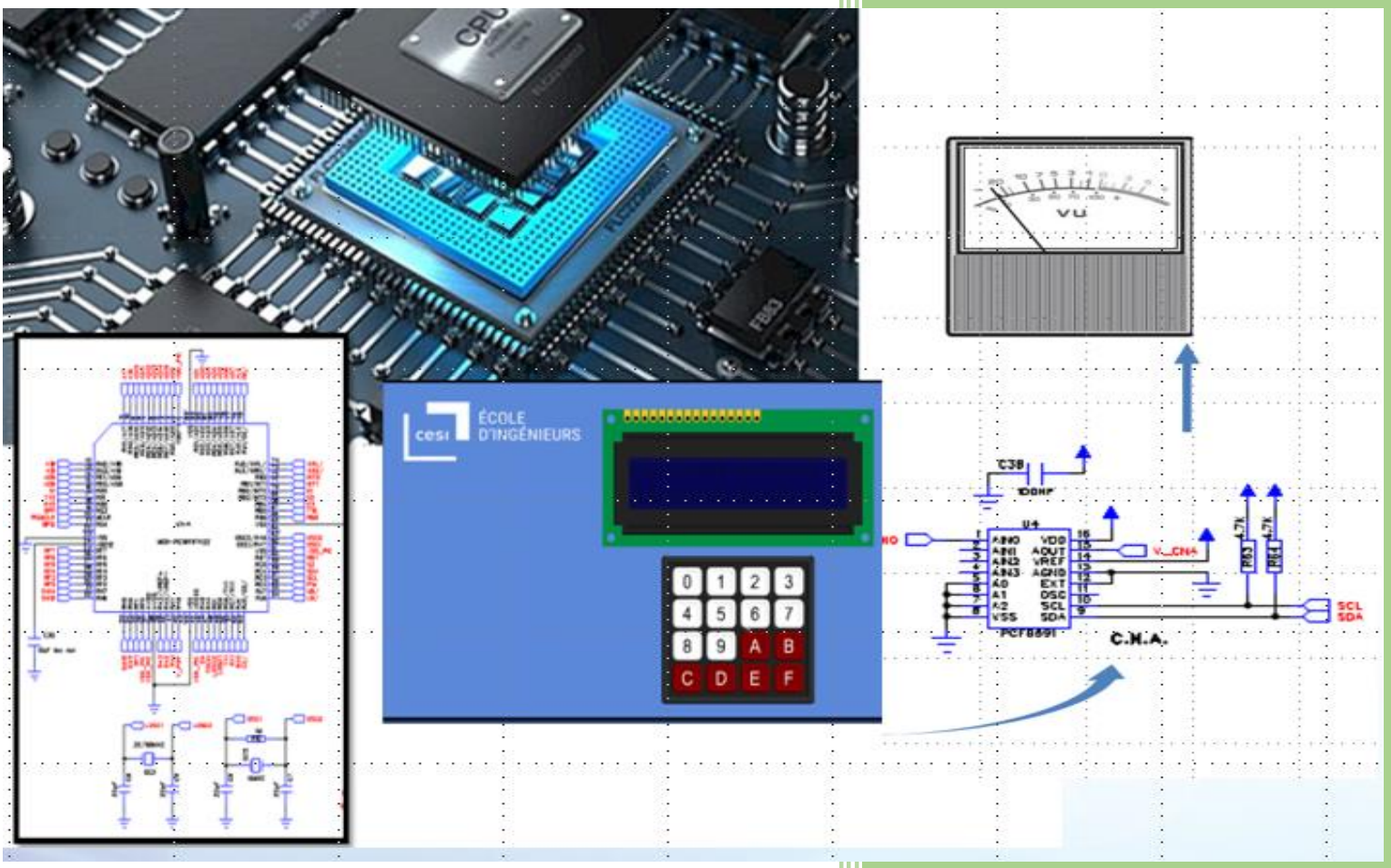
2022

TP3 FISA 26 Microcontrôleur



ANTINI Nicolas

CESI

01/01/2022

TP N°3 MICROCONTROLEUR PIC 18F87K22

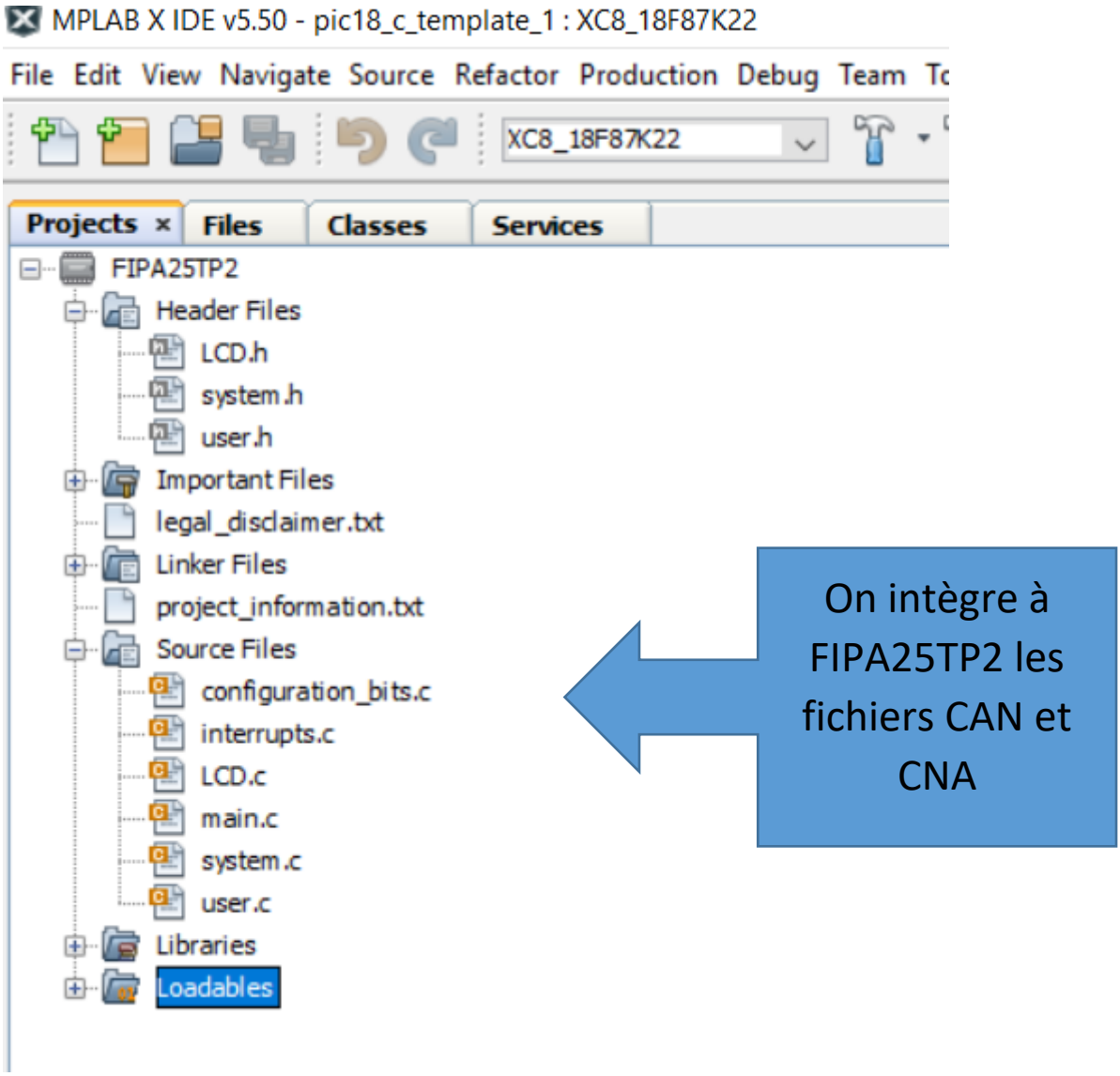
- **Mise en œuvre des convertisseurs Analogiques Digitaux / Digitaux analogiques.**
- **Objectifs du TP :**
- Mise en œuvre des conversions
- Affichage de la valeur
- Bus I2C

- **Expérimentation n°1:**
- Lire la valeur de tension présente sur le potentiomètre.
- Afficher la valeur sur l'afficheur 2 lignes 16 caractères.

- **Expérimentation n°2:**
- Déterminer l'adresse I2C du CNA PCF8591.
- Etablir la trame de communication I2C.
- La valeur de tension relevé sur le potentiomètre (12 bits) est reportée sur le CNA (8 bits).

Eclairer le voyant du CNA a l'atteinte du seuil 70 %. (S1 câblée sur RG0)

Etape 2) Créer le projet FIPA25TP3.



user.h regroupe les entêtes de mes fonctions.

Ex : void Init(void) ;

system.h regroupe les entêtes des fonctions spécifiques au système.

LCD .h regroupe les entêtes des fonctions spécifiques a la gestion de l'afficheur LCD.

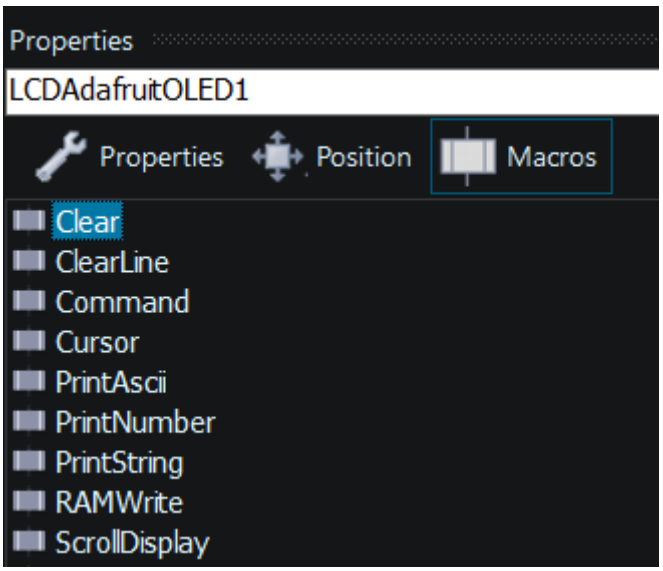
Source Files :

Configuration_bits.c représente le fichier de configuration du système

Interrupt.c regroupes les fonctions liées à la gestion des ISR

LCD.c fonctions liées à la gestion de l'afficheur LCD

Equivalence LCDAdafruitOLED1



CAN.c Fonctions liées à la gestion du Convertisseur Analogique Numérique

CNA.C : Fonctions liées à la gestion du Convertisseur Analogique Numérique

MicroC

10 .1 Registres du Convertisseur

REGISTER 23-8: ANCON0: A/D PORT CONFIGURATION REGISTER 0

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSEL7	ANSEL6	ANSEL5	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
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 7-0 **ANSEL<7:0>**: Analog Port Configuration bits (AN7 and AN0)
1 = Pin is configured as an analog channel; digital input is disabled and any inputs read as '0'
0 = Pin is configured as a digital port

ANCON0 ,1,2 permet de configurer la ligne en Entrée/Sortie logique ou en Entrée ANALOGIQUE
exemple : ANCON0bits.ANSEL4=1, l'entrée AN4 est une entrée analogique.

MicroC

REGISTER 23-1: ADCON0: A/D CONTROL REGISTER 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

ADCCON0 permet : la sélection du canal de conversion (32 canaux), alimenter le convertisseur (ADON), démarrer la conversion, scruter la fin de la conversion (GO/DONE).

REGISTER 23-2: ADCON1: A/D CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TRIGSEL1	TRIGSEL0	VCFG1	VCFG0	VNCFG	CHSN2	CHSN1	CHSN0
bit 7							bit 0

ADCCON1 permet : conditions de lancement de la conversion, sélection des tensions de références, les canaux négatifs.

REGISTER 23-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

ADCCON2 permet : Configuration de la fréquence de conversion, délai précédent la conversion.

ÉCOLE D'INGÉNIEURS CESI

CESI ei.cesi

CESI TOULOUSE

REGISTER 23-4: ADRESH: A/D RESULT HIGH BYTE REGISTER, LEFT JUSTIFIED (ADFM = 0)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES11	ADRES10	ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4
bit 7							bit 0

REGISTER 23-5: ADRESL: A/D RESULT HIGH BYTE REGISTER, LEFT JUSTIFIED (ADFM = 0)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES3	ADRES2	ADRES1	ADRES0	ADSGN	ADSGN	ADSGN	ADSGN
bit 7							bit 0

ADRES H et L contiennent la valeur convertie sur 12 bits et son signe.

ÉCOLE D'INGÉNIEURS CESI

CESI el.cesi

Annexes :

1) Afficheur 2*16 :

Séquence d'init recommandée :

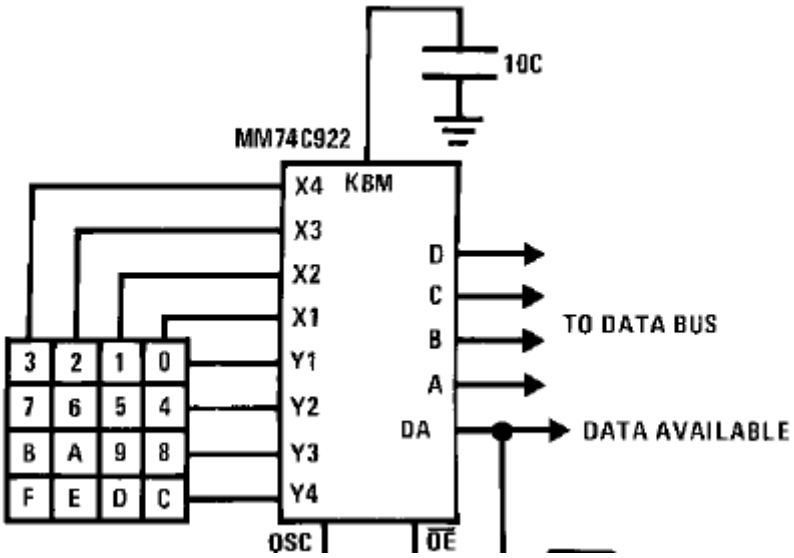
```
//Définitions des fonctions LCD
#define DISPLAY_CLEAR            0x01
#define RETURN_HOME              0x02
#define ENTRY_MODE_SET_CI_DNS    0x06 //Cursor increase, is not shifted
#define ENTRY_MODE_SET_CI_DS     0x07 //Cursor increase, shifted
#define DISPLAY_ON_CUR_ON_BLINK_ON  0x0F
#define DISPLAY_ON_CUR_ON_BLINKOFF  0x0E
#define SHIFT_DISPLAY_RIGHT      0x1C
#define SHIFT_DISPLAY_LEFT       0x18
#define SHIFT_CURSOR_RIGHT       0x14
#define SHIFT_CURSOR_LEFT        0x10
#define SET_FUNC_8BIT_2LINE_5x10   0x3C
#define SET_FUNC_8BIT_2LINE_5x7    0x38

void init_aff_lcd(void)
{
   delai_ms(100);
   LCD_FUNC=0x38;                  delai_ms(5);
   LCD_FUNC=0x38;                  delai_ms(1);
   LCD_FUNC=0x38;                  delai_ms(1);
   LCD_FUNC=SET_FUNC_8BIT_2LINE_5x7;     delai_ms(1);
   LCD_FUNC=SHIFT_CURSOR_RIGHT;          delai_ms(1);
   LCD_FUNC=DISPLAY_ON_CUR_ON_BLINKOFF;   delai_ms(1);
   LCD_FUNC=ENTRY_MODE_SET_CI_DNS;       delai_ms(1);
   LCD_FUNC=RETURN_HOME;              delai_ms(2);
   LCD_FUNC=DISPLAY_CLEAR;            delai_ms(2);
}
```

// delai_ms () est une fonction de temporisation qui utilise **__delay_ms() voir TP1.**

3) Niveau d'IT

Selon exemple Ci-dessous on imagine un bouton poussoir ou la ligne DA reliée à INT0 :



1 seul niveau d'IT

1- Un seul niveau d'IT
RCONbits.IPEN=0;
2- validation IT globales:
INTCONbits.GIE=1
3- validation IT périphériques
INTCONbits.PEIE= 1 ou 0
4- validation individuelle des IT
Validation du périphérique xxx: **xxxxIE=1**
5- acquittement des IT
Si IT en suspens l'acquitter : **xxxIF=0**

```
// exemple bouton poussoir sur RBO en IT
RCONbits.IPEN=0; // 1 seul nv
INTCONbits.GIE=1; // IT autorisé
INTCONbits.PEIE=0; // pas de périph en IT
INTCONbits.INT0IE=1; // IT bouton poussoir autorisé
INTCON2bits.INTEDG0=0; // choix du front
INTCONbits.INT0IF=0; // acquittement init
```

2 niveaux d'IT

1- Deux seul niveau d'IT
RCONbits.IPEN=1;
2- validation IT hautes:
INTCONbits.GIEH=1 ***(ou INTCONbits.GIE=1)***
3- validation IT périphériques
INTCONbits.GIEL=1 ***(ou INTCONbits.PEIE= 1)***
4 – Pour chaque source d'IT choisir si priorité haute ou basse
xxxxIP=……; //1= haute priorité 0=basse

5- validation individuelle des IOT
Validation du périphérique xxx: **xxxxIE=1**
6- acquittement des IT
Si IT en suspens l'acquitter : **xxxIF=0**