

République Algérienne Démocratique et Populaire
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEURE ET DE RECHERCHE SCIENTIFIQUE



Université Ibn-Khaldoun de Tiaret
Faculté des Sciences Appliquées
Département de Génie Électrique

SYSTÈMES EMBARQUÉS

RAPPORT DE FIN D'ÉTUDE DU MODULE

Écrit par l'étudiant : Laïd AKERMI
Master2 en Électronique des Systèmes Embarqués

Année Universitaire 2022-2023

Table des matières

1	Introduction	3
2	Présentation de la formation	4
2.1	Durée	4
2.2	Description des missions	4
2.3	Le rapport	4
3	Missions à accomplir	5
3.1	Qu'est-ce qu'un système embarqué?	5
3.2	Tester un PIC	6
3.3	De l'hexadécimal vers l'assembleur	6
3.4	Optimiser un code Assembleur	8
3.5	Un petit projet	8
4	Conclusion	12

1 Introduction

Un système embarqué est défini comme un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche précise. Le terme désigne aussi bien le matériel informatique que le logiciel utilisé.

Ses ressources sont généralement limitées spatialement (encombrement réduit) et énergétiquement (consommation restreinte).

Les systèmes embarqués utilisent généralement des microprocesseurs à basse consommation d'énergie ou des microcontrôleurs, dont la partie logicielle est en partie ou entièrement programmée dans le matériel, généralement en mémoire dans une mémoire morte (ROM), EPROM, EEPROM, FLASH, etc. (on parle alors de firmware).

source : [wikipedia.org](https://fr.wikipedia.org/wiki/Syst%C3%A8me_embarr%C3%A9)

2 Présentation de la formation

2.1 Durée

Du 11 Septembre 2012 au 15 Janvier 2023 (quatre (4) mois), nous avons effectué une formation au sein de l'université Ibn Khaldoune-Tiaret.

Au cours de cette formation dans le département de génie électrique, nous avons suivis des cours, TDs et TP des modules l'un d'entre eux (fondamental) est l'Électronique des Systèmes Embarqués.

2.2 Description des missions

Le module est constitué essentiellement de :

1. Qu'est-ce qu'un système embarqué ?
2. Tester les ports d'un microcontrôleur.
3. Du ".HEX" vers ".ASM".
4. Optimiser un code assembleur.
5. Réaliser un petit projet.

2.3 Le rapport

Le présent rapport est rédigé à l'aide du logiciel **L^AT_EX**.

Tout les schémas **Proteus** et les codes **XC8** que j'ai réalisé sont disponibles dans ce lien : <https://github.com/AKERMILAI77/Embedded-System>

3 Missions à accomplir

3.1 Qu'est-ce qu'un système embarqué?

3.1.1 Caractéristiques principales d'un système embarqué

- C'est un système électronique et informatique autonome, qui est dédié à une tâche bien précise.
- Il ne possède pas des entrées/sorties standards et classiques comme un clavier ou un écran d'ordinateur.
- L'affichage est limité (écran LCD...) ou n'existe pas du tout.
- Le système matériel et l'application sont intimement liés, le logiciel embarqué est noyé dans le matériel.
- Il met en œuvre généralement un processeur.

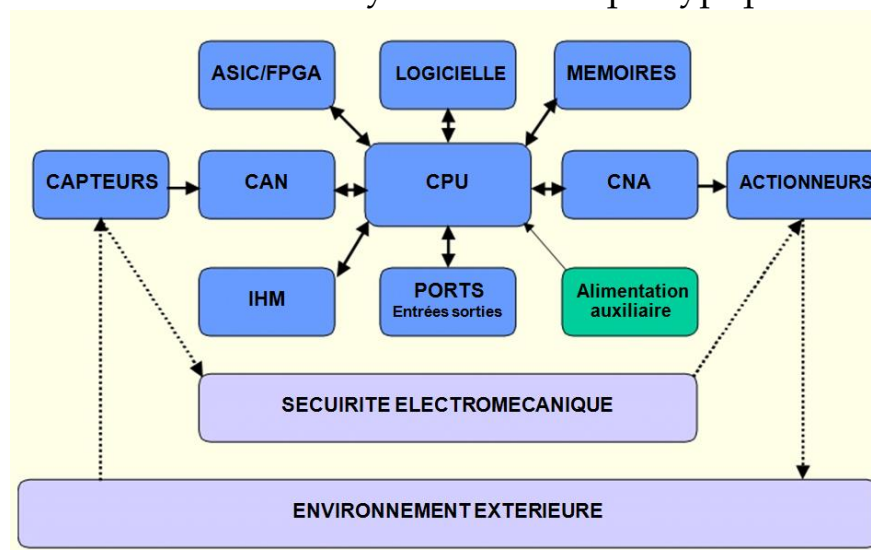
3.1.2 Les grands secteurs de l'embarqué

- Jeux et calcul général : application similaire à une application de bureau mais empaquetée dans un système embarqué.
- Contrôle de systèmes : automobile, process chimique, process nucléaire, système de navigation...
- Traitement du signal : radar, sonar ...
- Communication et réseaux : téléphonie, internet...

3.1.3 Un système embarqué typique

- En entrée des capteurs généralement analogiques couplés à des CAN.
- En sortie des actionneurs généralement analogiques couplés à des CNA.
- Au milieu, le(s) calculateur(s).

FIGURE 1 – Un système embarqué typique



3.1.4 Le rôle de l'environnement extérieur

Contrairement au PC, un système embarqué doit faire face à des environnements plus hostiles (paramètres agressifs) :

- Variations de la température.
- Vibrations, chocs.
- Variations des alimentations.
- Interférences RF.
- ...

3.2 Tester les entrées et les sorties d'un PIC 16F877

Cette mission a le rôle de savoir l'état interne de circuit (microcontrôleur)¹.

Remarque : Nous avons étudié et programmé le PIC 16F877 comme exemple, pour utiliser un autre circuit programmable il suffit de voir son **DataSheet**.

3.2.1 Caractéristiques électriques d'un PIC 16F877

- Voltage dans chaque pin (sauf VDD, MCLR et RA4) : de (-0.3V) à (VDD + 0.3V).
- Voltage (VDD - VSS) : de (-0.3) à (+7.5V).
- Voltage (MCLR - VSS) : de (0) à (+14V).
- Voltage (RA4 - Vss) : de (0) to (+8.5V).
- Courant maximal sortie de VSS :300mA.
- Courant maximal entré dans VDD :250mA.
- Courant maximal (entré ou sortie) de : PORTA, PORTB, PORTC, PORTD et PORTE : 200mA.
- Courant maximal (entré ou sortie) de chaque pin : 25mA.

3.2.2 Schéma et code

Le PIC16F877 **ne supporte pas un courant élevé**, donc nous avons testé uniquement un demi-port à la fois.

Proteus :testBroches.pdsprj

XC8 :testBroches.c

3.3 De l'hexadécimal vers l'assembleur

La famille Mid-Range "PIC16", utilise des mots de 14 bits.

1. Quels sont les pins fonctionnels?

FIGURE 2 – testBroches.hex

```
:060000000A128A11FC2F18
:100F7A000630831603139F0055308312031385002E
:100F8A0083160313850155308312031386008316D3
:100F9A000313860155308312031387008316031344
:100FAA0087015530831203138800831603138801BF
:100FBA0055308312031389008316031389018312A0
:100FCA000313850986098709880989090D30F20002
:100FDA00AF30F100B630F000F00BF12FF10BF12F2A
:100FEA00F20BF12F0000E42F0A128A110028830164
:060FFA000A128A11BD2F4E
:08400000FF3FFF3FFF3FFF3FC0
:02400E003F3D34
:00000001FF
```

0831 → 0183

FIGURE 3 – INSTRUCTION SET-PIC16F87XA DATASHEET

0183 = 0001 0001 1000 0011

Hex

Binair

TABLE 15-2: PIC16F87XA INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode		Status Affected	Notes
				MSb	LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS							
ADDWF	f, d	Add W and f	1	00	0111 dfff ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101 dfff ffff	Z	1,2
CLRF	f	Clear f	1	00	0001 1fff ffff	Z	2
CLRW	-	Clear W	1	00	0001 0xxx xxxx	Z	
COMF	f, d	Complement f	1	00	1001 dfff ffff	Z	1,2
DECf	f, d	Decrement f	1	00	0011 dfff ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011 dfff ffff		1,2,3

Ce que donne CLRF 3 → CLRF STATUS;

FIGURE 4 – REGISTER FILE MAP -PIC16F87XA DATASHEET

Indirect addr.(*)	00h	Indirect addr.(*)	80h
TMR0	01h	OPTION_REG	81h
PCI	02h	PCL	82h
STATUS	03h	STATUS	83h
FSR	04h	FSR	84h
PORTA	05h	TRISA	85h

3.4 Optimiser un code Assembleur

Trois niveaux d'optimisation :

1. Trouver puis supprimer les instructions "sans effet"² sur le déroulement de programme.
2. Trouver l'emplacement idéal pour chaque instruction pour éliminer autres instruction "sans effet".
3. Calculer le temps d'exécution des instructions pour mieux gérer les délais.

3.4.1 Code assembleur avant l'optimisation

XC8 :HexAsm.txt

3.4.2 Code assembleur après l'optimisation

XC8 :asmModified.txt

3.5 Un petit projet "Régler la température"

3.5.1 Schéma et code

Dans ce petit projet, nous essayons de régler la température d'une chambre (par exemple) à 28°, en utilisant une ventilateur montée sur une moteur, ce dernier est commandé par un microcontrôleur PIC16F877.

On remarque qu'il est difficile de calculer le coefficients de P.I.D. Pour cette simulation, on prend par expérience $f_P = \frac{1}{4}$, $f_I = \frac{1}{2}$, $f_D = \frac{1}{2}$.

On essaie d'éviter la multiplication et la division (classiques) qui rend notre programme lourd³.

```
// shift remplace la division  
dc = (vPr<<2) + (vDi<<1) + (vIn<<2);
```

Proteus :DACADC_4.pdsprj

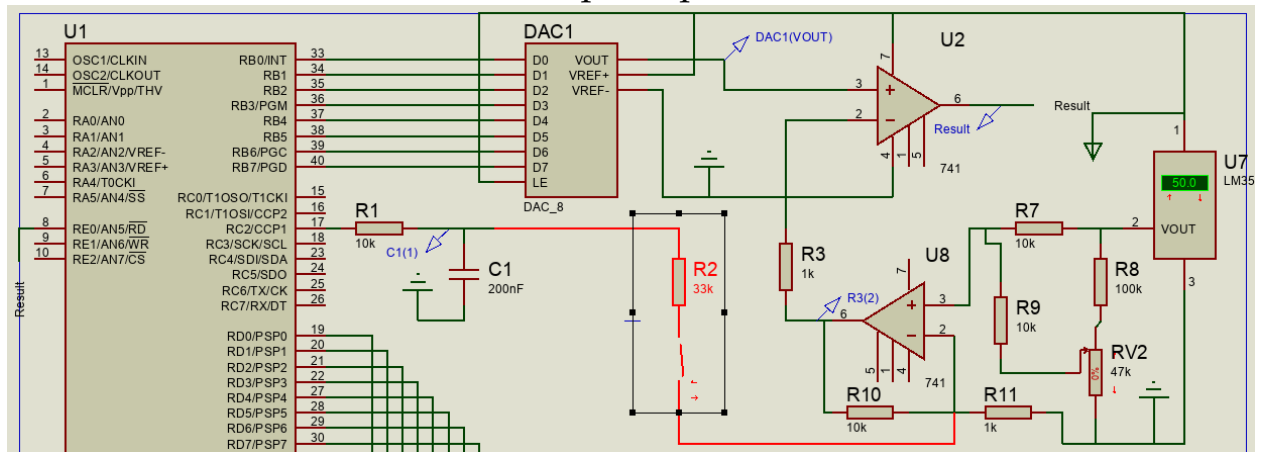
XC8 :embSysDACADC_4.c, embSysDACADC_4.h

2. Valeur déjà affectée par une instruction précédente ou elle présente par défaut.

3. On n'oublie pas qu'on travaille sur un Système Embarqué.

3.5.2 Simuler la régulation de la température!

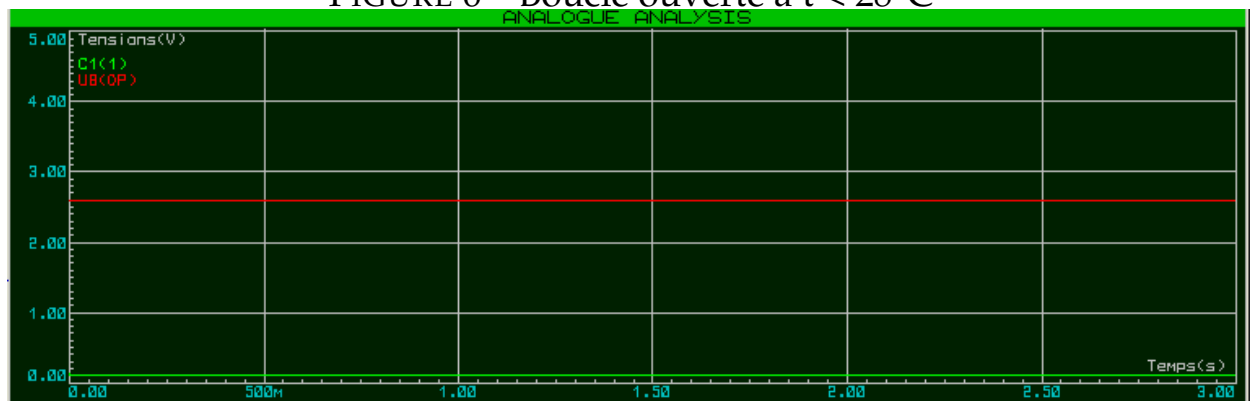
FIGURE 5 – Résistance et interrupteur pour simuler une boucle fermée



3.5.3 Quelques résultats de simulation pendant (3s)

- La ligne rouge représente la tension fournie par le capteur de température "LM35",
- La ligne verte représente la tension fournie par le microcontrôleur PIC16F877 pour alimenter le moteur à travers d'un circuit driver⁴.

FIGURE 6 – Boucle ouverte à $t < 28^{\circ}\text{C}$



4. Un microcontrôleur ne peut pas alimenter directement un moteur.

FIGURE 7 – Boucle ouverte à $t = 30^{\circ}\text{C}$

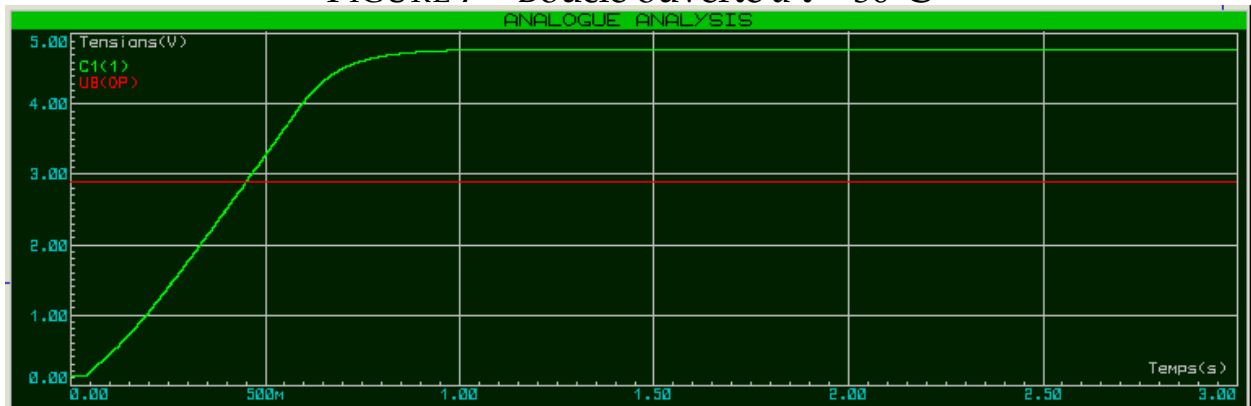


FIGURE 8 – Boucle ouverte à $t = 40^{\circ}\text{C}$ et $t = 50^{\circ}\text{C}$

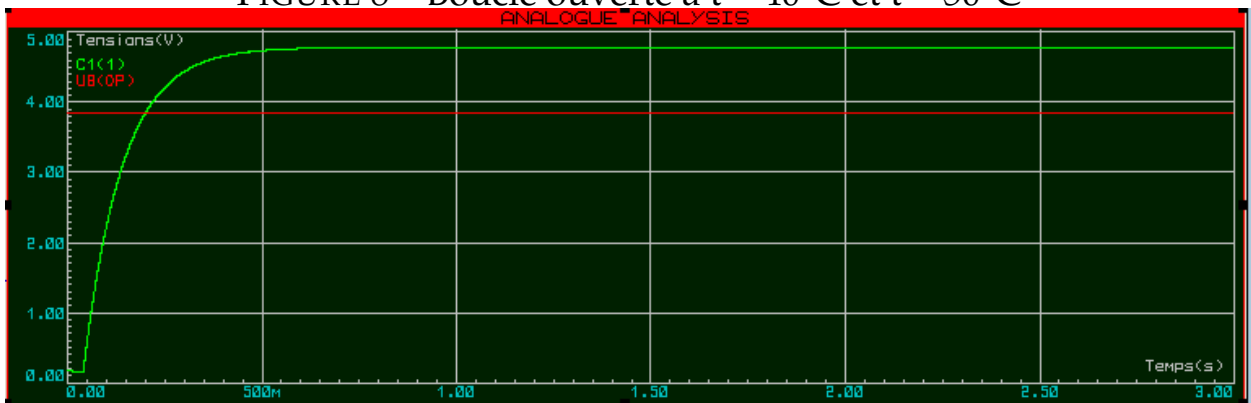


FIGURE 9 – Boucle fermée à $t < 28^{\circ}\text{C}$



FIGURE 10 – Boucle fermée à $t = 30^{\circ}\text{C}$

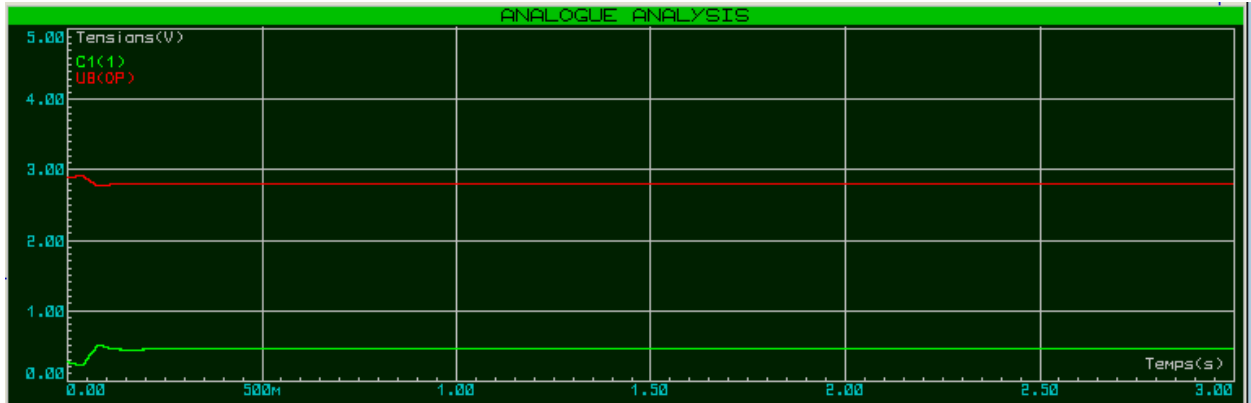


FIGURE 11 – Boucle fermée à $t = 40^{\circ}\text{C}$

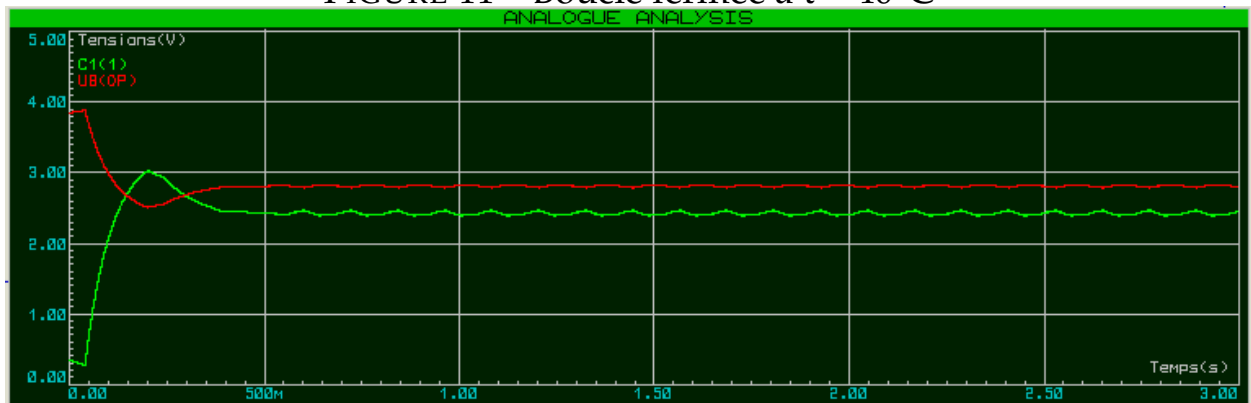
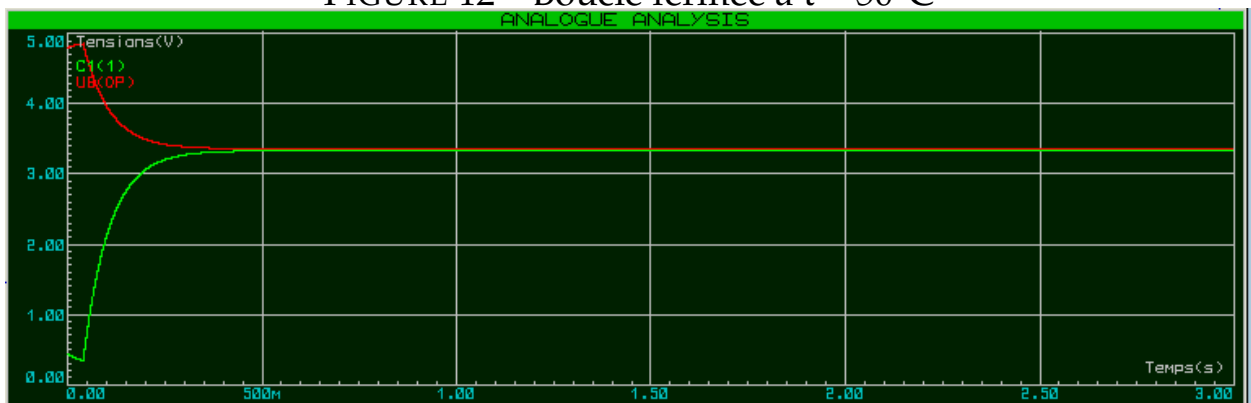


FIGURE 12 – Boucle fermée à $t = 50^{\circ}\text{C}$



4 Conclusion

Le suivi d'une formation "Électronique des Systèmes Embarqués" nécessite un temps important, une grande volonté et beaucoup de patience.

Cette formation a donc été une opportunité pour moi de percevoir l'intérêt d'un système embarqué.

Je tiens à remercier M. Mustafa BELARBI, responsable de notre spécialité. En tant que maître de module (Systèmes Embarqués), il nous a beaucoup appris et a partagé ses connaissances dans le domaine de l'Électronique des Systèmes Embarqués.