switch Over mode enabled
#FUSES NOEBTR                   //Memory not protected from table reads
#FUSES NOEBTRB                  //Boot block not protected from table reads
#FUSES MCLR                     //Master Clear pin enabled
#FUSES NOPROTECT                //Code not protected from reading
#FUSES NOCPB                    //No Boot Block code protection
#FUSES NOWRTB                   //Boot block not write protected

#use delay(clock=8000000)
```

Compilateur CCS C

- Fichier généré par PCW : « main.c »

```
#include "main.h"


void main()
{

  setup_adc_ports(sAN4|sAN5|VSS_VDD);
  setup_adc(ADC_OFF|ADC_TAD_MUL_0);
  setup_wdt(WDT_OFF);
  setup_timer_0(RTCC_INTERNAL);
  setup_timer_1(T1_DISABLED);
  setup_timer_2(T2_DISABLED,0,1);
  setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
  setup_oscillator(OSC_8MHZ|OSC_INTRC);

  // TODO: USER CODE!!

}
```

Compilateur CCS C

- Instructions de base
  - Entrées et Sorties :
    - output_high(PIN_xx);
    - output_low(PIN_xx);
    - *Variable* = Input(PIN_xx);
  - Structures conditionnelles :
    - If (*condition*) { *instruction* } else { *instruction* }
    - Switch (*variable*) {
      case x : *instructions*
      break;
      case y : *instructions*
      break;
      default: *instructions*
      break;
      }
  - Boucles
    - Do {
      instruction
      } while *(condition);*
    - *For* (conditions) *{*
      instructions
      };
  - Fonctions et Procédures
    - Void *nom* (*paramètres*) *{*
      instructions
      }
    - *Type nom* (*paramètres*) *{*
      instructions
      }

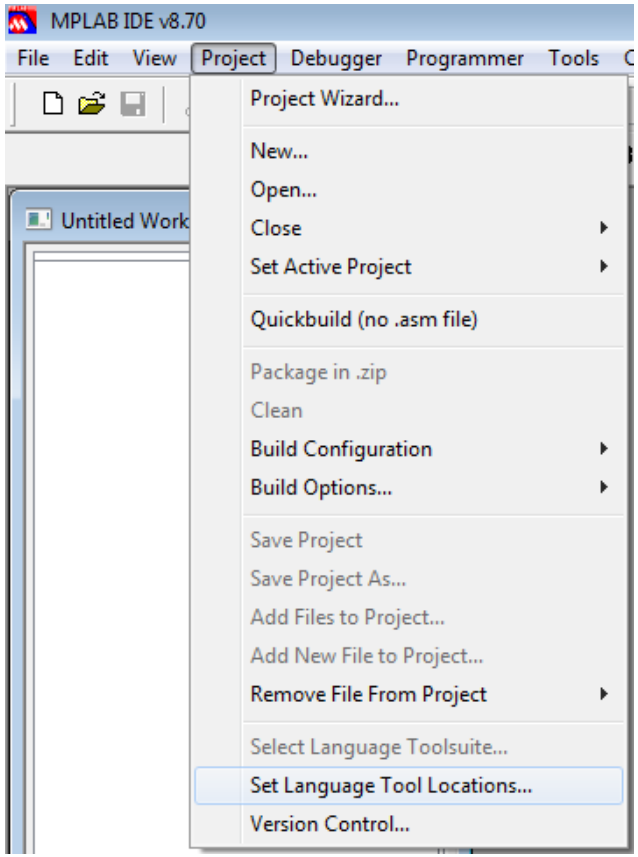
➔ Voir documentation du compilateur : « C compiler reference manual »

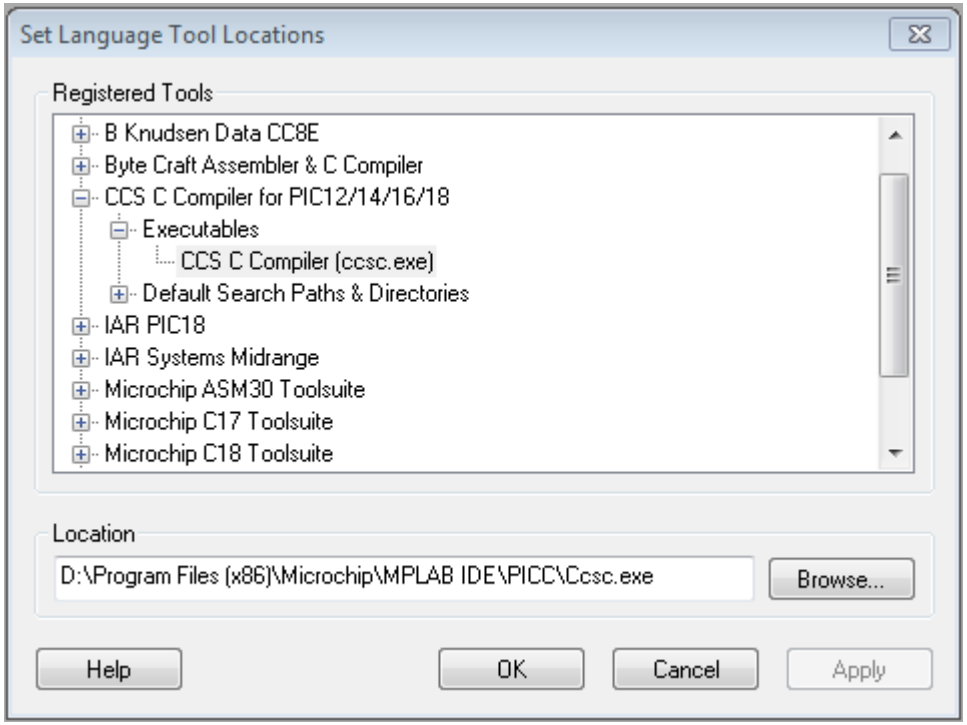
Programmateurs

- Pickit : Microchip. Version 2, +/- 33 euros TVAC chez Farnell
- ICD : Copie chinoise sur ebay, < 30 euros
- Home-Made : Internet

IDE, Mplab

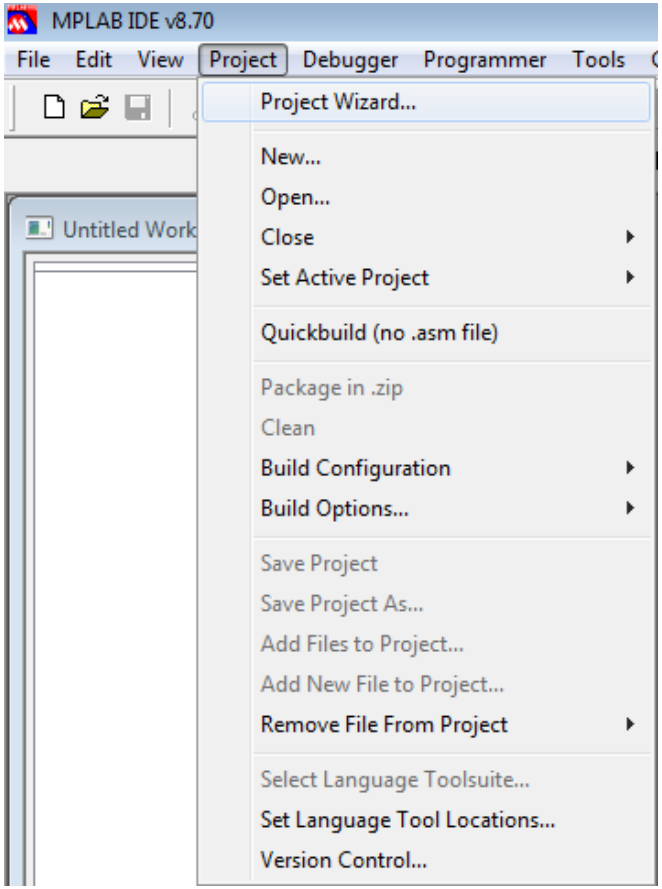
- Configuration du compilateur

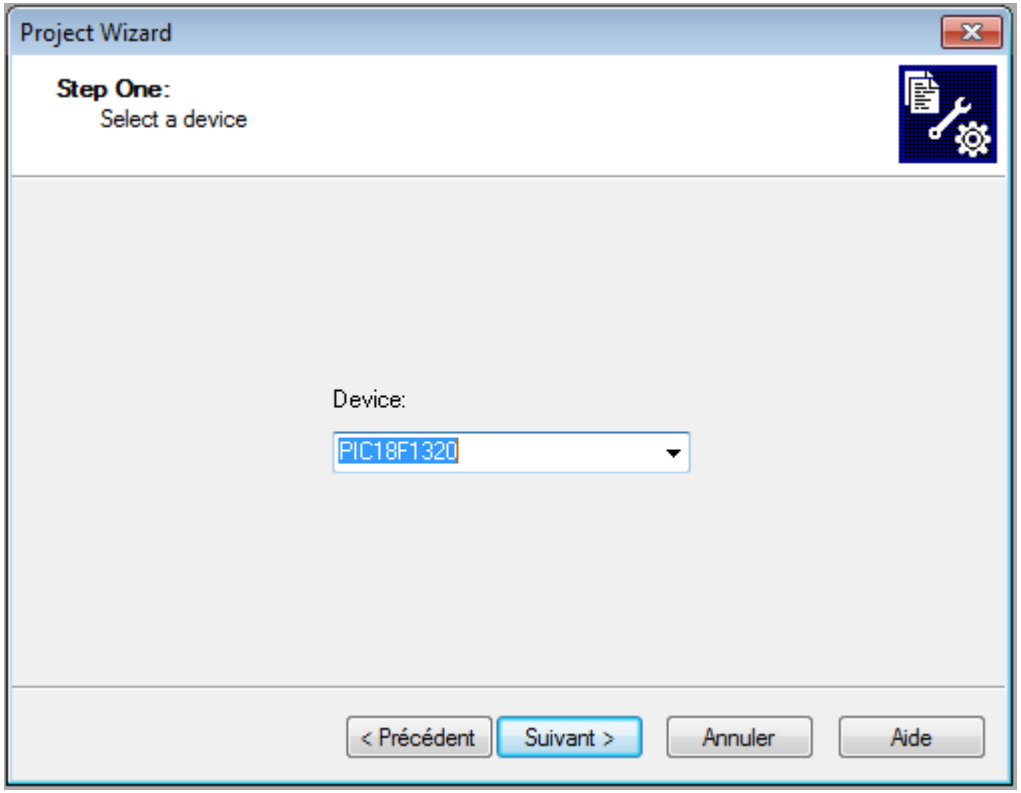




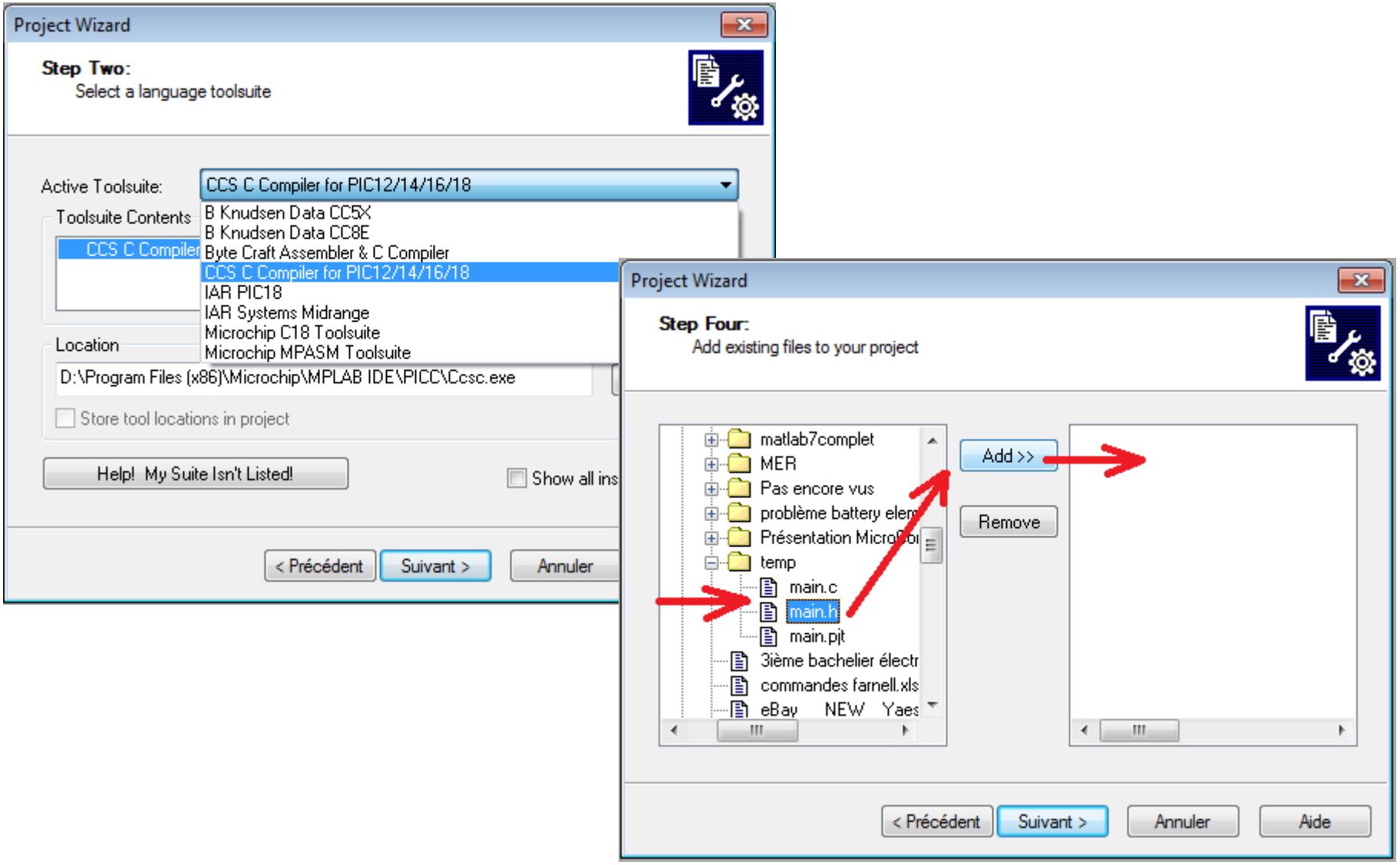
IDE, Mplab

- Nouveau projet



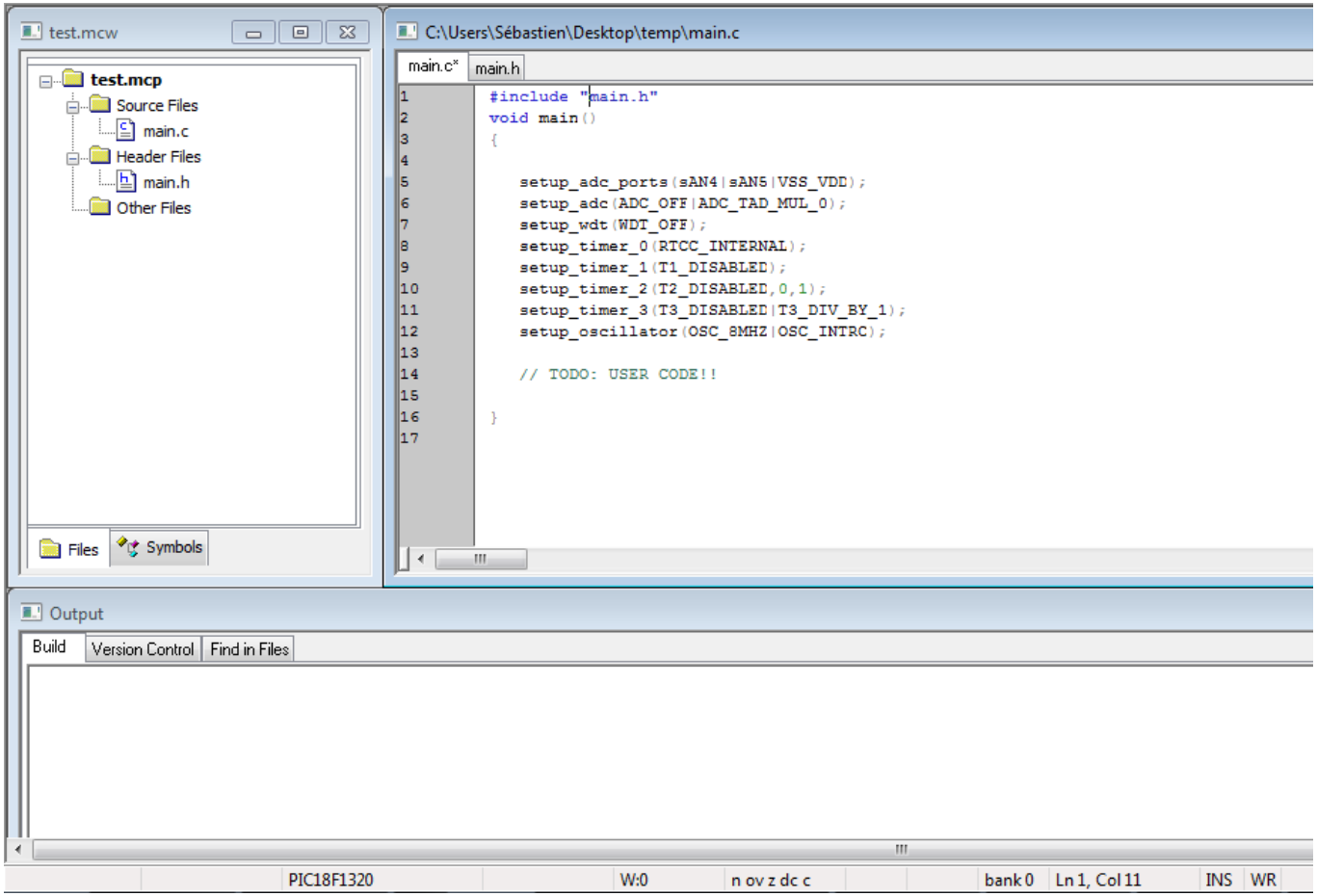


IDE, Mplab


Project Wizard
Step Two:
Select a language toolsuite
Active Toolsuite:
CCS C Compiler for PIC12/14/16/18
Toolsuite Contents
CCS C Compiler
B Knudsen Data CC5X
B Knudsen Data CC8E
Byte Craft Assembler & C Compiler
CCS C Compiler for PIC12/14/16/18
IAR PIC18
IAR Systems Midrange
Microchip C18 Toolsuite
Microchip MPASM Toolsuite
Location
D:\Program Files (x86)\Microchip\MPLAB IDE\PICC\Ccsc.exe
Store tool locations in project
Help! My Suite Isn't Listed!
Show all ins
< Précédent
Suivant >
Annuler
Project Wizard
Step Four:
Add existing files to your project
matlab7complet
MER
Pas encore vus
problème battery elem
Présentation Micro
temp
main.c
main.h
main.pjt
3ième bachelier électr
commandes farnell.xls
eBay NEW Yaes
Add >>
Remove
< Précédent
Suivant >
Annuler
Aide

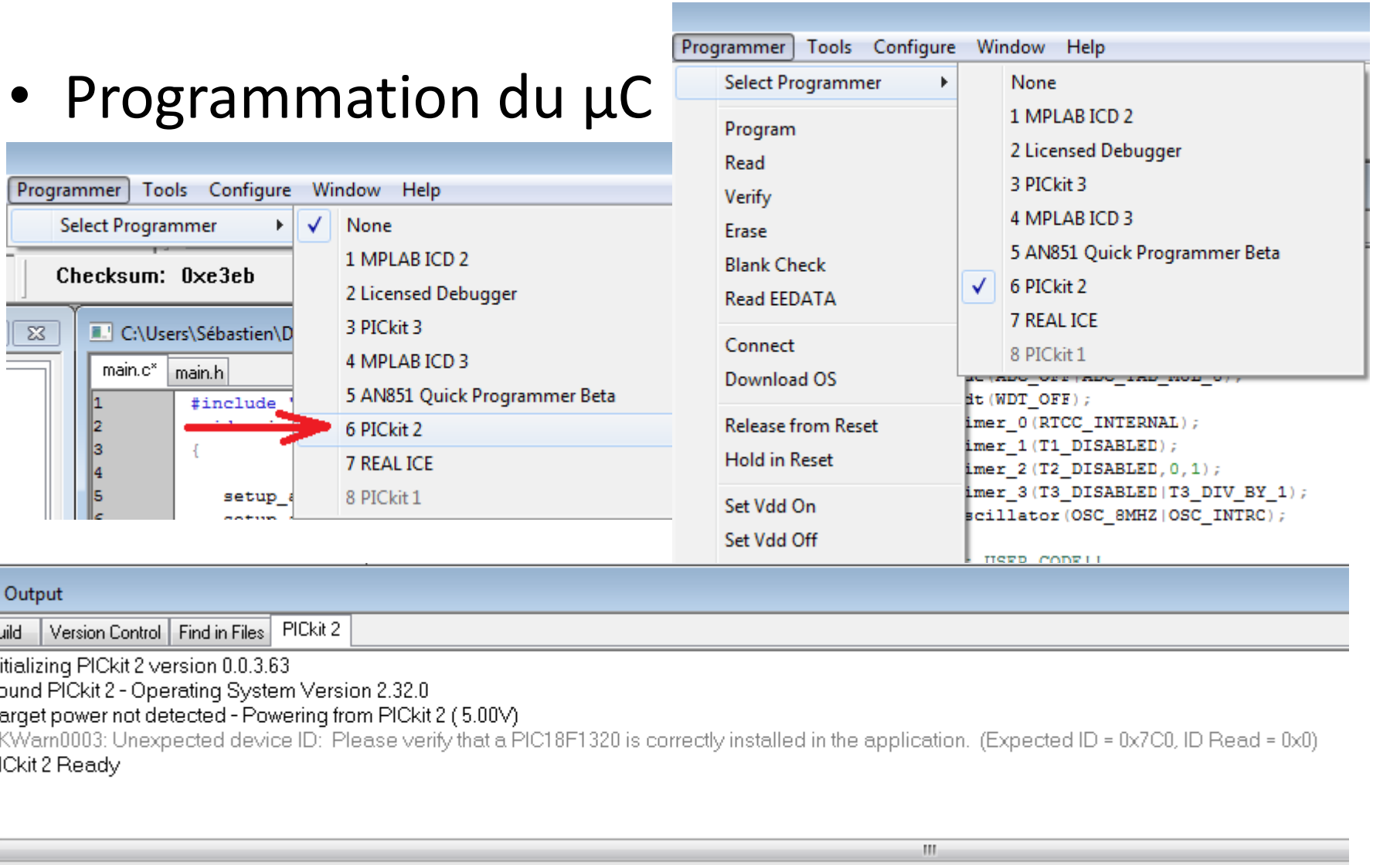
IDE, Mplab

- Environnement

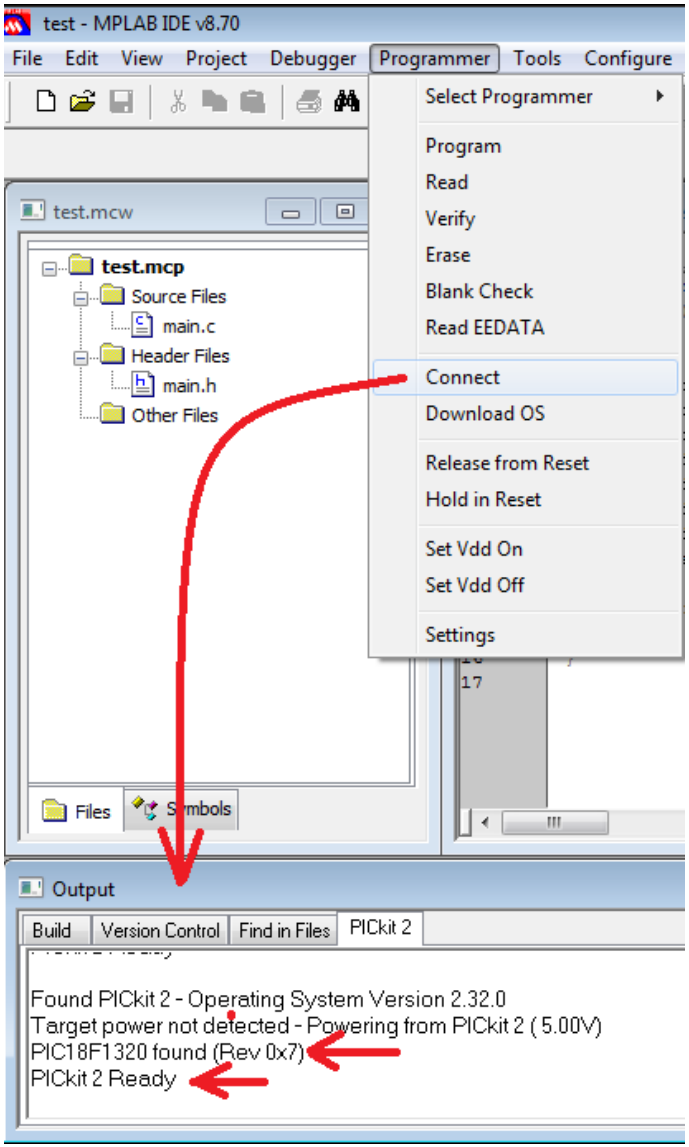


IDE, Mplab

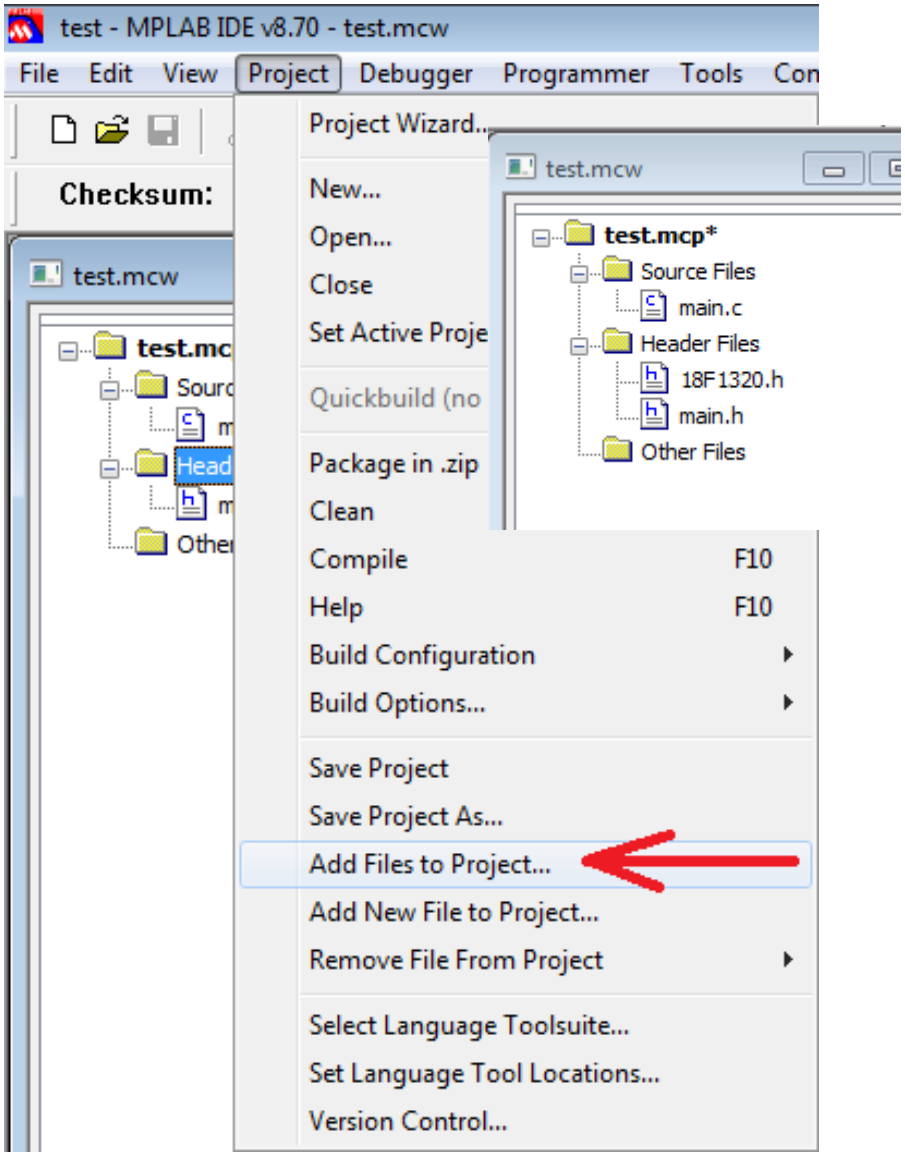
- Programmation du µC

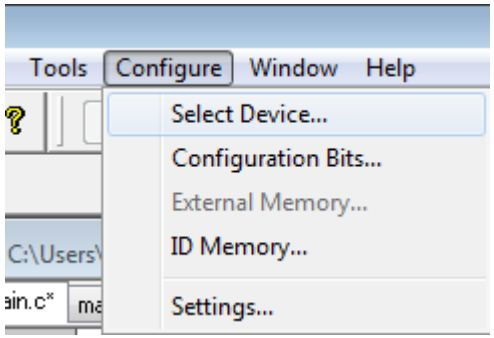


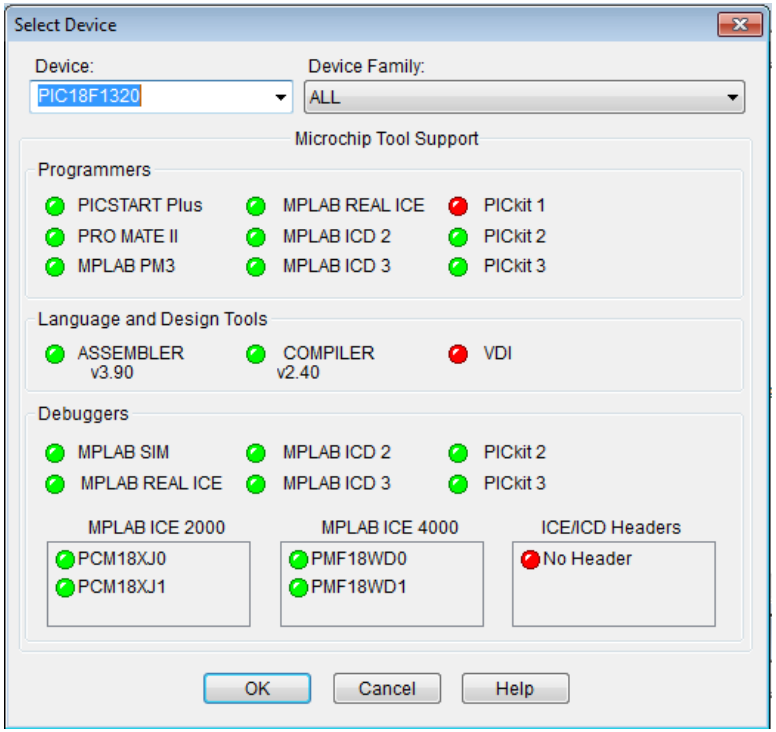
IDE, Mplab


test - MPLAB IDE v8.70
File Edit View Project Debugger Programmer Tools Configure
Select Programmer
Program
Read
Verify
Erase
Blank Check
Read EEDATA
Connect
Download OS
Release from Reset
Hold in Reset
Set Vdd On
Set Vdd Off
Settings
test.mcw
test.mcp
Source Files
main.c
Header Files
main.h
Other Files
Files
Symbols
Output
Build
Version Control
Find in Files
PICkit 2
Found PICkit 2 - Operating System Version 2.32.0
Target power not detected - Powering from PICkit 2 (5.00V)
PIC18F1320 found (Rev 0x7)
PICkit 2 Ready

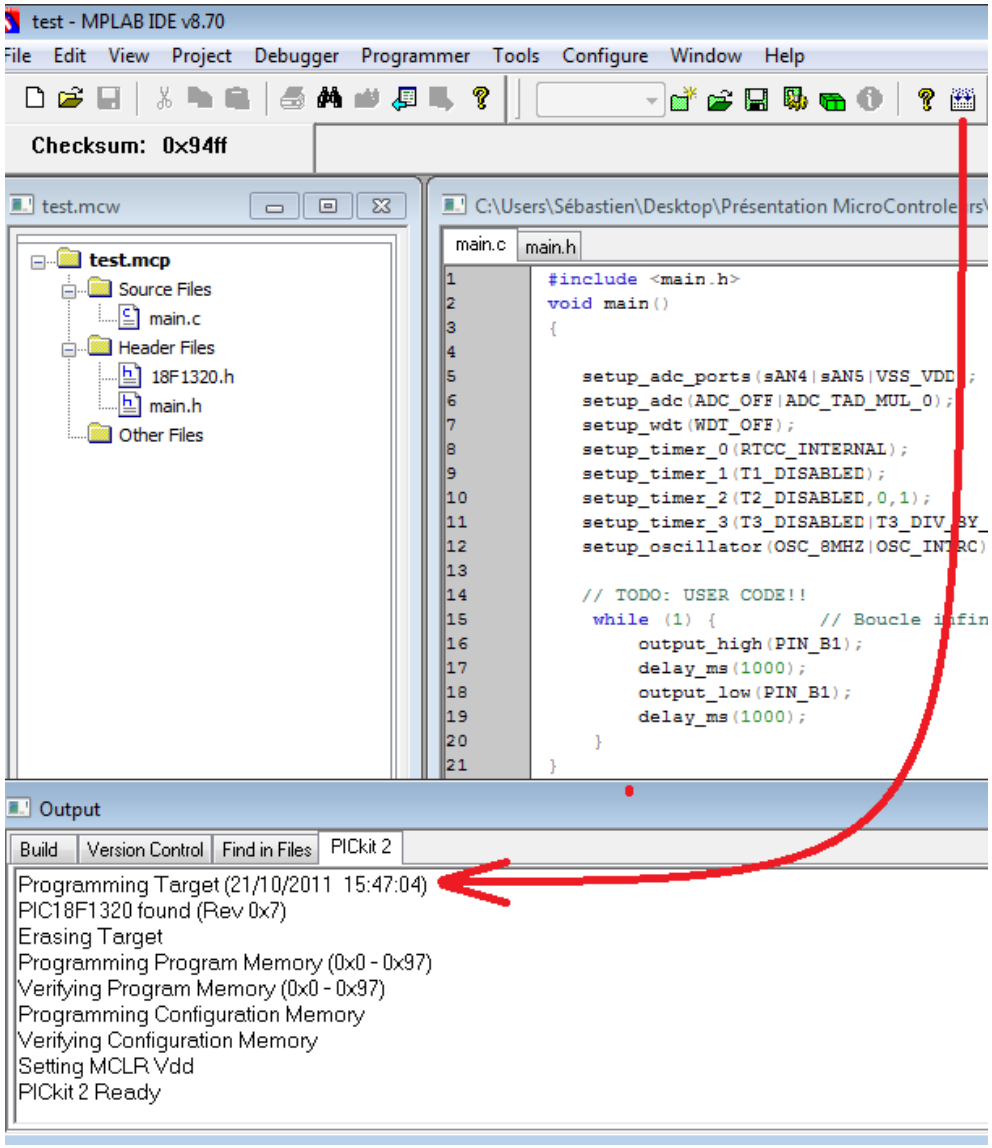
IDE, Mplab


test - MPLAB IDE v8.70 - test.mcw
File Edit View Project Debugger Programmer Tools Con
Checksum:
test.mcw
test.mcp*
Source Files
main.c
Header Files
18F1320.h
main.h
Other Files
Project Wizard...
New...
Open...
Close
Set Active Proje
Quickbuild (no
Package in .zip
Clean
Compile F10
Help F10
Build Configuration
Build Options...
Save Project
Save Project As...
Add Files to Project...
Add New File to Project...
Remove File From Project
Select Language Toolsuite...
Set Language Tool Locations...
Version Control...


Tools Configure Window Help
Select Device...
Configuration Bits...
External Memory...
ID Memory...
Settings...


Select Device
Device:
PIC18F1320
Device Family:
ALL
Microchip Tool Support
Programmers
PICSTART Plus
PRO MATE II
MPLAB PM3
MPLAB REAL ICE
MPLAB ICD 2
MPLAB ICD 3
PICkit 1
PICkit 2
PICkit 3
Language and Design Tools
ASSEMBLER v3.90
COMPILER v2.40
VDI
Debuggers
MPLAB SIM
MPLAB REAL ICE
MPLAB ICD 2
MPLAB ICD 3
PICkit 2
PICkit 3
MPLAB ICE 2000
PCM18XJ0
PCM18XJ1
MPLAB ICE 4000
PMF18WD0
PMF18WD1
ICE/ICD Headers
No Header
OK
Cancel
Help

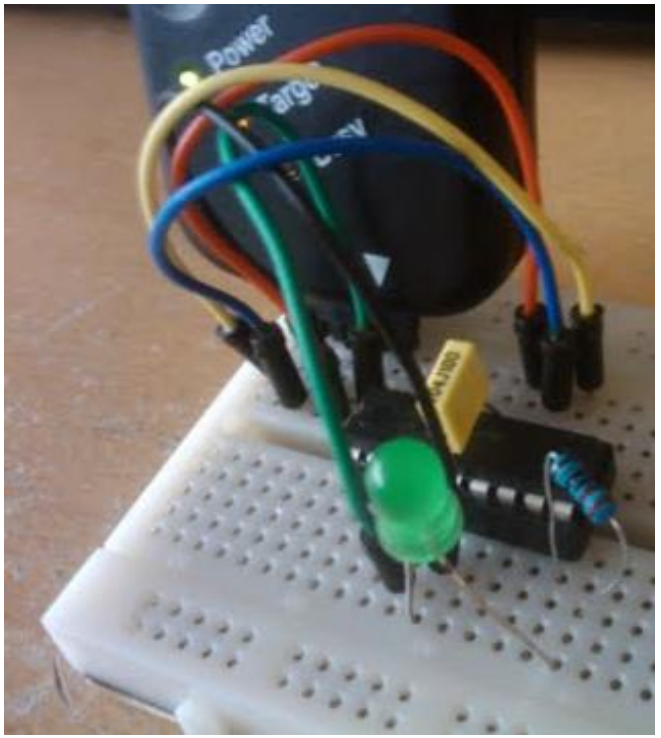
IDE, Mplab

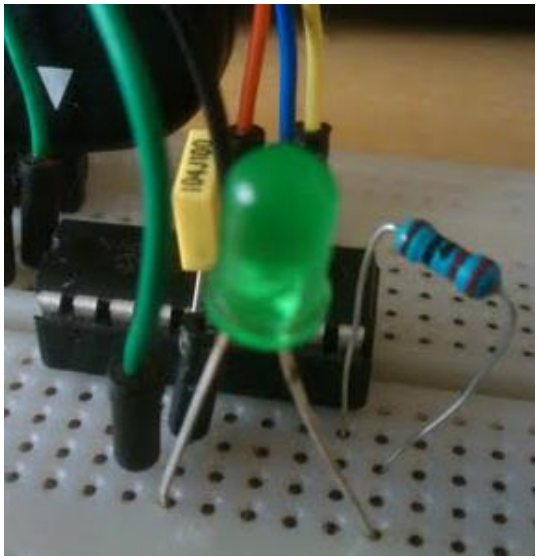

test - MPLAB IDE v8.70
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0x94ff
test.mcw
test.mcp
Source Files
main.c
Header Files
18F1320.h
main.h
Other Files
C:\Users\Sébastien\Desktop\Présentation MicroControle rs\
main.c
main.h
#include <main.h>
void main()
{
setup_adc_ports(sAN4|sAN5|VSS_VDD);
setup_adc(ADC_OFF|ADC_TAD_MUL_0);
setup_wdt(WDT_OFF);
setup_timer_0(RTCC_INTERNAL);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DISABLED,0,1);
setup_oscillator(OSC_8MHZ|OSC_INTRC)
// TODO: USER CODE!!
while (1) {
output_high(PIN_B1);
delay_ms(1000);
output_low(PIN_B1);
delay_ms(1000);
}
}
Output
Build
Version Control
Find in Files
PICkit 2
Programming Target (21/10/2011 15:47:04)
PIC18F1320 found (Rev 0x7)
Erasing Target
Programming Program Memory (0x0 - 0x97)
Verifying Program Memory (0x0 - 0x97)
Programming Configuration Memory
Verifying Configuration Memory
Setting MCLR Vdd
PICkit 2 Ready

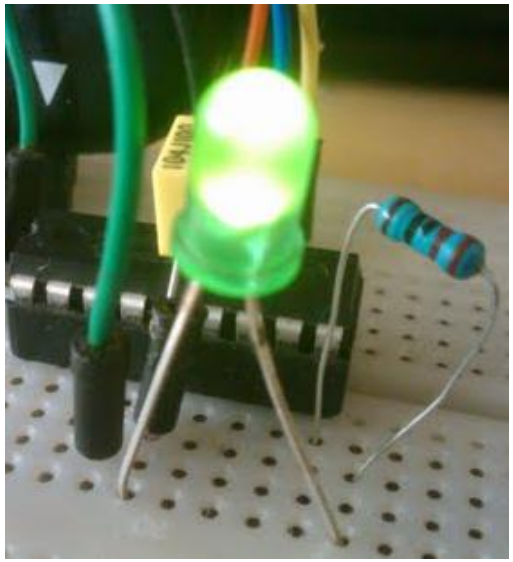
Programme test

- LED clignottante

```
while (1) {          // Boucle infinie : Sans elle, le programme ne s'exécuterait qu'une seule fois
        output_high(PIN_B1);
        delay_ms(1000);
        output_low(PIN_B1);
        delay_ms(1000);
}
```







Aller plus loin avec les µC

- Interruptions
  - Timers : Déclenchement d'un sous-programme sur base temporelle ;
  - Interruptions externes : Déclenchement d'un sous-programme sur base d'un stimulus extérieur ;
- Bus : Communication entre CI / entre applications.
  - I^2C
  - SPI
  - CAN
  - USB (18F2550 / 18F4550), émulation RS232 sur USB, p.ex.
  - Ethernet

Et donc, à quoi ça sert?

- CAN / CNA → Mesure/Source de tension/courant → Commande analogique
- Mesure/Source en tout ou rien → Commande numérique
- Modulation PWM (Pulse Width Modulation) → Variation de niveaux
- Bus de communications : Sondes de t°, hygrométrie, pression, …

Exemple d’applications

- Jeu de lumière (Matrice RGB en 3D, vumètre)
- Mesure de courant et tension (Multimètre, Sondes analogiques, Coupleur bidirectionnel VHF/UHF)
- Perroquet HF, ISD1730 et bus SPI
- Régulation (Automates programmables)
- Commande de moteurs (Fraiseuse CNC, imprimantes, lecteurs DVD)
- Afficheurs LCD
- Interfaçage informatique (USB, Ethernet, RS232)
- Etc.

En savoir plus…

- Algorithmique, informatique générale
- Manuel CCS C
- Datasheets
- Thomas Floyd « Systèmes numériques »
- John Catsoulis « Embedded hardware »
- Vincent Lemieux « L'USB pour tous »
- Xavier Fenard « Le bus USB, guide du concepteur »