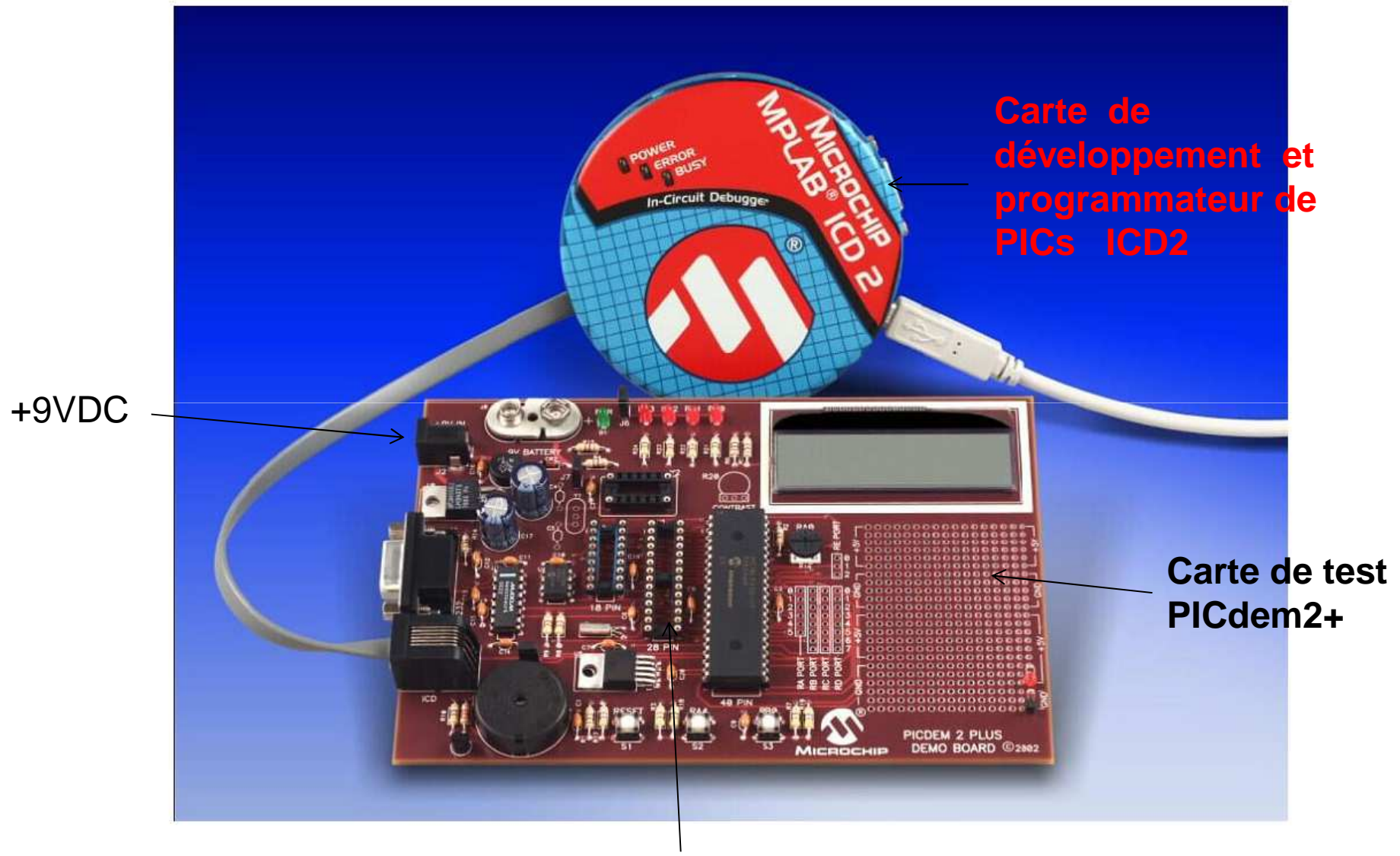


Programmation de **microcontrôleurs PIC** de Microchip  
*Initiation au langage C avec le compilateur PCW et MPLAB*



+9VDC

Carte de développement et programmeur de PICs ICD2

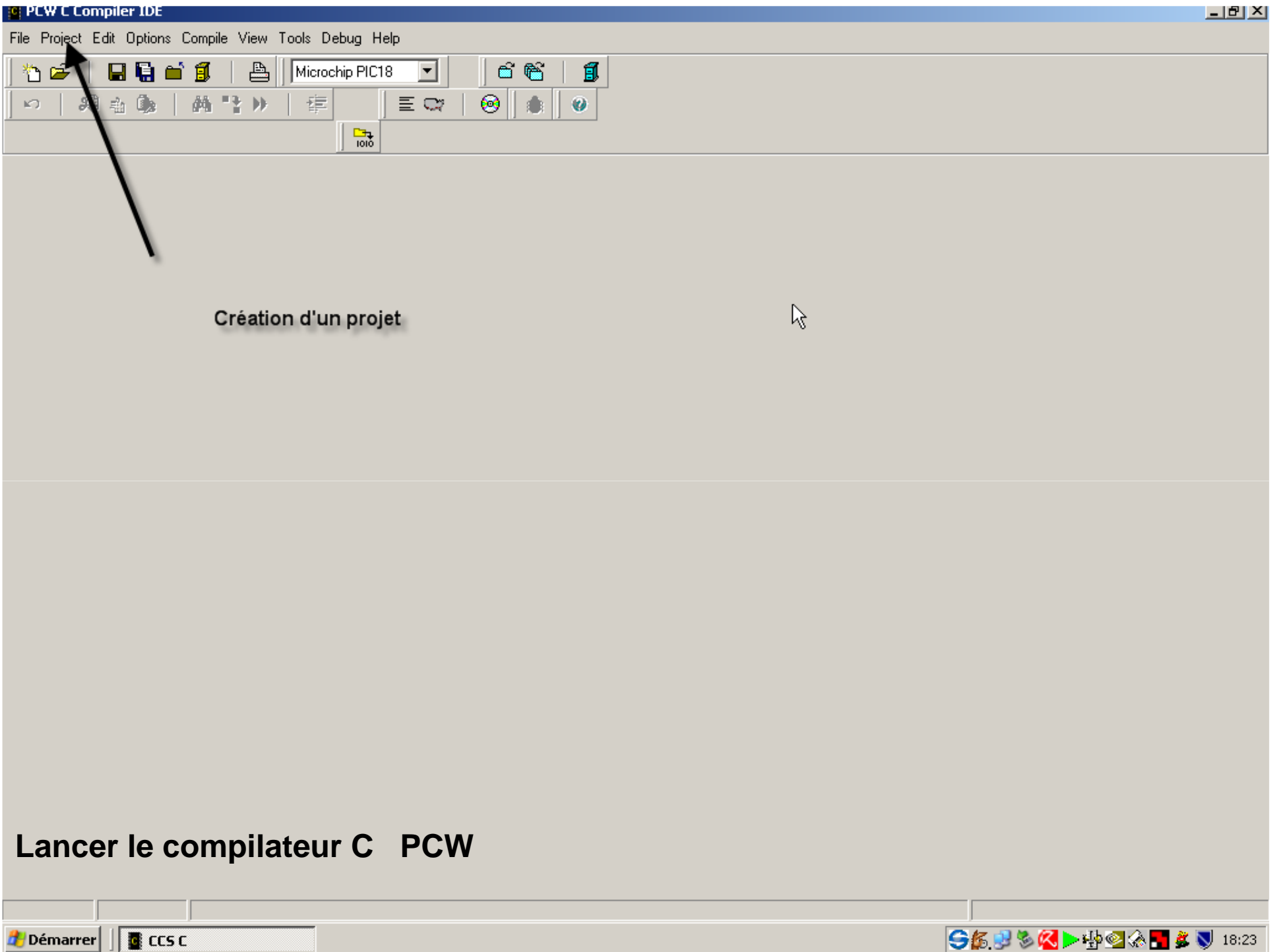
Carte de test PICdem2+

Supports de CI pour µC PICs

# Procédure pour éditer (en C), compiler, transférer un programme dans un microcontrôleur PIC

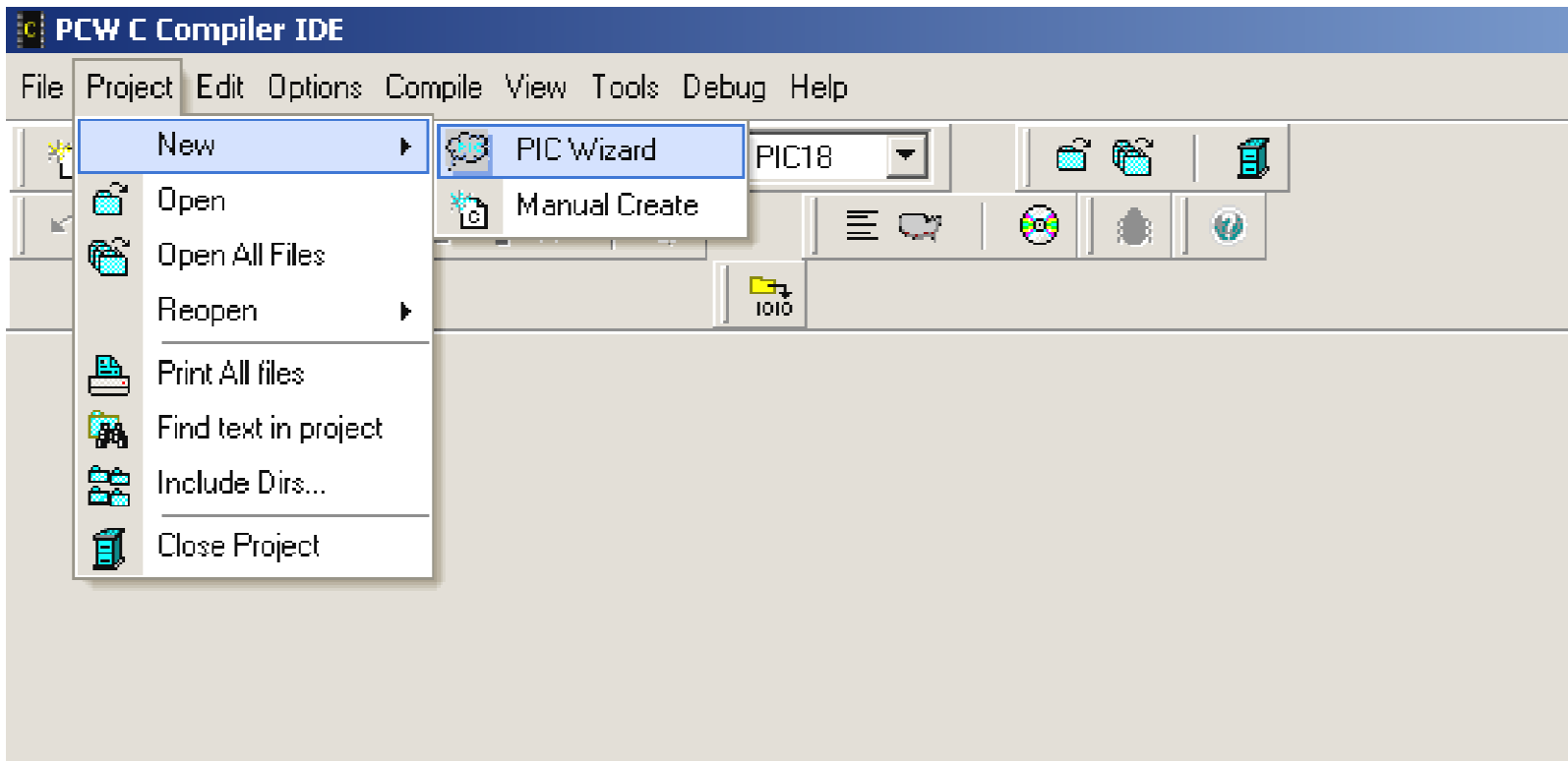


Cliquer sur cet icône ou aller sous  
C:\Program Files\PICC\Pcw.exe



Création d'un projet

Lancer le compilateur C PCW





**Enregistrer sous** [?] [X]

Enregistrer dans : 1ELGR1 [Refresh] [Up] [Down] [Home] [Back]

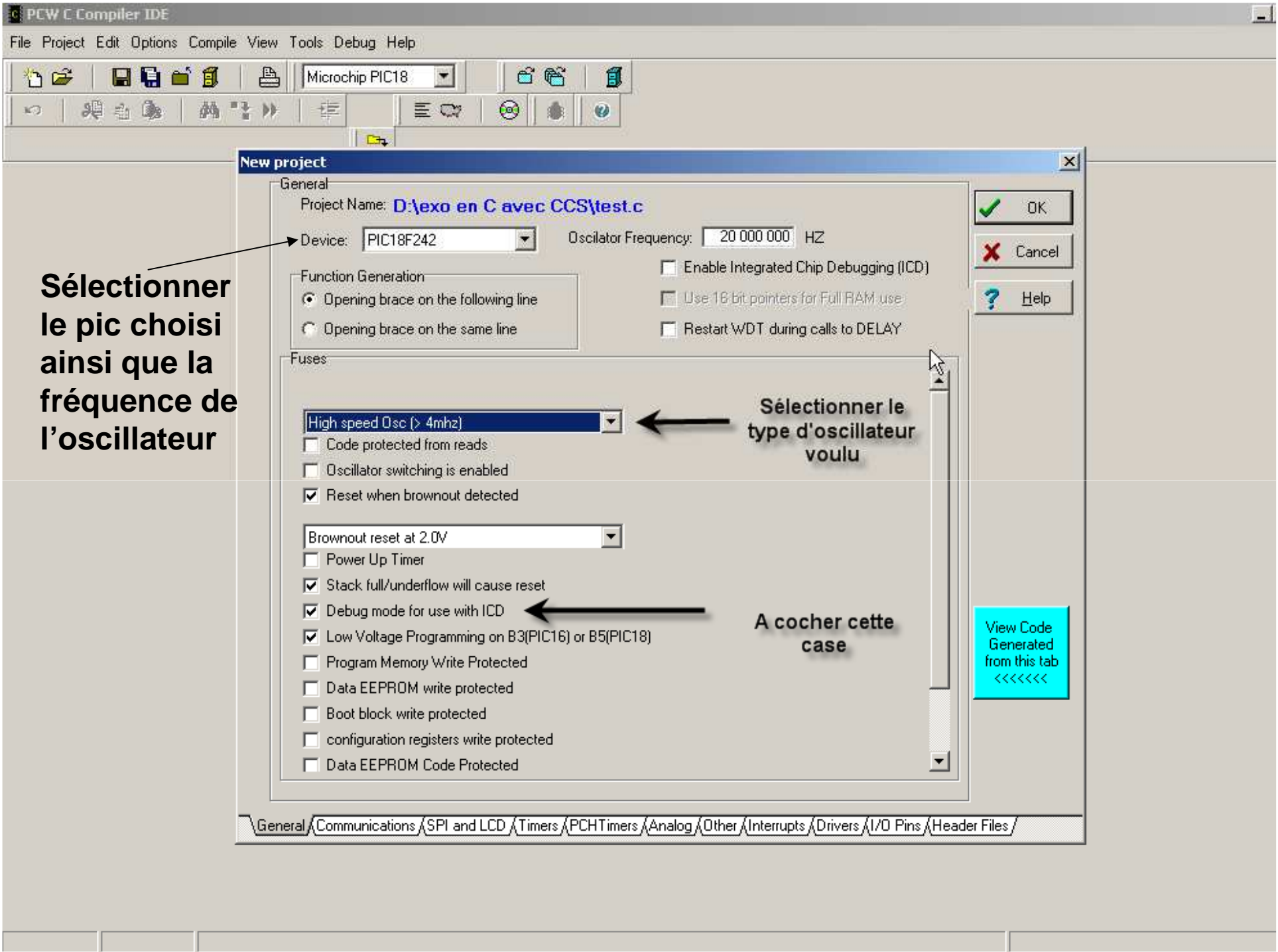
Mes documents récents  
Bureau  
Mes documents  
Poste de travail  
Favoris réseau

**Donner un nom à votre projet et l'enregistrer sous D:\1ELGR1 par exemple**

Nom du fichier :  [v]

Type :  [v]

Enregistrer Annuler





**New project** [X]

Communications

**RS-232**

Use RS-232 **La liaison série RS-232 n'est pas paramétrée**

RS232#1

Baud: 9600  restart\_wdt Bits: 9

Parity: None  Invert

Transmit: C6  Float\_high  Enable pin: A0

Receive: C7  Errors Stream:

BRGH10K

**I2C**

Use I2C  Master  Fast

SDA: C4  Slave  Slow

SCL: C3  Restart\_wdt Slave Address:

Force\_hw

Hardware PSP

View Code Generated from this tab <<<<<<

General / Communications / SPI and LCD / Timers / PCHTimers / Analog / Other / Interrupts / Drivers / I/O Pins / Header Files

## New project

Analog Input

Analog Pins

- None
- A0 A1 A2 A3 A5 E0 E1 E2 Ref=Vdd
- A0 A1 A2 A5 E0 E1 E2 Ref=A3
- A0 A1 A2 A3 A5 Ref=Vdd
- A0 A1 A2 A5 Ref=A3
- A0 A1 A3 Ref=Vdd
- A0 A1 Ref=A3
- A0 A1 A5 E0 E1 E2 Ref=A2,A3
- A0 A1 A2 A3 A5 E0 Ref=Vdd
- A0 A1 A2 A5 E0 Ref=A3
- A0 A1 A5 E0 Ref=A2,A3
- A0 A1 A5 Ref=A2,A3
- A0
- A0 Ref=A2,A3

sélection de la voie  
A0 du port A (ligne  
RA0)

Units: 0-255  
Units: 0-255  
Units: 0-1023  
Units: 0-65472  
Internal 2-bus

résolution du CAN sur  
8 bits (256 valeurs) , 10  
bits (1024 valeurs) ou  
16 bits

View Code  
Generated  
from this tab  
<<<<<<

Paramétrage du CAN intégré au  $\mu$ C PIC

General / Communications / SPI and LCD / Timers / PCHTimers / **Analog** / Other / Interrupts / Drivers / I/O Pins / Header Files



### New project

Pin names

Pin:	I/O Type	Identifiers
A0	Input	PIN_A0
A1	Input	PIN_A1
A2	Output	PIN_A2
A3	Input/Output	PIN_A3
A4	Not used	PIN_A4
A5	Analog	PIN_A5
A6	Input	PIN_A6
B0	Input	PIN_B0
B1	Input	PIN_B1
B2	Input	PIN_B2
B3	Input	PIN_B3
B4	Input	PIN_B4
B5	Input	PIN_B5
B6	Input	PIN_B6
B7	Input	PIN_B7
C0	Input	PIN_C0

Sélectionner la ligne désirée en entrée ou en sortie suivant l'application désirée.

Enable Pullups on port B

List multiple identifiers by separating with commas

OK  
Cancel  
Help

View Code Generated from this tab <<<<<<

General / Communications / SPI and LCD / Timers / PCHTimers / Analog / Other / Interrupts / Drivers / I/O Pins / Header Files



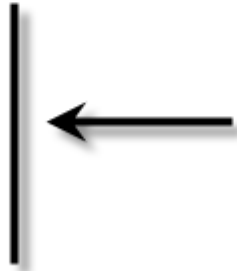
```
#include "D:\exo en C avec CCS\test.h"
```

```
void main()
```

```
{
```

```
    setup_adc_ports(NO_ANALOGS);  
    setup_adc(ADC_OFF);  
    setup_spi(FALSE);  
    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);
```

```
}
```



Paramètres d'initialisation du PIC  
intégré au programme.

I

```
File Project Edit Options Compile View Tools Debug Help
Microchip PIC18
test.c test.h
#include <18F452.h>
#device adc=8
#use delay(clock=4000000)
#fuses NOWDT,WDT128,XT, NOPROTECT, NOOSEN, BROWNOUT, BORV20,
```

µC PIC utilisé  
résolution numérique du CAN  
Fréquence du quartz

Pour contrôler votre paramétrage du PIC, ouvrir le fichier `****.h` et vérifier si le paramétrage est correct.

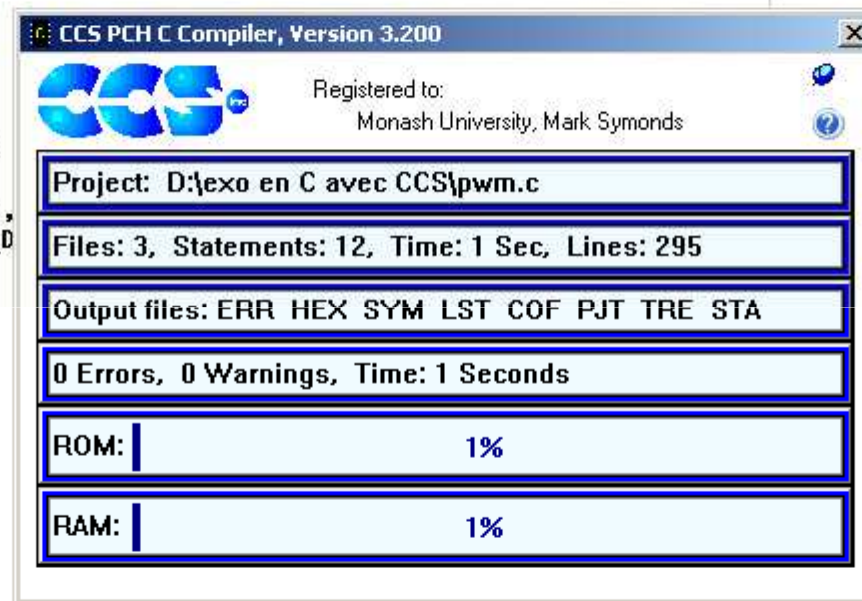


```
#include "D:\exo en C avec CCS\pwm.h"
```

Taper le programme et le compiler .  
Penser à le sauvegarder.

```
void main()  
{
```

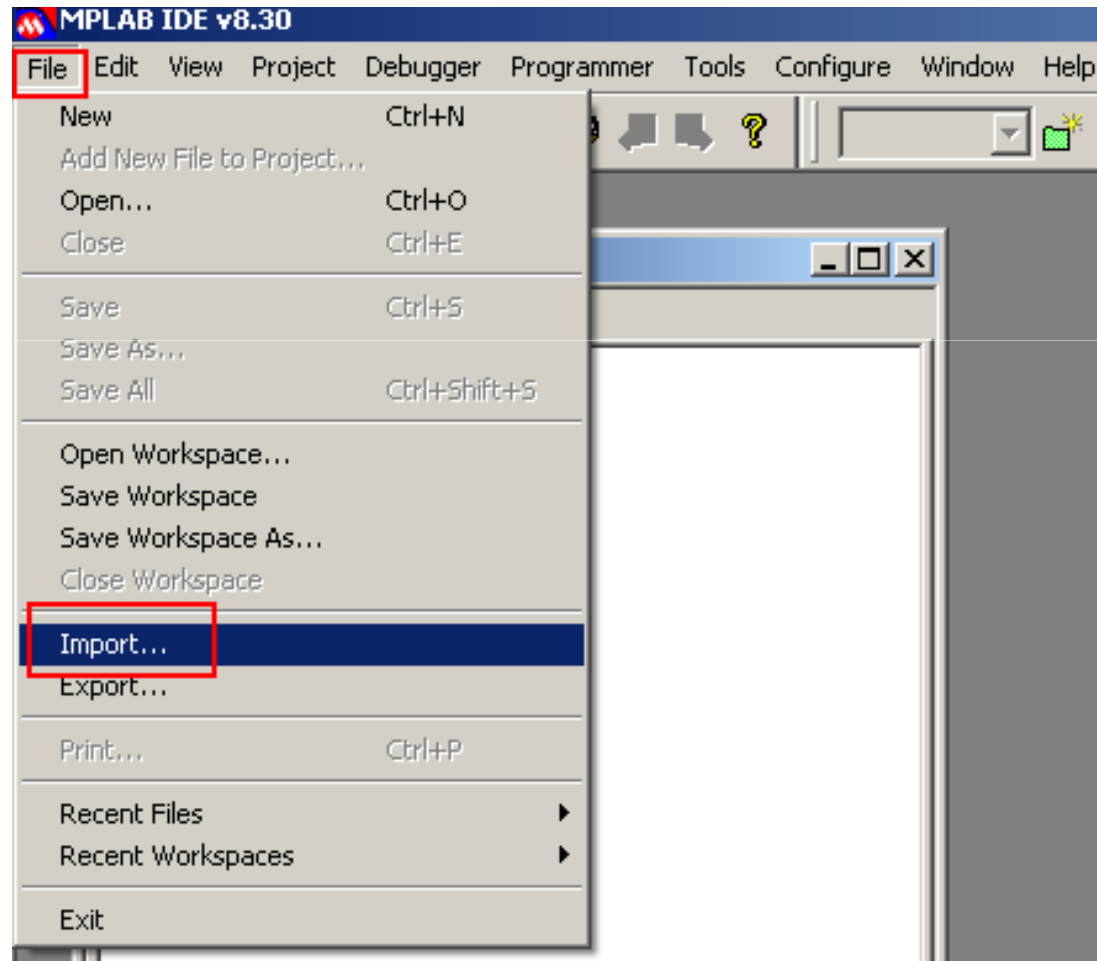
```
    setup_adc_ports(NO_ANALOGS);  
    setup_adc(ADC_OFF);  
    setup_spi(FALSE);  
    setup_wdt(WDT_OFF);  
    setup_timer_0(RTCC_INTERNAL);  
    setup_timer_1(T1_DISABLED);  
    setup_timer_2(T2_DIV_BY_1,255,  
    setup_timer_3(T3_DISABLED|T3_D  
    setup_ccp1(CCP_PWM);  
    set_pwm1_duty(500);  
    while(1);  
}
```



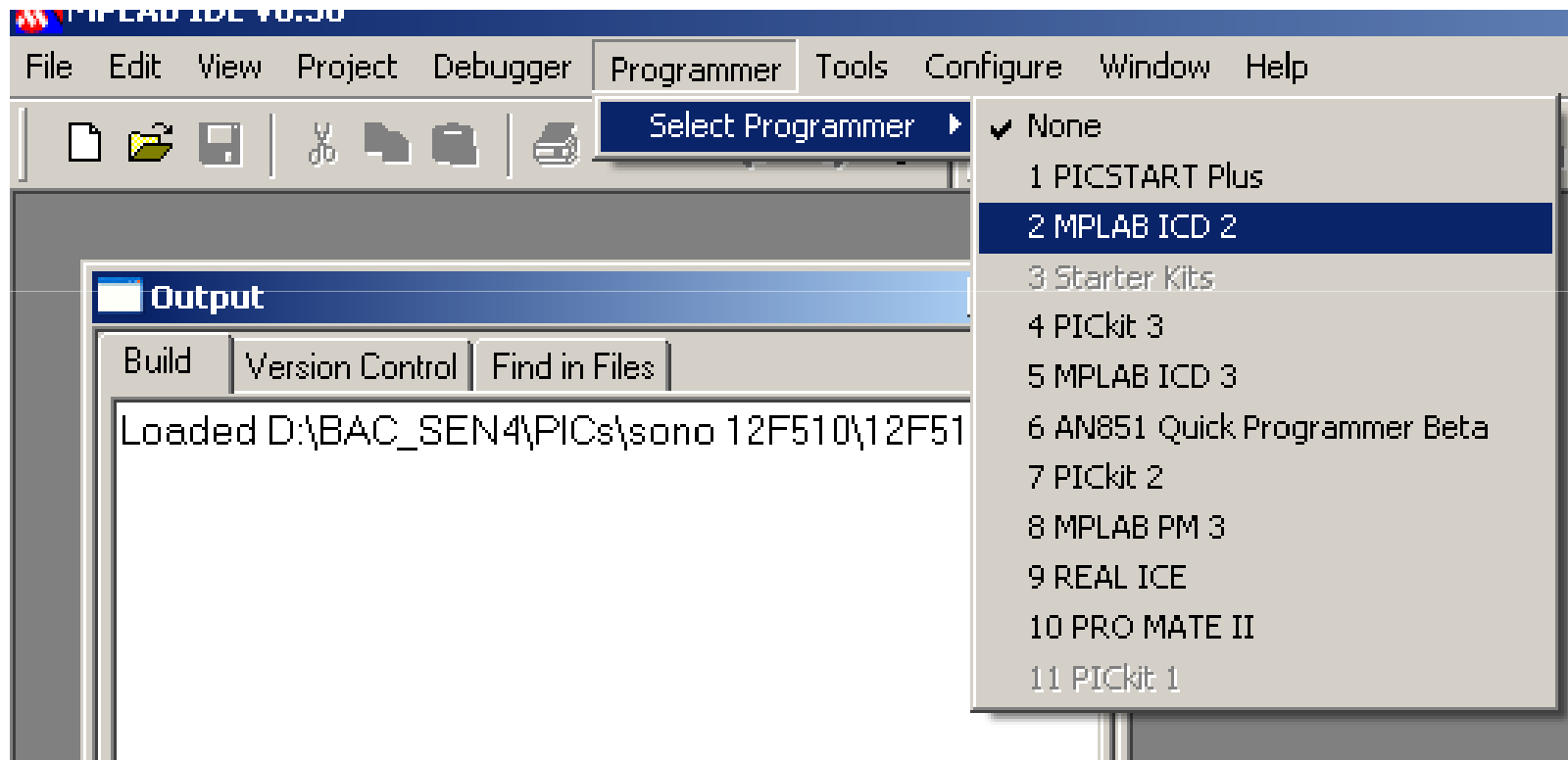
**Pour charger le programme dans un microcontrôleur PIC , nous allons utiliser le logiciel MPLAB et le programmeur ICD2 .**

**1) Ouvrir MPLAB**

**2) Importer le fichier \*\*\*\*\* .hex**



**Une fois le fichier .hex chargé , sélectionner le programmeur ICD2 pour charger ce programme dans le microcontrôleur PIC choisi.**





Output

id | Version Control | Find in Files | MPLAB ICD 2

```
Connecting to MPLAB ICD 2
Connected
Setting Vdd source to target
Target Device PIC18F242 found, revision = 0x5
Reading ICD Product ID
Running ICD Self Test
Passed
MPLAB ICD 2 Ready
```

Cliquer sur ce pictogramme pour charger le programme dans la mémoire flash du PIC.

```
    }
    if (!input(PIN_B4))
    {
        delay_ms(24000);
    }
}

void main()
```



**Output**

id | Version Control | Find in Files | MPLAB ICD 2

```
passed
MPLAB ICD 2 Ready
Programming Target...
Erasing Part
Programming Program Memory (0x0 - 0x1AF)
Verifying...
Programming Program Memory
Verify Succeeded
Programming Configuration Bits
Erasing Configuration Memory
Verifying configuration memory...
Verify Succeeded
Programming succeeded
MPLAB ICD 2 Ready
```

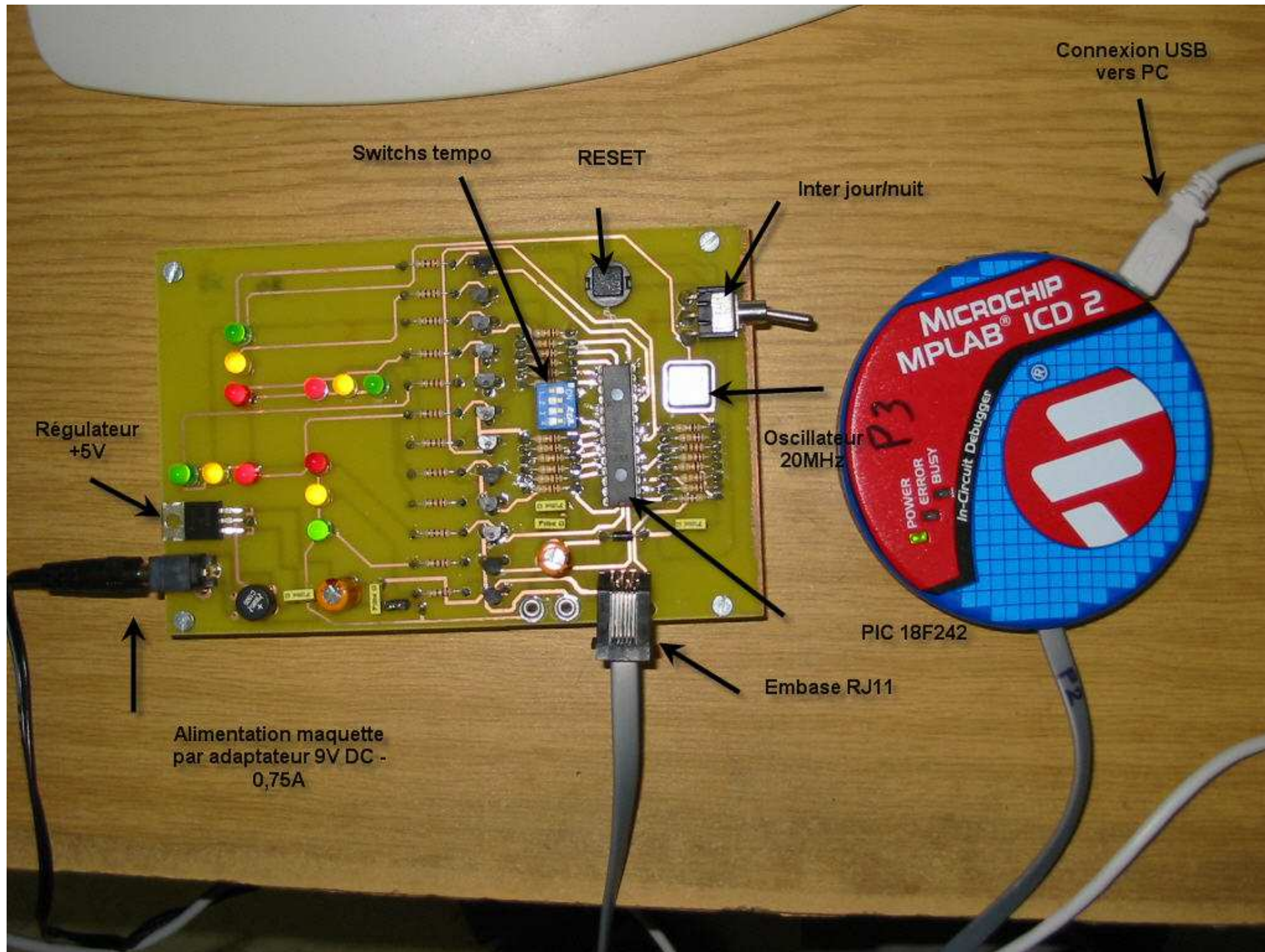
**Programmation du PIC réussie . Vous pouvez déconnecter le module ICD 2 de la maquette et tester le bon fonctionnement du programme sur la maquette.**

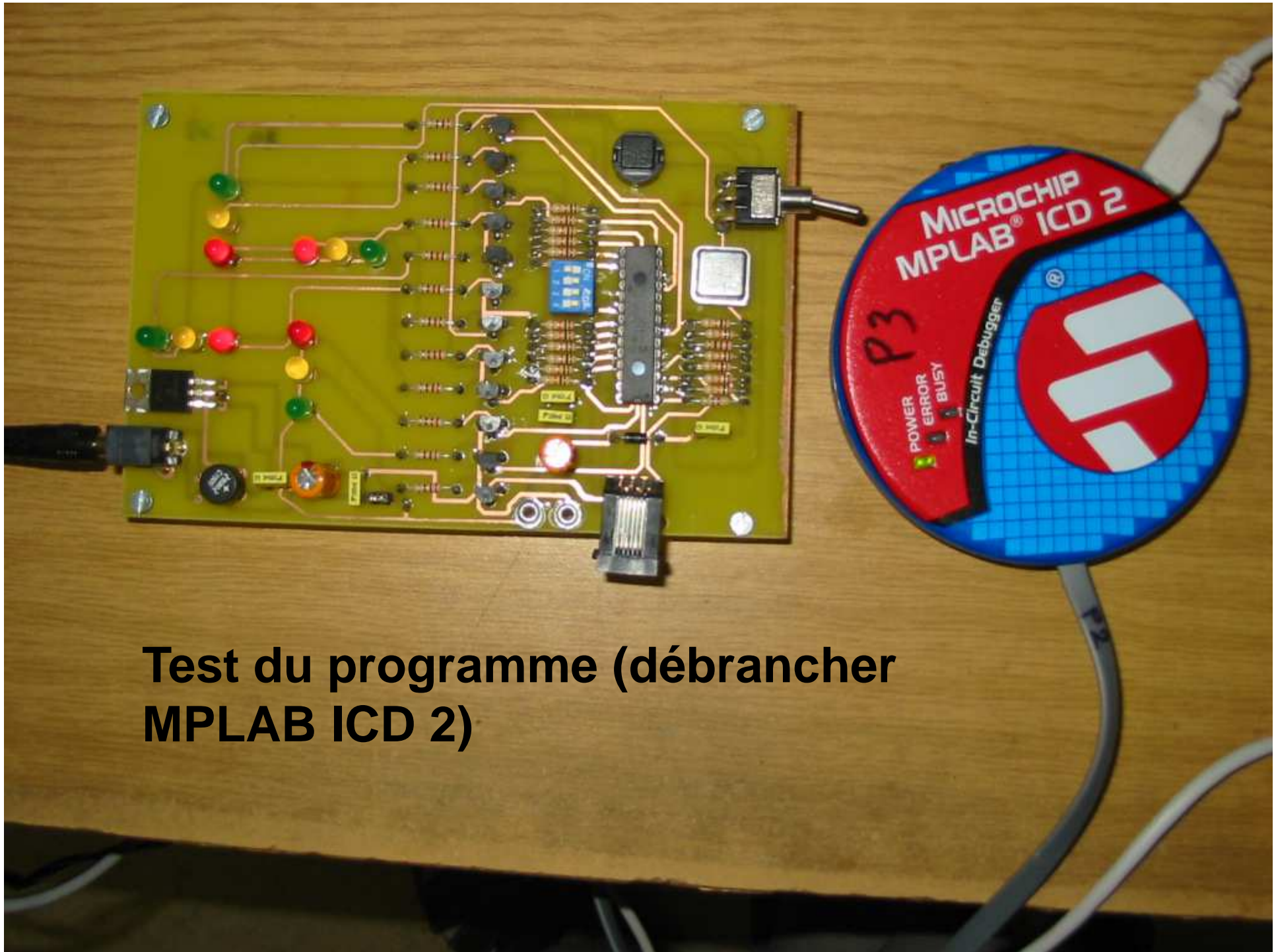


```
    }
    if (!input (PIN_B4))
    {
        delay_ms (24000);
    }
}

void main()
```







**Test du programme (débrancher  
MPLAB ICD 2)**