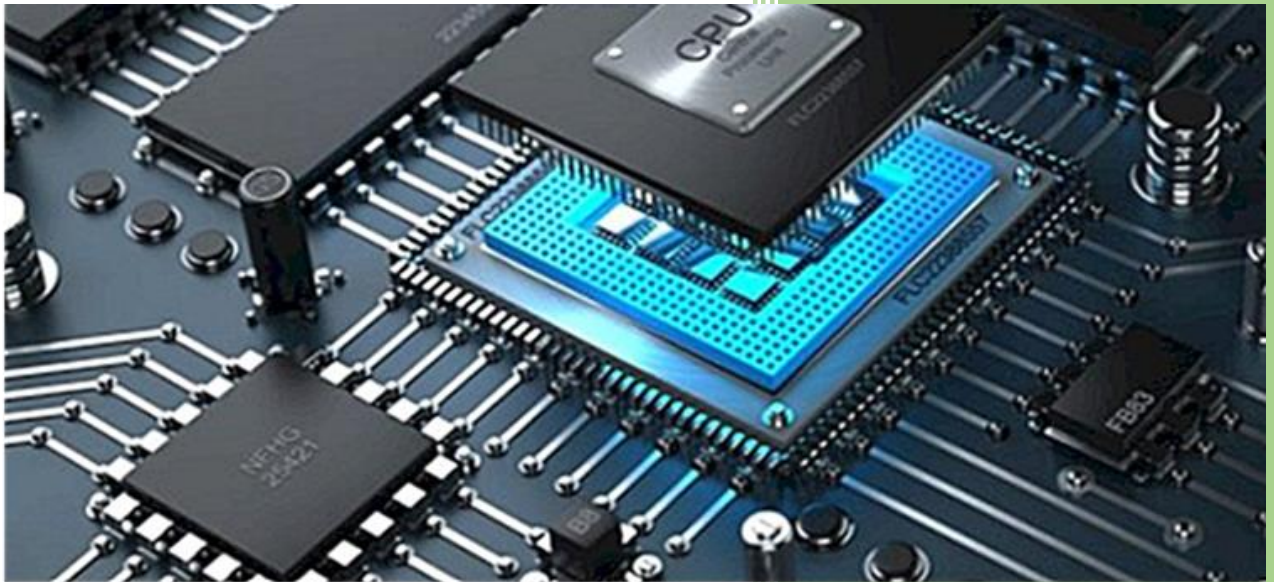


2022

TP1 FISA 26 Microcontrôleur



ANTINI Nicolas

CESI

01/04/2022

TP N°1

Mise en œuvre des ports d'entrée/sortie logique .

Objectifs du TP :

Découvrir l'environnement de développement.

Se familiariser avec la mise en œuvre des PORTS du PIC 18f87k22.

Matériel utilisé :

- Malette MDET,
- Un PC équipé des logiciels MPLABX et son compilateur XC8.
- Un oscilloscope numérique,
- Un analyseur logique .
- Documentation des logiciels et du micro contrôleur.

Expérimentation n°1:

Effectuer un chenillard à 8 LED , l'appui sur BP1 change le sens de rotation.
L'appui sur BP2 change la vitesse de rotation (environ 0,5s / 1s / 2s / 4s).

L'appuie simultanée sur les 2 boutons poussoir BP1 et BP2 permet la recopie du motif défini par les SWITCHS sur les 4 leds de poids fort

Expérimentation n°2 :

Simuler ce processus à l'aide du simulateur FLOWCODE.

A l'aide du debugger et de l'analyseur FLOWCODE fournir des chronogrammes démontrant le bon fonctionnement de notre CHENILLARD.
Générer et créer un projet incluant le code c généré par FLOWCODE et compatible MPLABX .



Directement sur le site Microchip :

Téléchargement de MPLABX

Téléchargement de XC8

Exécutez XC8

Selon Notice Préparation TP Microcontrôleurs

Créer un nouveau projet

File > New project

Choisir la catégorie Microchip Embedded puis Standalone project > Next

Choisir la famille Advanced 8-bit (PIC18) puis PIC18F87K22 > Next

Dans Hardware Tool, sélectionner PicKit 3 > Next

Dans Compiler toolchain, sélectionner XC8 > Next

Renseigner le nom du projet (ex : TP1_12-02-2013)

Choisir l'emplacement du projet

Cocher "Set as main project" > Finish

Dans le projet, cliquer droit sur Source files > New > C main file

Entrer le nom > Finish

Dans le projet, cliquer droit sur Header files > Add existing item

Sélectionner le fichier xc.h dans [C:\program files\Microchip\xc8\include\](#)

Attention si vous obtenez l'erreur target device was not found lors de la programmation du pic vérifiez l'option d'alimentation du pickit :

Run > Set project configuration > Customize





Cliquer sur Pickit à gauche dans Categories

Puis dans option categories sélectionner Power

Cocher power target from pickit et choisissez 5V

Simulateur

On peut sélectionner le simulateur à la place du pickit 3 lors de la création du projet.

Si le projet est déjà créé, on peut modifier le projet :

MENU Run

Set project configuration

Customize...

Dans hardware tool, sélectionner simulator OK

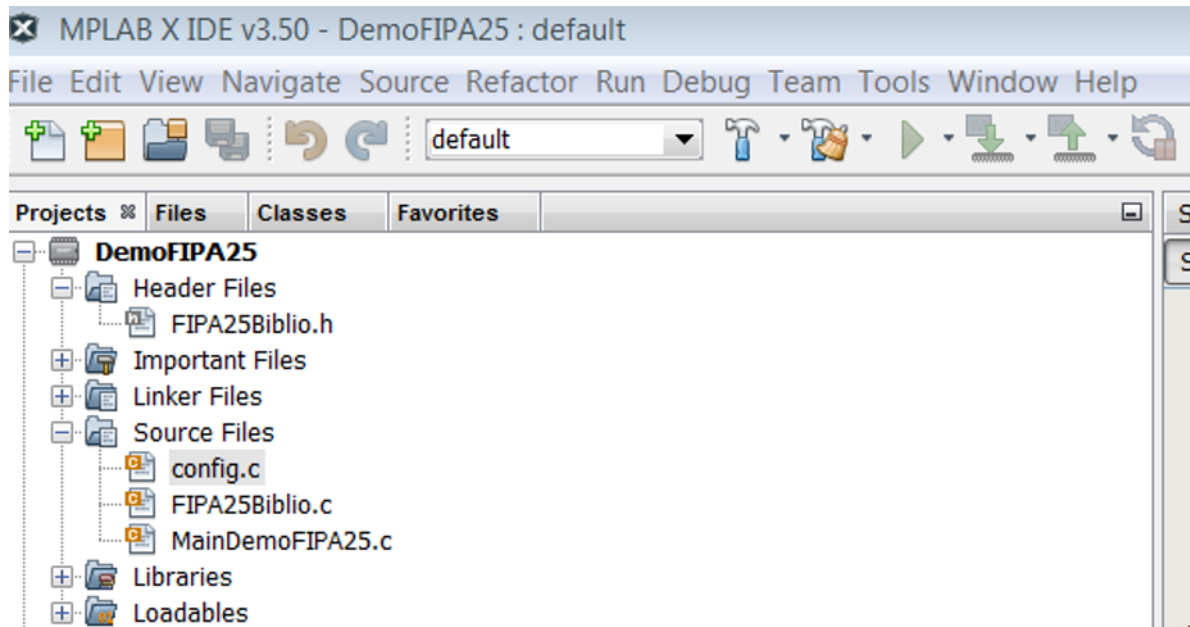
Pour lancer le simulator cliquer sur l'icone debug project ou dans le menu debug/debug project

Pour simuler avec le simulateur, la configuration en debug du pic n'a pas d'importance (CONFIG4L=0x81 ou 0x01)





Détail du projet (ex DemoFIPA26) :



FIPA25Biblio.h regroupe les entêtes de mes fonctions.

Dans le TP1 :

```
void Init(void) ;
```

```
void Tempo(unsigned char val) ;
```

```
void T_Sens(void) ; // Tâche dédiée à la gestion du sens de  
chenillard
```

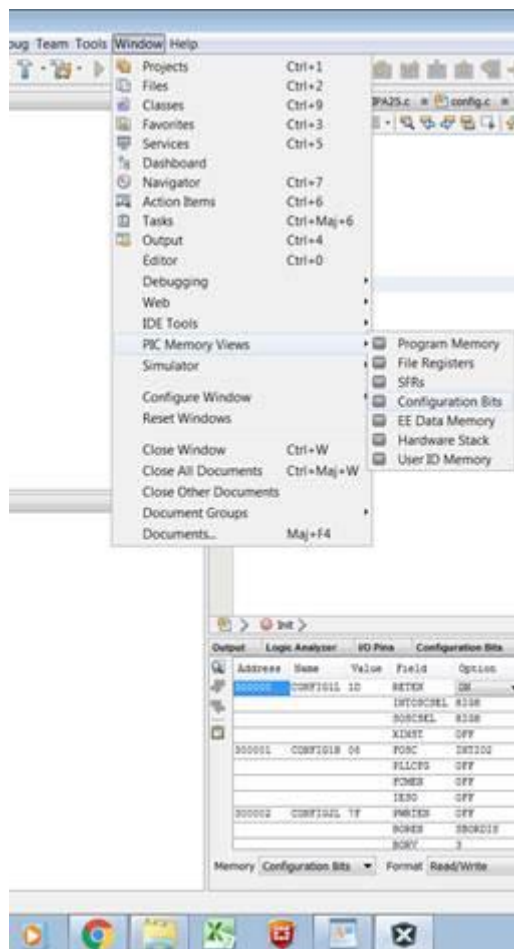
```
void T_Vitesse(void) ;// Tâche dédiée à la gestion de la vitesse
```

```
void T_Recopie(void) ;// Tâche qui gère la recopie des Switches sur  
les LED
```

```
void T_Chenillard(void) ;// Chenillard
```



Config.c représente le fichier de configuration du système



Start Page MainDemoFIPA25.c config.c FIPA25Biblio.h FIPA25Biblio.c Makefile-default.mk Makefile-impl.mk fmul.c MainDemoFIPA25V1.c

Source History

```

1 // PIC18F87K22 Configuration Bit Settings
2
3 // 'C' source line config statements
4
5 // CONFIG1L
6 #pragma config RETEN = ON // VREG Sleep Enable bit (Enabled)
7 #pragma config INTOSCSEL = HIGH // LF-INTOSC Low-power Enable bit (LF-INTOSC in High-power mode during Sleep)
8 #pragma config SOSCSEL = HIGH // SOSC Power Selection and mode Configuration bits (High Power SOSC circuit selected)
9 #pragma config XINST = OFF // Extended Instruction Set (Disabled)
10
11 // CONFIG1H
12 #pragma config FOSC = HS1 // Oscillator (HS oscillator (Medium power, 4 MHz - 16 MHz))
13 #pragma config PLLCFG = OFF // PLL x4 Enable bit (Disabled)
14 #pragma config FCMEN = OFF // Fail-Safe Clock Monitor (Disabled)
15 #pragma config IESO = OFF // Internal External Oscillator Switch Over Mode (Disabled)
16
17 // CONFIG2L
18 #pragma config PWRTEN = OFF // Power Up Timer (Disabled)
19 #pragma config BOREN = SBORDIS // Brown Out Detect (Enabled in hardware, SBORDIS disabled)
20 #pragma config BORV = 3 // Brown-out Reset Voltage bits (1.8V)
21 #pragma config BORPWR = ZPBORMV // BORMV Power level (ZPBORMV instead of BORMV is selected)
22
23 // CONFIG2H
24 #pragma config WDTCN = SWDTCN // Watchdog Timer (WDT enabled in hardware; SWDTCN bit disabled)
25 #pragma config WDTCS = 1048576 // Watchdog Postscaler (1:1048576)
26
27 // CONFIG3L
28 #pragma config RTCC = SOSCREF // RTCC Clock Select (RTCC uses SOSC)
29 #pragma config EASHT = ON // External Address Shift bit (Address Shifting enabled)
30
31

```

ABW

Output - Config Bits Source		Logic Analyzer	I/O Pins	Configuration Bits	Configuration Bits	Configuration Bits
Address	Name	Value	Field	Option	Category	Setting
300000	CONFIG1L 1D	RETE	ON		VREG Sleep Enable bit	Enabled
		INTOSCSEL	HIGH		LF-INTOSC Low-power Enable bit	LF-INTOSC in High-power mode during Sleep
		SOSCSEL	HIGH		SOSC Power Selection and mode Configuration bits	High Power SOSC circuit selected
		XINST	OFF		Extended Instruction Set	Disabled
300001	CONFIG1H 03	FOSC			Oscillator	HS oscillator (Medium power, 4 MHz - 16 MHz)
		PLLCFG	OFF		PLL x4 Enable bit	Disabled
		FCMEN	OFF		Fail-Safe Clock Monitor	Disabled
		IESO	OFF		Internal External Oscillator Switch Over Mode	Disabled
300002	CONFIG2L 7F	PWRTEN	OFF		Power Up Timer	Disabled
		BOREN	SBORDIS		Brown Out Detect	Enabled in hardware, SBORDIS disabled
		BORV	3		Brown-out Reset Voltage bits	1.8V

Memory Configuration Bits Format Read/Write Generate Source Code to Output

FIPA25Biblio.c regroupe le corps de mes fonctions. Le nom doit être identique au .h)

#include « xc.h »

#define _XTAL_FREQ 110592 // Quartz 11,0592 MHz nécessaires à
__delay_ms()

Void Init()

{

// Obligatoire : Désactivation du bus externe – Voir annexe 1

MEMCON=0xA0; //ebdis=1 bus désactivé (sauf en cas d'accès externe)





// Obligatoire : Désactivation des fonctions analogiques

ANCON0=0x00;

ANCON1=0x00;

ANCON2=0x00;

// définitions des E/S

// Initialisation des variables globales.

}

Void Tempo (unsigned char val)

{

/* Utilisation de la fonction temporisation delay_ms définie dans « xc.h »

__delay_ms(val) ;

*/

}

void T_Sens(void) // Tâche dédiée à la gestion du sens de chenillard

{

}

void T_Vitesse(void) // Tâche dédiée à la gestion de la vitesse

{

}





```
void T_Recopie(void)// Tâche qui gère la recopie des Switches sur les LED
```

```
{  
}
```

```
void T_Chenillard(void)// Chenillard
```

```
{  
}
```

MainDemoFIPA25.c

C'est la fonction Main qui comportera l'ordonnanceur séquentiel mono tâche.

```
#include <xc.h>
```

```
#include "FIPA25Biblio.h"
```

```
void main(void) {  
    Init();  
    while (1)  
    {  
        T_Sens() ;  
        T_Vitesse() ;  
        T_Recopie() ;  
        T_Chenillard() ;  
    }  
    return;  
}
```



Annexe :

1) Gestion de la mémoire externe :

PIC18F87K22 FAMILY

8.1 External Memory Bus Control

The operation of the interface is controlled by the MEMCON register (Register 8-1). This register is available in all program memory operating modes except Microcontroller mode. In this mode, the register is disabled and cannot be written to.

The EBDIS bit (MEMCON<7>) controls the operation of the bus and related port functions. Clearing EBDIS enables the interface and disables the I/O functions of the ports, as well as any other functions multiplexed to those pins. Setting the bit enables the I/O ports and other functions, but allows the interface to override everything else on the pins when an external memory operation is required. By default, the external bus is always enabled and disables all other I/O.

The operation of the EBDIS bit is also influenced by the program memory mode being used. This is discussed in more detail in Section 8.5 "Program Memory Modes and the External Memory Bus".

The WAIT bits allow for the addition of Wait states to external memory operations. The use of these bits is discussed in Section 8.3 "Wait States".

The WM bits select the particular operating mode used when the bus is operating in 16-Bit Data Width mode. These bits are discussed in more detail in Section 8.6 "16-Bit Data Width Modes". These bits have no effect when an 8-Bit Data Width mode is selected.

REGISTER 8-1: MEMCON: EXTERNAL MEMORY BUS CONTROL REGISTER⁽¹⁾

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7 EBDIS: External Bus Disable bit

1 = External bus is enabled when microcontroller accesses external memory; otherwise, all external bus drivers are mapped as I/O ports

0 = External bus is always enabled, I/O ports are disabled

Dans le cas du TP1 : MEMCON.EBDIS=1 ;

A partir du TP2 : MEMCON.EBDIS=1 ;

2) Le registre CONFIG1L

REGISTER 28-2: CONFIG1H: CONFIGURATION REGISTER 1 HIGH (BYTE ADDRESS 300001h)

R/P-0	R/P-0	U-0	U-0	R/P-1	R/P-0	R/P-0	R/P-0
IESO	FCMEN	—	PLLCFG ⁽¹⁾	FOSC3 ⁽²⁾	FOSC2 ⁽²⁾	FOSC1 ⁽²⁾	FOSC0 ⁽²⁾
bit 7							bit 0

Legend:	P = Programmable bit
R = Readable bit	W = Writable bit
-n = Value at POR	'1' = Bit is set
	U = Unimplemented bit, read as '0'
	'0' = Bit is cleared
	x = Bit is unknown

- bit 7 **IESO**: Internal/External Oscillator Switchover bit
1 = Two-Speed Start-up is enabled
0 = Two-Speed Start-up is disabled
- bit 6 **FCMEN**: Fail-Safe Clock Monitor Enable bit
1 = Fail-Safe Clock Monitor is enabled
0 = Fail-Safe Clock Monitor is disabled
- bit 5 **Unimplemented**: Read as '0'
- bit 4 **PLLCFG**: 4x PLL Enable bit⁽¹⁾
1 = Oscillator is multiplied by 4
0 = Oscillator is used directly
- bit 3-0 **FOSC<3:0>**: Oscillator Selection bits⁽²⁾
1101 = EC1, EC oscillator (low power, DC-160 kHz)
1100 = EC1IO, EC oscillator with CLKOUT function on RA6 (low power, DC-160 kHz)
1011 = EC2, EC oscillator (medium power, 160 kHz-16 MHz)
1010 = EC2IO, EC oscillator with CLKOUT function on RA6 (medium power, DC-160 kHz-16 MHz)
1001 = INTIO1, internal RC oscillator with CLKOUT function on RA6
1000 = INTIO2, internal RC oscillator
0111 = RC, external RC oscillator
0110 = RCIO, external RC oscillator with CLKOUT function on RA6
0101 = EC3, EC oscillator (high power, 4 MHz-64 MHz)
0100 = EC3IO, EC oscillator with CLKOUT function on RA6 (high power, 4 MHz-64 MHz)
0011 = HS1, HS oscillator (medium power, 4 MHz-16 MHz)
0010 = HS2, HS oscillator (high power, 16 MHz-25 MHz)
0001 = XT oscillator
0000 = LP oscillator

Note 1: Not valid for the INTIOx PLL mode.

Note 2: INTIO + PLL can be enabled only by the PLEN bit (OSCTUNE<6>). Other PLL modes can be enabled by either the PLEN bit or the PLLCFG (CONFIG1H<4>) bit.

Dans notre cas Quartz externe 11,0592 MHZ d'où une configuration HS1.