

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la A Recherche Scientifique

Université Mouloud Mammeri De Tizi-Ouzou



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

**Mémoire de Fin d'Etudes
de MASTER ACADEMIQUE**

Domaine : Science et technologie

Filière : Génie Electronique

Spécialité : **INSTRUMENTATION**

Présenté par
Garti Chahrazad
Hales Dyhia

Thème

**Etude de l'implémentation de ARM Cortex M3
dans un système embarqué**

Proposé et dirigé par :

Mr M.LAZRI

Année universitaire: 2017/2018

Remerciements

Nous remercions tout d'abord « LE BON DIEU » tout puissant qui nous a donné le courage, la patience et la volonté.

Nous remercions notre promoteur Mr LAZRI.M, pour ses précieuses orientations, son aide et ses conseils tout au long de ce projet.

Nous remercions également le président de jurys ainsi que les membres de jurys d'avoir accepté d'évaluer et juger notre travail.

Nos remerciements les plus chaleureux vont à nos chères FAMILLES pour leurs encouragements, leur patience et leur grand soutien durant toutes ces années d'études.

Enfin, un remerciement tout particulier à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicaces

Je dédie ce travail :

*En premier lieu à la mémoire de mon père et mon beau
père Mohamed que dieu le recueillera dans son vaste
paradis.*

*A ma très chère mère qui m'a soutenue tout au long de
ma vie en témoignage de son affection et sacrifices.*

*A mes très chers frères et sœurs spécialement « Hanane
et Esaid»*

A toute ma belle famille ;

A mon fiancé « kamel » qui a toujours été là pour moi ;

A toutes les familles Tafat et Garti ;

A ma binôme «Dyhia» ainsi qu'à toute sa famille ;

A tous ceux que j'aime ;

A tous ceux qui m'aiment ;

A tous mes ami(e)s et mes collègues

« Chahrazad »

Dédicaces

Je dédie ce travail :

*A mes chers parents qui m'ont toujours soutenu tout au
long de mes études ;*

A ma sœur et mon frère « Lydia et Ghilas » ;

*A mon adorable grand père que le DIEU le garde pour
nous ;*

A mes chères cousines « Kahina et Lila » ;

A ma binôme « chahra » ainsi qu'à toute sa famille ;

A tous ceux que j'aime ;

A tous ceux qui m'aiment ;

A tous mes ami(e)s ;

« Dyhia »

Remerciements	
Dédicaces	
Liste des abréviations	
Introduction.....	1

CHAPITRE I : L'électronique numérique

I.1.Préambule.....	4
I .2. L'électronique numérique.....	4
I.2.1. Logique combinatoire.....	4
I.2.1.1. Opération fondamentaux.....	4
I.2.1.2. Expression d'une fonction logique.....	6
I.2.1.3. Simplification des fonctions logiques.....	6
I.2.1.4. Représentation des nombres.....	7
I.2.1.5. Circuits logique combinatoires	9
I.2.2. logique séquentielle.....	11
I.2.2.1 Les bascules.....	12
I.2.2.3 Les registre	15
I.2.2.4. Les capteur	16
I.3. Discussion.....	17

CHAPITRE II : Les systèmes embarqués

II.1 Préambule.....	19
II.2 Définition des systèmes embarqués.....	19
II.2.1 Caractéristiques des systèmes embarqués.....	19
II.2.2 Contraintes de conception des systèmes embarqués.....	20
II.2.3 Domaines d'applications des systèmes embarqués.....	21
II.2.4 Systèmes embarqués spécifiques et les systèmes monpuces.....	22

II.2.5 Architecture des systèmes embarqués.....	23
II.2.5.1 Systèmes embarqués de première génération.....	23
II.2.5.2 Systèmes embarqués de deuxième génération.....	24
II.2.5.3 Systèmes embarqués de troisième génération.....	26
II.2.6 Avantages des systèmes embarqués.....	28
II.2.7 Systèmes embarqués temps réel.....	29
II.2.7.1 Caractéristiques	30
II.2.8 Types de temps réel.....	30
II.3 Discussion.....	30

CHAPITRE III : Les systèmes à microprocesseur

III.1 Préambule.....	32
III.2 Systèmes à microprocesseur.....	32
III.3 Microprocesseur.....	33
III.4 Mémoires	35
III.4.1 Caractéristiques d'une mémoire.....	36
III.4.2 Différents types des mémoires.....	36
III.5 Mémoire cache.....	39
III.6 Interfaces d'entrée/sortie.....	40
III.6.1 Liaison parallèle	40
III.6.2 Liaison série.....	40
III.7 Le microcontrôleur	41
III.7.1 ARM Cortex-M3.....	42
III.8 Discussion.....	44

CHAPITRE IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

IV.1 Préambule	46
IV.2 Présentation du schéma synoptique de la station météorologique.....	46

IV.3 Présentation des éléments de la station.....	47
IV.4 Fonctionnement de la station.....	55
IV.5 Discussion.....	56

Conclusion

Bibliographie

Introduction :

Introduction :

Un système embarqué est un système électronique et informatique qui se trouve au sein d'un produit plus large. Son rôle principal est d'exécuter une tâche spécifique bien précise reçue depuis son environnement.

On retrouve des systèmes embarqués dans de nombreux appareils, tels que les appareils électroménagers, les machines à laver, fours à micro-ondes... dans ces appareils, on retrouve typiquement un seul système embarqué, peu évolué, dont le but est de gérer leurs différents états de fonctionnement. Les systèmes se présentent également dans tout ce qui est électronique grand public tels que les appareils photos, les caméras numériques, consoles de jeu, téléviseurs.... Il y en a également dans les systèmes d'information et de communication, comme les téléphones portables et Smartphones. Ils sont aussi présents dans d'autres dispositifs de communications tels que les modems, fax, les antennes, les satellites.....

Les systèmes embarqués ne sont pas seulement présents chez le grand public, mais sont largement utilisés dans le monde industriel. On les retrouve aussi dans les chaînes de production, dans des stations de contrôles (centrale nucléaire, activité volcanique, balises d'alertes tsunami...). Il s'agit souvent de systèmes embarqués devant être beaucoup plus robuste afin de supporter des conditions bien plus strictes que celles des applications destinées au grand public. On les trouve aussi dans les moyens de transport tels que les avions, voiture moderne. Sans être exhaustif, on en retrouve pour contrôler la boîte de vitesse, le moteur, la climatisation, les systèmes de divertissement

Pour mettre au point un système embarqué, plusieurs travaux ont été réalisés. Dans ce contexte, nous avons réalisé une application en utilisant le microprocesseur ARM. Ce dernier est très utilisé dans les systèmes embarqués. Notre travail consiste donc à l'étude matérielle de l'implémentation du Cortex M-3 dans une station météorologique.

Notre station est équipée d'un capteur de pression, capteur d'humidité, un capteur de lumière, et un convertisseur RS232- Bluetooth Free2move.

Pour bien mener notre travail, nous avons organisé notre mémoire en quatre chapitres :

Le premier chapitre est consacré à la présentation de l'électronique numérique et ses notions de base.

Dans le deuxième chapitre, nous nous sommes intéressés à étudier le système embarqué, ses caractéristiques, ses avantages et inconvénients, ainsi que son architecture.

Dans le troisième chapitre, nous avons défini le système à microprocesseur, en décrivant son architecture de base et les principes de son fonctionnement, ensuite nous avons détaillé le microcontrôleur STM32, qui est le processeur utilisé dans notre application,

Pour finir, le quatrième et dernier chapitre est consacré à la présentation de l'application que nous avons étudiée qui consiste en l'implémentation du microprocesseur Cortex-M3 qui est

de type ARM et qui se trouve dans le microcontrôleur de la famille STM32, dans une station météorologique.

Nous terminons notre travail par une conclusion.

Chapitre I :
L'électronique numérique

I.1. Préambule

Dans ce chapitre, nous décrivons les notions de base sur l'électronique numérique, dans cette broche de l'électronique, les signaux électriques ne peuvent prendre que deux états 0 ou 1. On distingue deux types de logique selon que le temps qui s'écoule est pris en compte ou non. Dans la logique combinatoire, une fonction sera exécutée chaque fois qu'un ensemble de conditions sera réuni, et maintenue tant que ces conditions demeurent. A l'opposé, la logique séquentielle fait intervenir le temps, ou en d'autres termes la mémorisation.

I.2. Electronique numérique

Elle regroupe les systèmes électroniques fonctionnant sur la base d'états électriques précis, dont le nombre et les valeurs sont fixés à leur conception, à chaque état correspond une valeur numérique.

Pour bien comprendre l'électronique numérique, nous allons nous intéresser à étudier les deux logiques combinatoire et séquentielle.

I.2.1. Logique combinatoire

La logique combinatoire est une logique de combinaison de variables, c'est-à-dire que pour une combinaison d'entrée donnée, il ne correspond qu'une et une seule combinaison de sortie.[3]

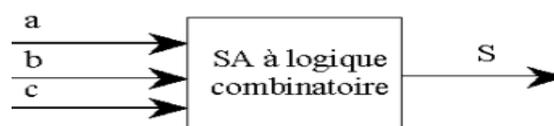


Figure I.1 : Schéma de la logique combinatoire

I.2.1.1. Opérateurs fondamentaux

L'une des caractéristiques de l'électronique numérique est sa simplicité, il n'existe en effet que trois circuits : le ET, le OU, et le NON, les autres circuits étant des variantes de ces trois :

v

- **INV (NON)** : L'opérateur d'inversion ne porte que sur une seule variable d'entrée. Si A est la variable d'entrée, S la variable de sortie vaut : $S = \bar{A}$.

- **AND (ET) :** L'opérateur AND (ET) porte sur deux variables d'entrée. Si A et B sont les variables d'entrée, alors $S = A.B$. S est vraie si A **ET** B sont vraies.
- **OR (OU) :** L'opérateur OR (OU) porte sur deux variables d'entrée. Si A et B sont les variables d'entrée, alors $S = A+B$. S est vraie si A **OU** B sont vraies.
- **NAND (NON ET) :** L'opérateur NAND (NON ET) porte sur deux variables d'entrée. Si A et B sont les variables d'entrée, alors $S = \overline{A.B}$. S est **fausse** si A **ET** B sont vraies. L'opérateur NAND est l'inverse de l'opérateur AND.
- **NOR (NON OU) :** L'opérateur NOR (NON OU) porte sur deux variables d'entrée. Si A et B sont les variables d'entrée, alors $S = \overline{A+B}$. S est **fausse** si A **OU** B sont vraies. L'opérateur NOR est l'inverse de l'opérateur OR.
- **XOR (OU exclusif) :** L'opérateur XOR (OU exclusif) n'est pas un opérateur de base car il peut être réalisé à l'aide des portes précédentes. Il porte sur deux variables d'entrée. Si A et B sont les variables d'entrée, alors $S = A \oplus B = \overline{A}B + A\overline{B}$. A est vraie si A est **différent** de B.
- **XNOR (NON OU exclusif) :** L'opérateur XNOR (NON OU exclusif) n'est pas non plus un opérateur de base. Il porte sur deux variables d'entrée. Si A et B sont les variables d'entrée, alors $S = \overline{A \oplus B} = \overline{A}B + A\overline{B}$; S est vraie si A **égale** B. L'opérateur XNOR est l'inverse de l'opérateur XOR.[3]

Opération	Symbole usuel	Symbole normalisé	Table de vérité	Tableau de Karnaugh																												
NOT - INV			<table border="1"> <tr><td>A</td><td>\overline{A}</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	\overline{A}	0	1	1	0	<table border="1"> <tr><td colspan="2">\overline{A}</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table>	\overline{A}		1	0	0	1																
A	\overline{A}																															
0	1																															
1	0																															
\overline{A}																																
1	0																															
0	1																															
AND - ET			<table border="1"> <tr><td>A</td><td>B</td><td>AB</td><td>A+B</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	AB	A+B	0	0	0	0	0	1	0	1	1	0	0	1	1	1	1	1	<table border="1"> <tr><td colspan="2">\overline{B}</td></tr> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td></tr> </table>	\overline{B}		0	0	0	1	1	1
A	B	AB	A+B																													
0	0	0	0																													
0	1	0	1																													
1	0	0	1																													
1	1	1	1																													
\overline{B}																																
0	0																															
0	1																															
1	1																															
OR - OU			<table border="1"> <tr><td>A</td><td>B</td><td>A+B</td><td>$\overline{A+B}$</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	A+B	$\overline{A+B}$	0	0	0	1	0	1	1	0	1	0	1	0	1	1	1	0	<table border="1"> <tr><td colspan="2">\overline{B}</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td></tr> </table>	\overline{B}		0	1	1	1		
A	B	A+B	$\overline{A+B}$																													
0	0	0	1																													
0	1	1	0																													
1	0	1	0																													
1	1	1	0																													
\overline{B}																																
0	1																															
1	1																															
XOR - OU Exclusif			<table border="1"> <tr><td>A</td><td>B</td><td>A ⊕ B</td><td>$\overline{A \oplus B}$</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> </table>	A	B	A ⊕ B	$\overline{A \oplus B}$	0	0	0	1	0	1	1	0	1	0	1	0	1	1	0	1	<table border="1"> <tr><td colspan="2">\overline{B}</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	\overline{B}		0	1	1	0		
A	B	A ⊕ B	$\overline{A \oplus B}$																													
0	0	0	1																													
0	1	1	0																													
1	0	1	0																													
1	1	0	1																													
\overline{B}																																
0	1																															
1	0																															
XNOR - NON OU Exclusif			<table border="1"> <tr><td>A</td><td>B</td><td>$\overline{A \oplus B}$</td><td>A+B</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	$\overline{A \oplus B}$	A+B	0	0	1	0	0	1	0	1	1	0	0	1	1	1	1	0	<table border="1"> <tr><td colspan="2">\overline{B}</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table>	\overline{B}		1	0	0	1		
A	B	$\overline{A \oplus B}$	A+B																													
0	0	1	0																													
0	1	0	1																													
1	0	0	1																													
1	1	1	0																													
\overline{B}																																
1	0																															
0	1																															
NAND - NON ET			<table border="1"> <tr><td>A</td><td>B</td><td>$\overline{A.B}$</td><td>A+B</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> </table>	A	B	$\overline{A.B}$	A+B	0	0	1	0	0	1	1	1	1	0	1	1	1	1	0	0	<table border="1"> <tr><td colspan="2">\overline{B}</td></tr> <tr><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	\overline{B}		1	1	1	0		
A	B	$\overline{A.B}$	A+B																													
0	0	1	0																													
0	1	1	1																													
1	0	1	1																													
1	1	0	0																													
\overline{B}																																
1	1																															
1	0																															
NOR - NON OU			<table border="1"> <tr><td>A</td><td>B</td><td>$\overline{A+B}$</td><td>$\overline{A.B}$</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> </table>	A	B	$\overline{A+B}$	$\overline{A.B}$	0	0	1	1	0	1	0	0	1	0	0	0	1	1	0	0	<table border="1"> <tr><td colspan="2">\overline{B}</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> </table>	\overline{B}		1	0	0	0		
A	B	$\overline{A+B}$	$\overline{A.B}$																													
0	0	1	1																													
0	1	0	0																													
1	0	0	0																													
1	1	0	0																													
\overline{B}																																
1	0																															
0	0																															

Tableaux I.1: les circuits fondamentaux

L'algèbre de BOOLE ne traite que de la logique combinatoire, c'est à dire des circuits numériques dont la sortie ne dépend que de l'état présent des entrées (sans mémoire des états passés).

Les lois fondamentales de l'algèbre de BOOLE se vérifient en écrivant les tables de vérités et en testant tous les cas possibles. Ces lois sont les suivantes :

Commutativité	$A.B=B.A$ $A+B=B+A$
Associativité	$A(B.C)=(A.B).C$ $A+(B+C)=(A+B)+C$
Distributivité	$A.(B+C)=(A.B)+(A.C)$ $A+(B.C)=(A+B).(A+C)$ $A(B.C)=(A.B).(A.C)$ $A+(B+C)=(A+B)+(A+C)$
Idempotence	$A.A=A$ $A+A=A$
Complémentarité	$A.\bar{A}=0$ $A+\bar{A}=1$
Identités remarquables	$1.A=A$ $1+A=1$ $0+A=A$ $0.A=0$

Tableau I.2 : les lois fondamentales de l'algèbre de BOOLE

I.2.1.2. Expression d'une fonction logique

Il existe deux méthodes pour exprimer une fonction logique : soit donner directement son équation logique, soit utiliser une table de vérité. A un nombre fini N de variables d'entrée correspond 2^N combinaisons possibles. Si par exemple S est une fonction de trois variables, il y aura 2^3 combinaisons possibles.

I.2.1.3. Simplification des fonctions logiques

L'expression d'une fonction logique sous sa forme la plus simple n'est pas quelque chose d'évident. Trois méthodes sont utilisables :

- Le raisonnement, on cherche, à partir du problème à résoudre, l'expression la plus simple possible. Evidemment, cette méthode ne garantit pas un résultat optimal.
- La table de vérité et les propriétés de l'algèbre de BOOLE, c'est plus efficace, mais il est facile de rater une simplification, notamment quand la fonction est compliquée.
- La méthode graphique des tableaux de Karnaugh, c'est la méthode la plus efficace car elle garantit le bon résultat..

I.2.1.4. Représentation des nombres : les codes

On regroupe souvent les bits par paquets afin de coder des nombres entiers, réels ou bien des caractères. Pour cela, un groupe de bits doit former un code. Il existe de très nombreux codes et nous allons voir maintenant les principaux.

➤ Base d'un système de numérotation

Une base est constituée par l'ensemble des chiffres (ou caractères) différents qu'utilise le système de numération. Tout nombre peut se décomposer en fonction des puissances entières, positives ou négatives, de la base de numération. Il peut donc se mettre sous forme polynomiale.

➤ Changement de base

Une question importante est le passage d'une base à l'autre : le changement de base. Soit $(N)_{10}$, combien vaut $(N)_2$ ou $(N)_{16}$.

La conversion vers la base 10 est la plus simple. Il suffit de calculer la valeur du polynôme.

Le changement de base décimal \rightarrow binaire, Il faut diviser le nombre à convertir par 2. On conserve le reste (qui vaut 0 ou 1) et on répète la division sur le quotient, jusqu'à ce que le quotient vaille 0. On écrit alors tous les restes à la suite, le premier reste obtenu étant le poids faible.

Le passage d'hexadécimal vers binaire et de binaire vers hexadécimal est plus simple. Dans le premier cas, il suffit de remplacer chaque chiffre hexadécimal par le groupe de 4 bits correspondant.

Dans le deuxième cas, il suffit de regrouper les bits par paquet de 4 en partant du poids faible. Il faudra éventuellement rajouter des bits à 0 à gauche pour terminer le groupe de 4 bits de poids fort.

Pour passer d'hexadécimal en décimal (et vice versa), on peut passer par l'intermédiaire du binaire ou faire le calcul directement par la méthode des divisions successives.

➤ **Code BCD**

Le code BCD (Binary Coded Decimal) permet de coder un nombre décimal en binaire. A chaque chiffre décimal, on fait correspondre un groupe de 4 bits comme pour la base hexadécimale. Mais ici, il n'y a pas de valeur supérieure à 9.

➤ **Le code Gray**

Le code Gray est un code adjacent (ou un seul bit change quand on passe d'une valeur à la valeur suivante). On l'appelle aussi le code binaire réfléchi. On l'utilise dans les tableaux de Karnaugh mais aussi en conception numérique.

➤ **Le code ASCII**

Il n'y a pas que les nombres qui doivent être codés en binaire, mais aussi les caractères comme les lettres de l'alphabet, les signes de ponctuation. Le code le plus connu et le plus utilisé est le code ASCII (American Standard Code for Information Interchange). Le code ASCII est un jeu normalisé de 128 caractères codés sur 7 bits, devenu un standard quasi universel. Il comporte tous les caractères alphanumériques non accentués et est lisible par pratiquement n'importe quelle machine. Ce sont les 8 premières lignes du tableau suivant :

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht	nl	vt	np	cr	so	si
1	dle	dc1	dc2	dc3	dc4	nak	syn	etb	can	em	sub	esc	fs	gs	rs	us
2	sp	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	del
8	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht	nl	vt	np	cr	so	si
9	dle	dc1	dc2	dc3	dc4	nak	syn	etb	can	em	sub	esc	fs	gs	rs	us
a		ı	ć	ł	đ	ě	ę	ğ	ı	©	ı			ŋ	õ	
b	°	±	²	³	ł	μ	¶	ű	ÿ	ı	ž	ž	ıj	ı	ı	ı
c	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
d	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	
e	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
f	ö	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Tableau I.3 : le code ASCII

Les 32 premiers codes sont utilisés comme caractères de contrôle pour représenter, par exemple, une fin de ligne ou une tabulation. Le code ASCII ne contient pas de caractères accentués et il a été complété par le code ISO-8859-1 (ou Latin 1). Les 128 premiers caractères correspondent au code ASCII, les 128 suivants aux caractères accentués et caractères spéciaux (voir les 8 dernières lignes du tableau) qu'est le code ASCII étendu.

Unicode est un jeu de caractères codé sur 16 bits (contre 7 ou 8 bits pour les standards actuels) qui permet le codage des caractères utilisés par toutes les langues du monde au sein d'une table unique. 16 bits permettent de coder 65 536 (2 puissance 16) caractères différents ce qui couvre largement les besoins en la matière. Les 256 premiers caractères d'Unicode correspondent au jeu ISO Latin 1.

I.2.1.5. Les circuits logiques combinatoires

Les circuits logiques combinatoires sont des circuits constitués des portes fonctionnant simultanément et réalisant une ou plusieurs fonctions logiques.[3]

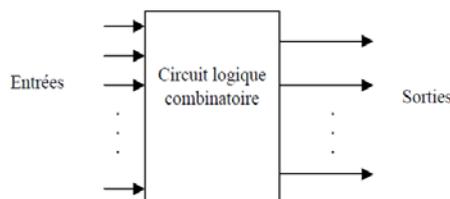


Figure I.2 : Circuit logique combinatoire

A une combinaison d'entrées ne correspond qu'une seule combinaison de sorties. La sortie apparaît après application de l'entrée avec un certain retard qui est le temps de propagation dans la logique interne. Ce temps est déterminé par la technologie utilisée, le nombre de portes traversées et la longueur des interconnexions métalliques.

Les circuits combinatoires peuvent servir par exemple :

- à traduire des bits en chiffres représentant un nombre (ou des lettres ou un code particulier). On appelle ces circuits des codeurs (ou bien des décodeurs pour l'opération inverse).
- à effectuer des opérations arithmétiques sur des nombres. Par exemple, un additionneur ou un multiplieur.
- à transmettre ou recevoir des informations sur une ligne unique de transmission (une ligne série), ce qui nécessite de transformer un nombre écrit sous forme parallèle en une suite de bits mis en série et vice-versa. C'est le rôle des circuits multiplexeur/démultiplexeur.

- **Le multiplexeur**

Un multiplexeur est un dispositif qui permet de transmettre sur une seule ligne des informations en provenance de plusieurs sens ou à destination de plusieurs cibles. Il a pour rôle d'aiguiller sur une sortie unique un signal prélevé parmi plusieurs autres de nature différente. C'est en quelque sorte un circuit qui réalise une fonction inverse du décodeur.

C'est un circuit comportant 2^n entrées d'informations, n entrées d'adresse et une seule sortie. Lorsque la validation est active la sortie prend l'état d'une entrée.

- **Le démultiplexeur**

C'est un circuit complémentaire du multiplexeur, il comporte une entrée, 2^n sorties et n entrées d'adresse.

- **Le codeur**

Un codeur permet de transmettre une information de la meilleure façon. Cette transmission est souvent plus simple, plus rapide et plus sécurisée. Le codeur fait correspondre à l'activation d'une entrée particulière une combinaison de bits en sortie. C'est un circuit à une entrée et n sorties.

- **Le décodeur**

le décodeur réalise la fonction inverse du codeur. C'est un circuit comportant n entrées et 2^N sorties. Lorsque le signal de validation est actif, seule la sortie dont le numéro correspond à la valeur binaire affichée sur l'entrée est active. Toutes les autres sont inactives.

Le décodeur est un circuit très employé dans les microprocesseurs, son rôle est de sélectionner une adresse précise de mémoire parmi un lot important d'adresses différentes.

- **L'encodeur**

Un encodeur est un système qui comporte N lignes d'entrées et n lignes de sorties, lorsqu'une des lignes d'entrée est active, l'encodeur fournit en sortie un mot de n bits correspondant au codage de l'information identifiée par la ligne activée.

- **Les circuits arithmétiques**

Les circuits arithmétiques effectuent des opérations binaires telles que l'addition, la multiplication, la comparaison. Beaucoup de ces circuits sont aujourd'hui obsolètes et ne sont plus commercialisés en tant que composant discret mais la fonction existe toujours à l'intérieur des circuits comme les microprocesseurs ou les ASIC, on en trouve les additionneurs, les soustracteurs, les comparateurs

I.2.2. Logique séquentielle

Un système combinatoire est tel que l'état de ses sorties ne dépend que de l'état des entrées. Il peut être donc représenté par une table de vérité ou un tableau de karnaugh, pour chaque sortie. Il est possible donc d'écrire l'équation logique de chaque sortie, en fonction seulement des entrées.

Quant au système séquentiel, il dépend non seulement des entrées, mais aussi de l'état précédant du système. On retrouvera les mêmes états des entrées à plusieurs étapes, alors que les sorties seront différentes. Il est donc impossible de représenter par un tableau de karnaugh.

Pour les circuits de logique séquentielle nous devons tenir compte de l'état du système. Ainsi les sorties dépendent des entrées mais également de l'état du système. Celui-ci

dépend aussi des entrées. Si nous notons Q l'état d'un système séquentiel, X ses entrées et Y ses sorties, nous avons de manière générale :

$$Q = f(X, Q) \qquad Y = g(X, Q)$$

La logique séquentielle permet de réaliser des circuits dont le comportement est variable avec le temps. L'état d'un système constitue une mémoire du passé.[3]

Lorsque les changements d'état de divers composants d'un circuit séquentiel se traduisent à des instants qui dépendent des temps de réponse des autres composants et des temps de propagation des signaux on parle de la logique séquentielle asynchrone. Cependant les retards peuvent ne pas être identiques pour toutes les variables binaires et conduisent à certains aléas.

Ceux-ci peuvent être évités en synchronisant la séquence des diverses opérations logiques sur les signaux périodiques provenant d'une horloge. La logique séquentielle est alors dite synchrone : tous les changements d'état sont synchronisés sur un signal de contrôle.

Les éléments de base sont les bascules, et une association de bascules nous permettra de construire des registres, des compteurs, des décompteurs et des diviseurs.

Nous allons étudier successivement les bascules RS, RST, D et JK

I.2.2.1. Les bascules

Les bascules sont les éléments de base de la logique séquentielle, un peu comme l'étaient les portes logique en logique combinatoire. Nous verrons par la suite qu'elles permettent de réaliser de nombreux systèmes (compteurs, registres, mémoires...).

De façon générale une bascule se caractérise par :

- L'existence de deux états de sortie stables. C'est-à-dire deux états dans lesquels la bascule peut se maintenir infiniment sans action extérieure.
- Des entrées de commande permettant de passer au choix d'un état à un autre.
- Nous allons tout de suite donner l'exemple de la bascule la plus simple très connue sous le nom de bascule RS.

Il existe deux types de bascules : les bascules synchrones, possédant une entrée d'horloge de synchronisation, et les bascules asynchrones, ou verrous

➤ La bascule RS

La bascule RS est la plus simple des bascules. Elle est réalisée à partir de deux portes NOR ou de deux portes NAND. La bascule possède deux entrées notées R et S ainsi que deux sorties conventionnellement notées Q et \bar{Q}

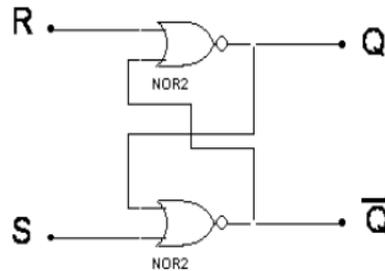


Figure I.3: schéma de la bascule RS

➤ Bascule RSH

La bascule RSH est une bascule RS synchronisée par un signal d'horloge H. Lorsque H=0, la bascule est dans l'état mémoire. Lorsque H=1, la bascule fonctionne comme une bascule RS. Cette bascule a toujours un état interdit et fonctionne sur les niveaux d'horloge.

➤ La bascule RS avec validation (RS latch)

Le schéma de principe d'une bascule RS avec validation est représenté sur la Figure ci-dessus. On reconnaît facilement la bascule RS dont les entrées sont maintenant notées R' et S' et sur lesquelles on a ajouté deux portes AND. L'entrée de validation E (pour Enable en anglais) permet de contrôler l'ouverture des 2 portes AND.

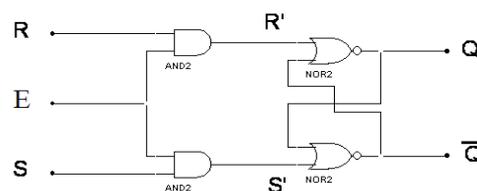
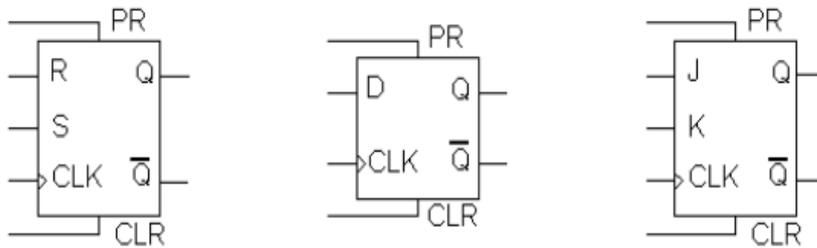


Figure I.4: Schéma d'une bascule RS avec entrée de validation

➤ La bascule JK

Identique aux bascules RS pour les entrées autres que 11 (l'entrée J correspond à S et l'entrée K à R). Pour les entrées 11 ces bascules fonctionnent en mode Toggle à savoir que leur sortie Q change de valeur à chaque impulsion d'horloge. Par rapport aux bascules RS,

les bascules JK permettent d'utiliser toutes les combinaisons des entrées. On a représenté sur la figure les symboles utilisés pour chacune de ces bascules :



a)

b)

c)

Figure I.5 : Représentation des principales de bascules synchrones. a) la bascule RS, b) la bascule D, c) la bascule JK

➤ **La bascule D**

La bascule D (pour *Data*) est une bascule comportant uniquement une entrée de données : D. La valeur de l'entrée est recopiée sur la sortie à chaque front d'horloge. Cette bascule permet d'assurer un état de sortie stable entre deux fronts d'horloge, et ainsi d'ignorer toute valeur transitoire apparaissant sur son entrée au cours d'un cycle d'horloge. On ajoute parfois un signal reset afin de pouvoir forcer la valeur initiale à la mise sous tension. Il existe des versions où les changements d'état ont lieu au moment des fronts descendants de l'horloge. Elles se signalent par une barre supplémentaire (en dessous à 45°) sous l'entrée H.

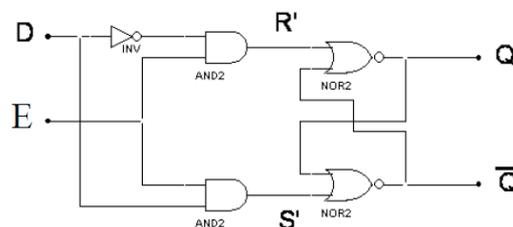


Figure I.6 : schéma de la bascule D

➤ **Bascule D-latch**

La bascule D-Latch est une bascule conçue sur le même principe que la RSH. Elle est obtenue à partir d'une bascule RSH en ne considérant que les deux combinaisons $(R, S)=(0,1)$ et $(1,0)$

- **Avantage :**
 - Pas d'état interdit
 - Insensibilité aux parasites ($H=0$)
- **Inconvénients :**
 - Sensibilité aux parasites ($H=1$)

➤ **Bascule D (Maître-Esclave)**

Les bascules maître-esclaves permettent de diminuer la sensibilité aux parasites en minimisant la période de transparence. Elles fonctionnent sur le front d'horloge.

❖ **Différents types de déclenchements de bascules:** on distingue principalement deux types :

➤ **Basculs synchrones/basculs asynchrones**

Les systèmes séquentiels sont classés en 2 catégories :

Contrairement aux systèmes asynchrones qui peuvent changer d'état sur une transition d'une entrée quelconque, les systèmes synchrones ne changent d'état que sur l'activation de l'horloge. Ces systèmes utilisent habituellement des bascules D, JK et T. Les systèmes synchrones sont habituellement plus stables et prévisibles que les machines asynchrones et sont aussi plus faciles à réparer lorsque des fautes sont présentes. Comme pour les machines asynchrones, les machines synchrones sont divisées en deux types de machines. Maître

- **Asynchrone :** on remarque précédemment que la bascule change d'état que lorsque R ou S changent d'état. On dit alors que la bascule a un fonctionnement asynchrone.
- **Synchrone :** une bascule fonctionne de manière synchrone lorsque ses sorties changent d'états à la cadence d'une horloge.

I.2.2.3. Registres

Un registre est un ensemble de bascules, synchronisées par la même horloge.

Les registres sont à la base d'opérations couramment utilisées dans les ordinateurs : mémorisation provisoire (mémoires-tampon), décalages, rotations, etc.

❖ Différents types de registres

On distingue quatre types de registres selon la façon dont sont utilisées les entrées et les sorties : en parallèle ou en série.

- Registres à entrées parallèles
- Registres à entrée série, sortie série
- Registres à entrée série, sorties parallèles
- Registres à entrées parallèles, sortie série

I.2.2.4. Compteurs

Les compteurs sont des éléments essentiels de la logique séquentielle car ils sont utilisés pour les systèmes arithmétiques, mais ils permettent aussi d'établir une relation d'ordre de succession d'événements. Ils sont utilisés notamment pour le comptage d'événements, pour diviser la fréquence, dans les automates programmables, etc.

Les compteurs sont classés en deux catégories selon leur mode de fonctionnement. On distingue les compteurs asynchrones des compteurs synchrones.

➤ **Compteurs asynchrones ou compteurs série**

La première bascule est synchronisée par une horloge externe mais le déclenchement des autres bascules est déterminé par une combinaison logique des sorties des bascules précédentes. La propagation de l'ordre de changement d'état se fait en cascade. Les sorties des bascules ne changent pas d'état exactement en même temps car elles ne sont pas reliées au même signal d'horloge.

Le schéma ci-dessous est un exemple de connexion d'horloge pour obtenir un compteur asynchrone avec propagation de l'ordre de changement d'état en cascade.

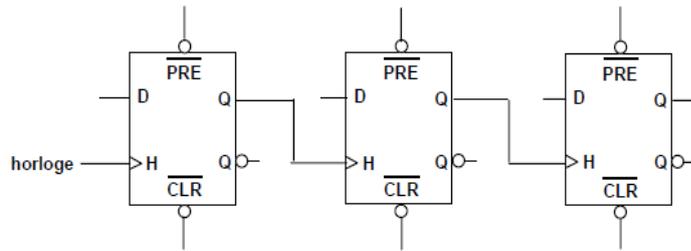


Figure I.7 : Compteur asynchrone ou série

➤ Compteurs synchrones ou parallèle

Le signal d'horloge externe est connecté à toutes les bascules et permet de les déclencher toutes simultanément.

Le schéma ci-dessous est un exemple de connexion de l'horloge pour obtenir un compteur synchrone : la synchronisation des bascules est faite par le même signal d'horloge.

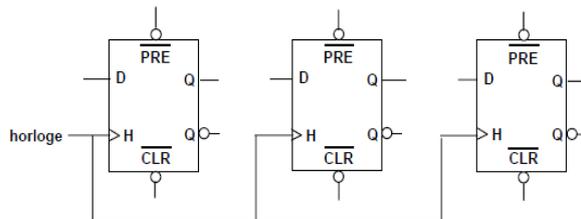


Figure I.8 : compteur synchrone ou parallèle

Pour chacune de ces catégories, les compteurs sont classés selon leurs séquences, le nombre d'états ou le nombre de bascules qu'ils comportent.

I.3. Discussion :

Le numérique est très utilisé dans notre vie moderne, il permet à moindre coût d'obtenir une qualité excellente dans nombreuses applications.

Il s'agit d'un domaine scientifique qui s'intéresse aux systèmes dont les états parcourent un ensemble fini de possibilité. Grâce à la prédétermination des états électriques, on dispose de systèmes qui se comportent de manière parfaitement stable, permettant de s'affranchir de bon nombre de parasites se superposant à tout courant électrique et électronique.

Chapitre II :
Les systèmes embarqués

II.1. Préambule

L'objectif de ce chapitre est de présenter clairement les systèmes embarqués. Pour ce faire, nous présentons dans une première section l'architecture et les méthodes des diverses générations des systèmes embarqués et notamment les systèmes sur une puce. Dans la deuxième section sont présentés quelques modèles qui peuvent être utilisés pour concevoir ces systèmes, c'est-à-dire les décrire, les simuler, les synthétiser ou les générer.

II.2. Définition des systèmes embarqués

Un système embarqué est un système inclus dans un autre système, il est défini comme un système électronique et informatique autonome souvent temps réel, spécialisé dans une tâche bien précise. Ses ressources sont généralement limitées, le système comprend une partie matérielle et une autre logicielle qui sont conçues spécifiquement pour réaliser une fonction dédiée. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires. [5]

II.2.1. Caractéristiques des systèmes embarqués

Les systèmes embarqués ont pour but de permettre aux objets usuels de réagir à l'environnement, ils peuvent aussi apporter une interface avec l'utilisateur. Généralement ils sont caractérisés par :

- Coût réduit, maximisation rapport performance/prix.
- Capacité mémoire adaptée.
- Volume restreint (compact, pas modulaire).
- Exécution temps réel (souvent).
- Fiabilité et sécurité de fonctionnement.
- Consommation d'énergie maîtrisée (très faible en cas d'utilisation sur batterie)
- Capacité de calcul approprié à l'application.

La structure de base de ces systèmes est donnée par la figure 1 : l'environnement est mesuré par divers capteurs. L'information des capteurs est échantillonnée pour être traitée par le cœur du système embarqué. Puis le résultat du traitement est converti en signaux analogiques qui génèrent les actions sur l'environnement (afficheur d'informations pour l'utilisateur, actionneurs, transmission d'informations ... etc.).

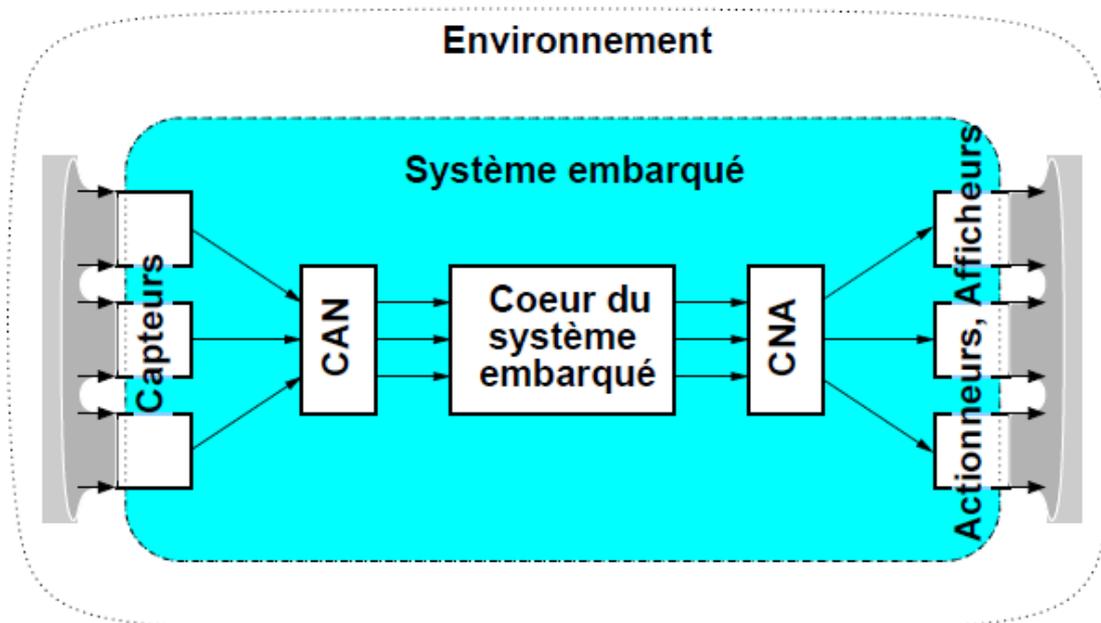


Figure II.1 : Un système embarqué dans son environnement

II.2.2 Contraintes de conception des systèmes embarqués

L'appellation système embarqué regroupe de nombreux systèmes clairement distincts, tant au niveau des applications qu'au niveau des performances requises. De ce fait, il est difficile de définir des contraintes universelles. Néanmoins, il est tout de même possible de lister un certain nombre de contraintes que l'on doit couramment respecter dans le cadre de la conception des systèmes embarqués, parmi lesquels nous trouvons :

- La limitation des ressources de traitement est liée à une limitation de la surface disponible pour le système et pour les composants qui le constitue. Il s'agit aussi d'une limitation des ressources de mémorisation.
- Le débit de données en entrée et en sortie est aussi limité. Néanmoins, la tendance actuelle des applications va vers une augmentation des besoins en termes de vitesse et de quantité des échanges de données, donc vers une augmentation forte des débits.
- Le nombre de connexion du système vers l'extérieur est limité.
- La limitation en termes de consommation de puissances et d'énergie est souvent importante, car un système embarqué n'est qu'une partie d'un système hôte qui peut être lui-même limité en source d'énergie. C'est bien entendu le cas de tout système alimenté par une batterie.

- Les besoins de flexibilité logicielle et matérielle souvent grands. D'une part, la flexibilité permet une meilleure intégration du système dans un large domaine d'application. D'autre part, il permet au système d'évoluer dans le temps par des mises à jour logicielle, matérielle grâce à l'utilisation de mémoires programmables et de circuits matériels reconfigurables.

II.2.3. Domaines d'application des systèmes embarqués

Les systèmes embarqués sont désormais utilisés dans des applications diverses tels que le transport (avionique, espace, automobile, ferroviaire), dans les appareils électriques et électroniques (appareils photo, jouets, postes de télévision, électroménager, systèmes audio, téléphones portables), dans la distribution d'énergie, dans l'automatisation, etc.

Les systèmes embarqués sont définis comme des ensembles matériels logiciels intégrés dans le but d'effectuer des tâches fonctionnelles précises. Les systèmes et logiciels embarqués jouent un rôle majeur dans les quasi-intégralités des secteurs industriels et sont très présents au sein d'industries historiques : les transports terrestres et l'aéronautique, le nucléaire, la défense et l'espace, les télécommunications (téléphones portables, assistants personnels, application internes chez les opérateurs, ils jouent et joueront un rôle de plus important dans de nombreux autres secteurs comme :

- La production, la distribution et la gestion de l'utilisation de l'énergie.
- La production industrielle (automatique, systèmes discrets et continus).
- L'instrumentation médicale.
- Le bâtiment (domotique).
- L'électronique grand public (terminaux mobiles, multimédia, jeux et loisirs numériques).
- La logistique (commerce et distribution).
- Les infrastructures urbaines (eau, trafic, captation de la qualité de l'aire).
- La sécurité (vidéosurveillance, moyens d'identification)
- Les transactions bancaires et commerciales (terminaux de paiement, cartes à puce).

II.2.4. Systèmes embarqués spécifiques et les systèmes monopuce

Les systèmes électroniques sont de plus en plus présents dans la vie courante. Les ordinateurs et micro-ordinateurs sont des systèmes électroniques bien connus. Mais l'électronique se trouve maintenant embarquée dans de très nombreux objets usuels : les téléphones, les agendas électroniques, les voitures. Ce sont ces systèmes électroniques enfouis dans les objets usuels qui sont appelés systèmes embarqués.

➤ **Système embarqué spécifique**

Lorsqu'un système est utilisé dans une tâche bien précise, il est souvent plus efficace et économe s'il est spécifique à cette fonctionnalité que s'il est général. Les systèmes embarqués sont très souvent utilisés dans ces conditions, et il est donc intéressant qu'ils soient conçus spécifiquement pour les fonctions qu'ils doivent remplir.

➤ **Système monopuce**

Les progrès réalisés par les fondeurs de circuits permettent aujourd'hui d'envisager l'intégration sur une même puce d'un système embarqué complet. Ces systèmes monopuces (SOC : system on a chip) apportent des changements importants dans les flots de conception classiques.

Dans les systèmes électroniques classiques, une grande difficulté était les problèmes électriques dus aux grandes dimensions des cartes électroniques. Ces problèmes limitaient notamment la vitesse de communication, ce qui faisait qu'il pouvait y avoir sur une carte des composants allant d'une grande vitesse, mais ne pouvant communiquer entre eux qu'à vitesse réduite. C'était particulièrement critique pour la communication entre le processeur et la mémoire : quelque soit la vitesse du processeur, il devait aller lire ses instructions en mémoire. Pour pallier ces problèmes il était nécessaire d'utiliser des caches. L'inconvénient des caches est qu'ils sont des systèmes très complexes à étudier et qu'ils sont de gros facteurs d'indéterminisme. Avec les systèmes monopuces, la communication reste toujours un goulet d'étranglement, car elle est très consommatrice de surface, mais avec un facteur bien moindre. En effet le fait d'avoir sur la même puce l'ensemble du système raccourcit les chemins de communication et facilite la construction d'architecture s'accordant aux localités de calcul de l'application. Ces facilités permettent ainsi souvent de s'affranchir des caches. Les systèmes

monopuces sont aussi moins encombrant et surtout, ils peuvent consommer moins : en effet les données doivent transiter par des chemins beaucoup moins longs, l'énergie nécessaire à cette transmission est donc plus faible.

Les systèmes monopuces apportent aussi des changements dans les habitudes de conception. Notamment la frontière entre le logiciel et le matériel n'est plus aussi nette : en effet avec les anciens systèmes le matériel était déjà conçu lorsqu'il fallait concevoir le logiciel. Avec les systèmes monopuces, les deux doivent être conçus en même temps, cela augmente la complexité de la conception, mais cela offre aussi plus de liberté ; chaque partie peut être réalisé en logiciel, en matériel, ou d'une manière mixe.

II.2.5. Architecture des systèmes embarqués

Dans cette section nous présentons les architectures supportées par les trois générations d'outils de conception :

II.2.5.1. les systèmes embarqués de première génération :

➤ Partie matérielle des systèmes embarqués première génération

Les premiers systèmes embarqués supportés par des outils tels que COSYMA et VULCAN étaient très simples : ils étaient constitués par un processeur qui contrôlait un nombre restreint de CIAS (Circuits Intégrés à Applications Spécifiques) ou ASIC qui étaient appelés périphériques. Cette architecture est représentée dans la figure II.2, II.3 Les communications de cette architecture se situent au niveau du bus du processeur et sont de type : maître/esclave où le processeur est le maître et les périphériques sont les esclaves.[8]

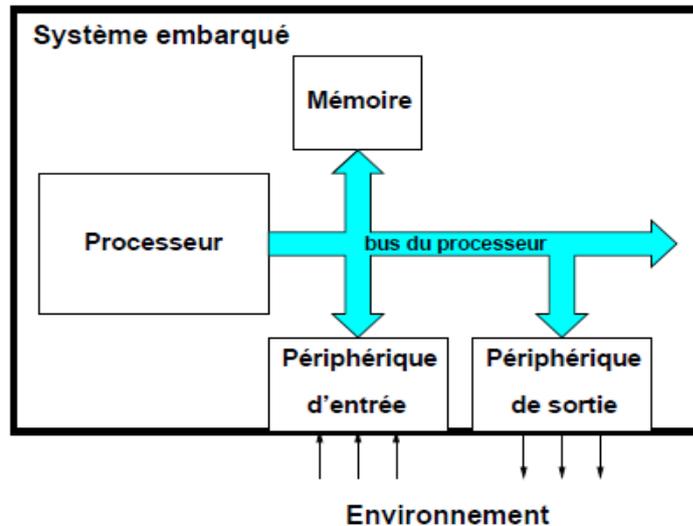


Figure II.2 : architecture embarquée de première génération

Les périphériques de ces architectures étaient essentiellement des capteurs et des actionneurs (contrôleurs magnétiques, sorties, etc.), le processeur est dédié au calcul et au contrôle de l'ensemble du système.

Les microcontrôleurs assemblent sur une même puce le processeur et des périphériques.

➤ **Partie logicielle des systèmes embarqués de première génération**

Ne devant pas exécuter des nombreuses opérations simultanées (le nombre de périphériques et de fonctions restent restreint), les parties logicielles étaient constituées d'un seul programme. La réaction aux événements était effectuée par le biais de routines de traitement d'interruptions.

Cette partie logicielle était décrite directement d'un langage d'assemblage ce qui permettait d'obtenir un code efficace d'une taille petite.

II.2.5.2. Les systèmes embarqués de deuxième génération

Les premiers systèmes embarqués ne pouvaient fournir que des fonctions simples ne requérant que peu de puissance de calcul. Leur architecture ne peut pas supporter des fonctionnalités requises pour les systèmes embarqués actuels à qui il est demandé non seulement d'effectuer du contrôle, mais aussi des calculs complexes tels que ceux requis pour

le traitement numérique du signal. De nouveaux outils tels que N2C permettent de traiter des architectures plus complexes. [8]

➤ **Partie logicielle des systèmes embarqués de deuxième génération**

L'architecture des systèmes embarqués de deuxième génération est composée d'un processeur central, de nombreux périphériques, et souvent de quelques processeurs annexes contrôlés par le processeur central. Le processeur central est dédié au contrôle de l'ensemble du système. Les processeurs annexes sont utilisés pour le calcul, il s'agit souvent de processeurs spécialisés comme les DSP. Une telle architecture est représentée dans la figure II.3, dans laquelle plusieurs bus de communication peuvent être nécessaires : chaque processeur dispose de son bus de communication.[8]

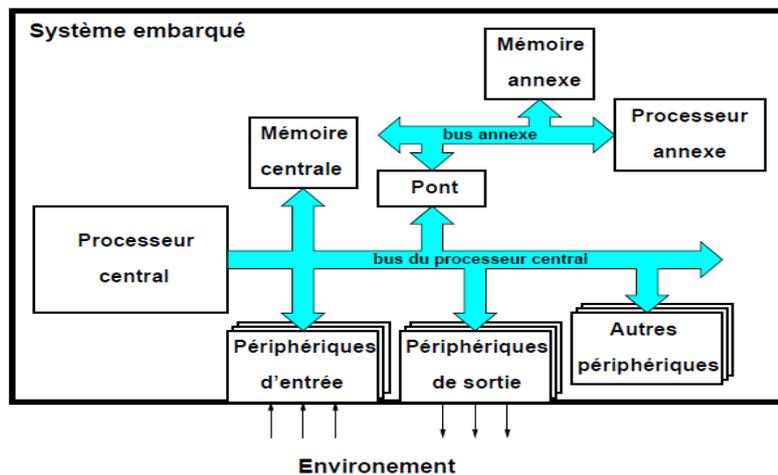


Figure II.3: Architecture embarquée de deuxième génération

➤ **Partie logicielle des systèmes embarqués de deuxième génération**

La partie logicielle des systèmes embarqués de deuxième génération est répartie sur plusieurs processeurs. Les systèmes actuels sont trop complexes pour pouvoir être gérés par un unique programme sur le processeur principal. Il est donc nécessaire d'avoir une gestion multitâche sur ce processeur, et un système d'exploitation est couramment employé dans ce but.

Le logiciel du processeur central est souvent décrit dans un langage de haut niveau tel que le C. le logiciel des processeurs annexes est souvent trop spécifique pour être entièrement

décrit dans un langage de haut niveau, et l'utilisation des langages d'assemblage est nécessaire.

II.2.5.3. Les systèmes embarqués de troisième génération

Les progrès de l'intégration permettent d'envisager des circuits pouvant contenir plusieurs milliers de portes. Il devient donc techniquement possible de fabriquer des systèmes embarqués pouvant remplir toutes les fonctionnalités souhaitées.

➤ **Partie matérielle des systèmes embarqués de troisième génération**

Pour pouvoir supporter conjointement les besoins en puissance et en flexibilité, ces architectures comprennent de plus en plus des processeurs, qui peuvent chacun se comporter en maître, l'architecture couramment utilisée, basée sur un processeur central contrôlant le reste du système, n'est donc plus suffisante.

Alors qu'auparavant le goulet d'étranglement était les ressources en calcul, de nos jours il est situé plutôt au niveau des communications. Ce sont celles qui définissent désormais l'architecture, et non plus les ressources de calcul. La figure II.4 donne des exemples d'architectures centrées sur les communications. Dans cette figure, tous les éléments (processeur, ASIC ou mémoires), sont traités de la même manière. Le premier exemple est basé sur des communications par bus : ce modèle de communication consomme peu de surface, mais risque de devenir un goulet d'étranglement. Le deuxième est basé sur des communications en barres croisées très performantes mais aussi très coûteuses en surface. Le troisième exemple donne une solution intermédiaire, par réseau commuté. Enfin le dernier exemple montre qu'il est possible de mixer plusieurs modèles de communication et d'apporter de la hiérarchie

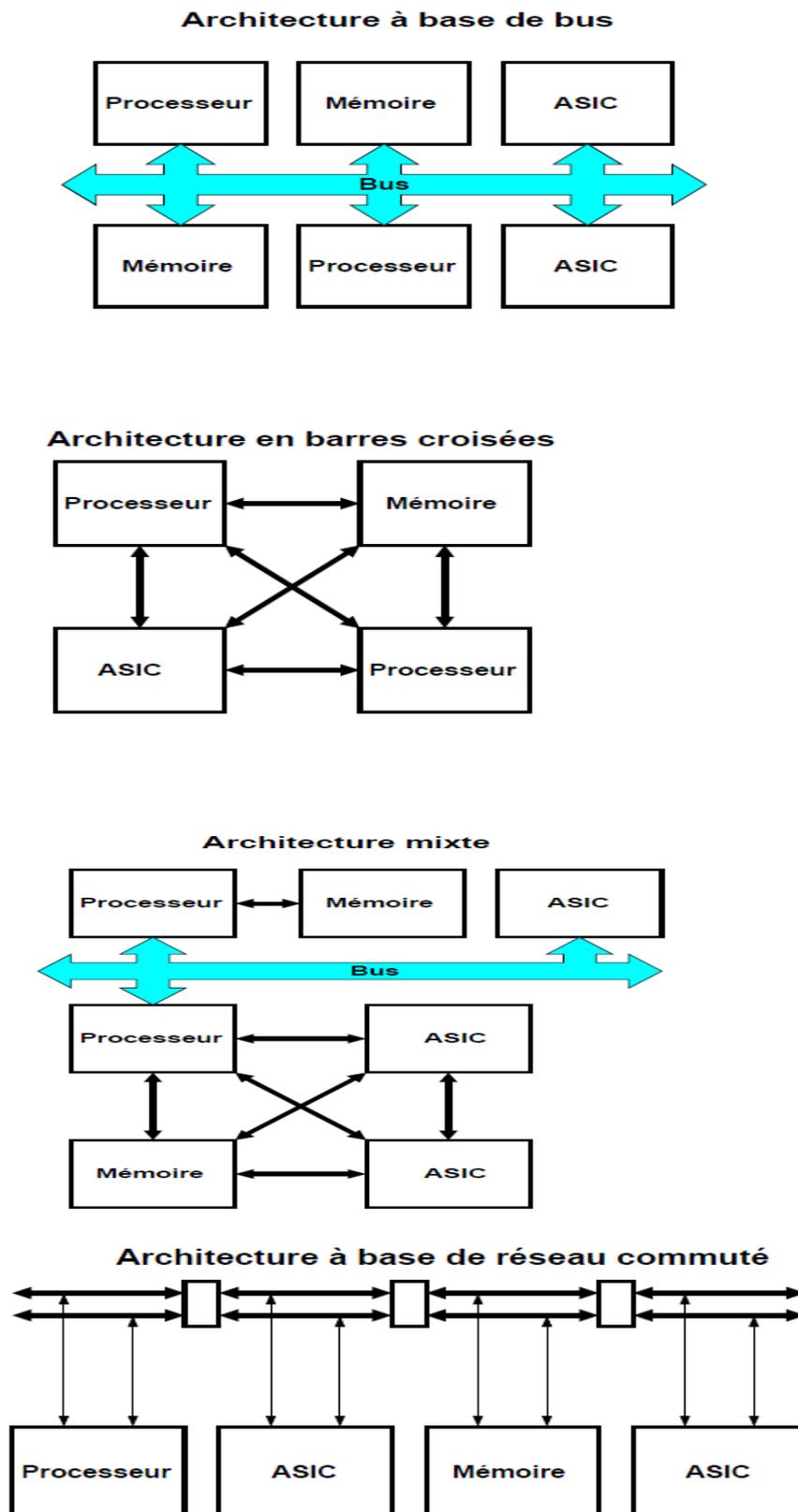


Figure II.4: Architectures embarquées de troisième génération

Autour de ce modèle d'architecture centré sur les communications, se greffent les autres modèles d'architectures : architecture des éléments de calculs et des mémoires. L'architecture des éléments de calcul consiste à définir quels sont les éléments principaux et quels sont leurs périphériques de manière à les grouper dans une architecture locale. L'architecture des mémoires sert à définir quelles sont les mémoires locales à un groupe et qui sont celles qui seront partagées. [8]

➤ **Parties logicielles des systèmes embarqués de troisième génération**

Les parties logicielles ont beaucoup gagné en importance dans les systèmes embarqués. Plusieurs systèmes d'exploitation sont parfois nécessaires pour les divers processeurs de l'architecture. De plus, la complexité et la diversité des architectures possibles font qu'il devient de plus en plus nécessaire d'abstraire les tâches logicielles des détails du matériel.

Toute cette complexité est donc reportée dans les systèmes d'exploitation, qui deviennent de plus en plus complexes.

Cette complexité logicielle et matérielle entraîne de nombreuses alternatives. En particulier l'aspect multiprocesseur apporte des alternatives pour les systèmes d'exploitation : il peut y avoir un seul système pour tous les processeurs (solution difficilement applicable lorsque les processeurs sont hétérogènes), ou il peut y avoir un système par processeur (solution qui peut être plus coûteuse. [8])

II.2.6. Avantage des systèmes embarqués

Les avantages des systèmes intégrés sont :

- La faible consommation d'énergie
- Utilisation de faibles capacités mémoire et de petits processeurs (4 bits ou 8 bits)
- La consommation de ressources telle que : le poids, la taille physique, la résistance aux vibrations
- Il rend des services bien précis (contrôle, surveillance, communication)
- Le système doit toujours fonctionner correctement (faible coût avec une redondance minimale)

- Sûreté de fonctionnement du logiciel (système opérationnel même quand un composant électronique lâche)
- Consommation électrique minimisée. [9]

Parmi les systèmes embarqués les plus utilisés actuellement : l'automobile. En voici quelques avantages de ce dernier :

➤ **Réduction des réponses :**

- Diminution des parasites en optimisant les trajets de livraison
- Optimisation de parc véhicules en visualisant en temps réel l'ensemble de flotte sur une carte interactive
- Diminution des heures supplémentaires en optimisant le temps de travail de conducteurs

➤ **Efficacité :**

- Communication à distance par la télé transmission des tournées, l'ajout et la suppression de livraisons en cours de tournée.
- Gagne du temps par l'automatisation et la centralisation de toutes les données.
- Evite les erreurs par la suppression de la saisie manuelle.
- Réduction du temps de clôture en fin de journée (conducteur et administratif).
- Amélioration de la clarté et la qualité des documents remis aux clients.
- Pilotage d'activité sur le terrain en accédant à toutes les informations de pilotage grâce au site Web.

➤ **Sécurité :**

- Amélioration de la sécurité des conducteurs en leur mettant à disposition des outils modernes qui supprimeront l'utilisation du téléphone portable au volant.
- Sécurité des conducteurs grâce à l'utilisation d'un Dispositif d'Alarme Travailleur Isolé.
- Amélioration de la réactivité d'intervention grâce à la remontée en temps réel des situations d'urgence : accident, renversement de camion.
- Sécurité des encaissements par une facturation sur site et l'utilisation de terminaux de paiement.

- Sécurité des cargaisons et la gestion des écarts de stocks afin d'éviter le risque de fraude.

II.2.7. Système embarque temps réel

Un système embarqué doit généralement respecter des contraintes temporelles fortes (Hard Real Time).

Un système est dit Temps Réel lorsque l'information après acquisition et traitement reste encore pertinente ; cela veut dire dans le cas d'une information arrivant de façon périodique (interruption), les temps d'acquisition et de traitement doivent rester inférieurs à la période du système.

Ce sont des systèmes liés au contrôle de procédés l'exécution de programmes de dans ces système doit terminer avant une date appelée échéance au-delà de laquelle les résultats ne sont plus valides. On distingue deux types de systèmes embarqué temps-réel :

II.2.7.1. Caractéristiques

Exactitude logique (comme tout système) :

- les sorties sont déterminées en fonction des entrées et de l'état interne exactitude temporelle.
- les sorties sont positionnées au bon moment.

II.2.8. Types de temps-réel

- Temps-réel mou: un retard dans l'obtention du résultat n'est pas dramatique (distributeur de billets).
- Temps-réel dur : un retard dans l'obtention du résultat le rend inutile (détection de missile).
- Temps-réel ferme : un retard, s'il arrive très peu souvent, peut être toléré (téléphonie).

La plupart des systèmes temps-réel sont hybrides

II.3. Discussion :

Dans notre vie quotidienne, nous utilisons de plus en plus de dispositifs électroniques pour le divertissement, la navigation GPS et la communication. Ces appareils sont appelés des dispositifs embarqués.

Dans ce chapitre, nous avons présenté brièvement les systèmes embarqués, ensuite nous avons introduit les systèmes monompuces, les systèmes embarqués spécifiques et les systèmes à temps réel, enfin, nous avons abordé l'architecture des systèmes embarqués des différentes générations.

Chapitre III :
Les systèmes à
microprocesseur

III.1. Préambule

Dans ce chapitre, nous décrivons l'architecture de base d'un système à microprocesseur et les principes de son fonctionnement,

Ensuite nous allons présenter d'une manière générale les microcontrôleurs puis nous allons détailler le microcontrôleur STM32 qui est le processeur que nous avons utilisé dans notre application au dernier chapitre.

III.2. Systèmes à microprocesseur

Un système à microprocesseur en digital fait souvent apparaître les fonctions ayant pour rôle le traitement de l'information entre plusieurs signaux d'entrée permettant de générer les signaux de sortie

➤ Architecture d'un système à microprocesseur

Pour traiter une information, un microprocesseur seul ne suffit pas, il faut l'insérer au sein d'un système minimum de traitement programmé de l'information. John Von Neumann est à l'origine d'un modèle de machine universelle de traitement programmé de l'information. La figure suivante montre l'architecture générale d'un tel système.

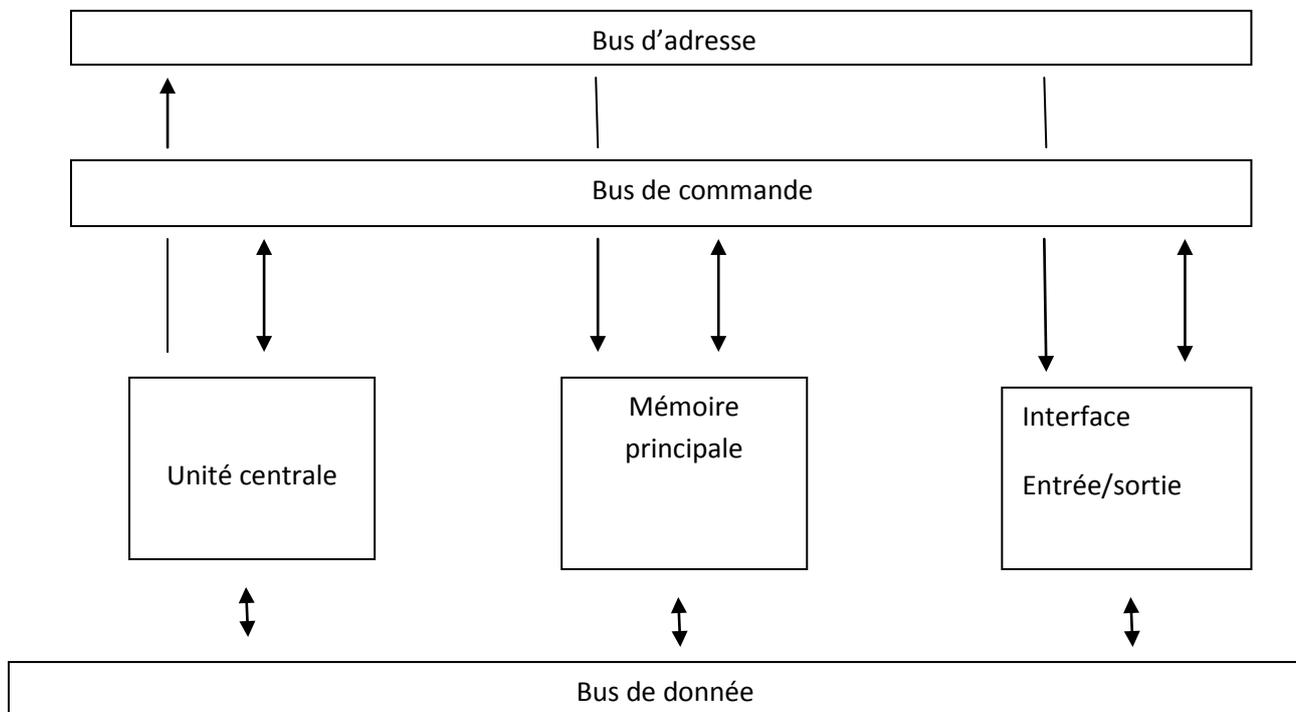


Figure III.1: Architecture d'un système à microprocesseur

Chapitre III: systèmes à microprocesseur

Cette architecture sert de base pour tous les systèmes à microprocesseur. Elle est composée des éléments suivants :

- Unité centrale
- Mémoire principale
- Interface entrée /sortie

III.3. Le microprocesseur

Un microprocesseur est un composant électronique, numérique et programmable.

- **Electronique** : un microprocesseur est un circuit intégré constitué par les composants électroniques de bases.
- **Numérique** : les entrées et les sorties aussi que les commandes au niveau du microprocesseur sont de nature numérique.
- **Programmable** : c'est cette qualité qui rend le microprocesseur intelligent. On peut modifier le comportement d'un système à base de microprocesseur en modifiant le programme qu'il exécute.

Un microprocesseur est construit autour de deux éléments principaux:

- Une unité de commande
- Une unité de traitement

Ces unités sont associées à des registres chargés de stocker les différentes informations à traiter. Ces éléments sont reliés entre eux par des bus interne permettant les échanges d'informations.[2]

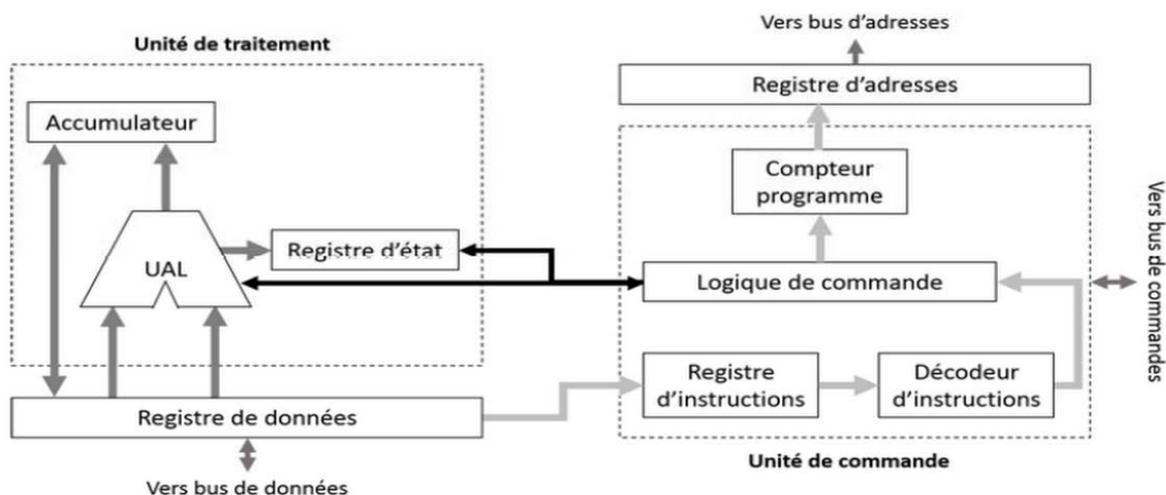


Figure III.2 : Architecture de base d'un microprocesseur

➤ l'unité de commande

Elle permet de séquencer le déroulement des instructions. Elle effectue la recherche en mémoire de l'instruction. Comme chaque instruction est codée sous forme binaire, elle en assure le décodage pour enfin réaliser son exécution puis effectue la préparation de l'instruction suivante :

Pour cela, elle est composée par :

- Le compteur de programme
- Le registre d'instruction et le décodeur d'instruction
- Bloc logique de commande (ou séquenceur)

➤ l'unité de traitement

C'est le cœur du microprocesseur. Elle regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions :

- **L'unité Arithmétique et Logique UAL**

Elle est composée de circuits logiques tels que les additionneurs, soustracteurs, comparateurs logiques...etc., afin d'effectuer les calculs et les opérations logiques des différentes instructions à exécuter, les données à traiter se présentent aux entrées de l'UAL, sont traitées, puis le résultat est fourni en sortie et généralement stocké dans un registre dit accumulateur. Les informations qui concernent l'opération sont envoyées vers le registre d'état. Le schéma suivant montre l'UAL ainsi que ses entrées et ses sorties :[2]

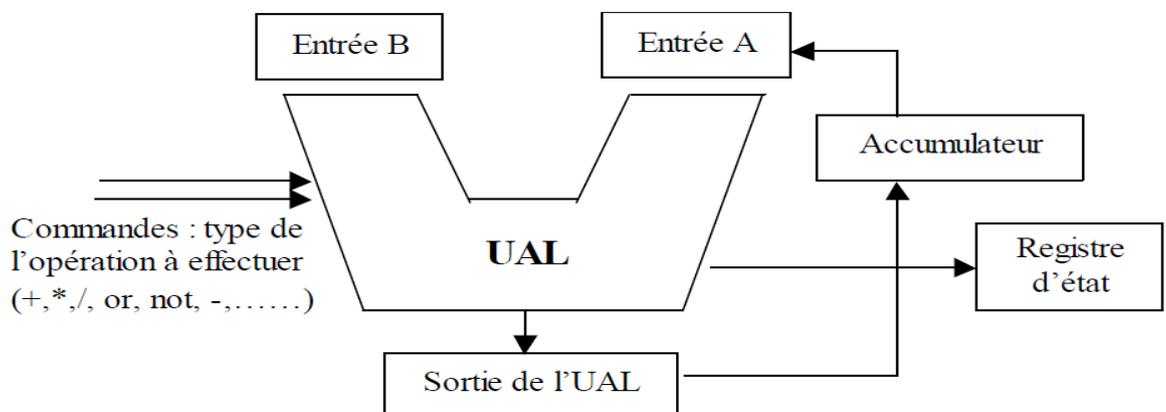


Figure III.3 : Schéma de l'unité arithmétique et logique

Chapitre III: systèmes à microprocesseur

- **Le registre d'état** : est généralement composé de 8 bits à considérer individuellement.

Chacun de ces bits est un indicateur dont l'état dépend du résultat de la dernière opération effectuée par l'UAL. On les appelle indicateur d'état ou flag ou drapeaux.

- **Les accumulateurs** : sont des registres de travail qui servent à stocker un opérande au début d'une opération arithmétique et le résultat à la fin de l'opération.

III.4. les mémoires

Une mémoire est un circuit à semi-conducteur permettant d'enregistrer, de conserver et de restituer des informations (instructions et variables). C'est cette capacité de mémorisation qui explique la polyvalence des systèmes numériques et leur adaptabilité à de nombreuses situations. Les informations peuvent être écrites ou lues. Il y a écriture lorsqu'on enregistre des informations en mémoire, lecture lorsqu'on récupère des informations précédemment enregistrées. [2]

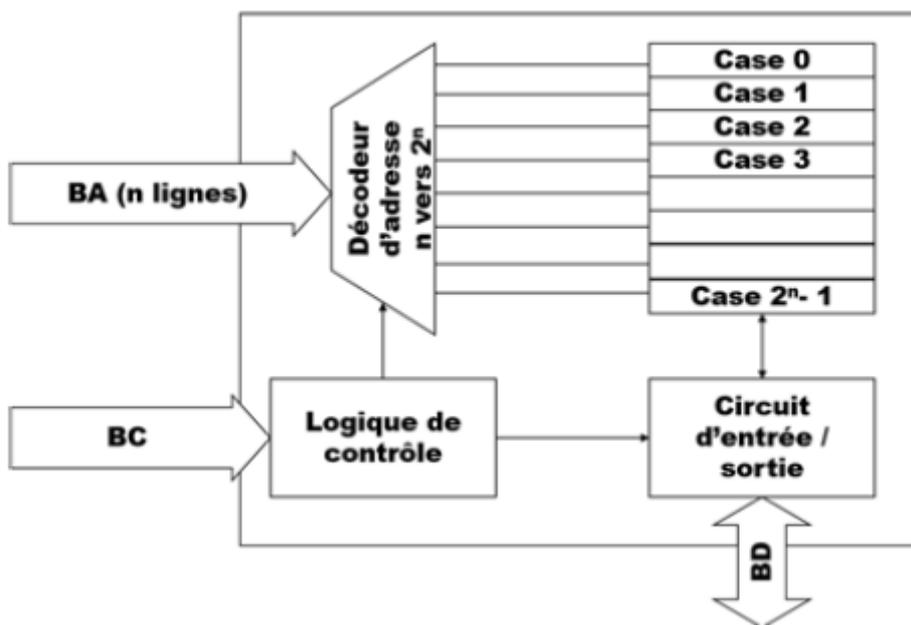


Figure III.4 : Structure d'une mémoire

Chapitre III: systèmes à microprocesseur

Avec une adresse de n bits il est possible de référencer au plus 2^n cases mémoire. Chaque case est remplie par un mot de données (sa longueur m est toujours une puissance de 2). Le nombre de fils d'adresses d'un boîtier mémoire définit donc le nombre de cases mémoire que comprend le boîtier. Le nombre de fils de données définit la taille des données que l'on peut sauvegarder dans chaque case mémoire.

En plus du bus d'adresses et du bus de données, un boîtier mémoire comprend une entrée de commande qui permet de définir le type d'action que l'on effectue avec la mémoire (lecture/écriture) et une entrée de sélection qui permet de mettre les entrées/sorties du boîtier en haute impédance.

III.4.1. Les caractéristiques d'une mémoire : une mémoire est caractérisée principalement de :

- **La capacité :** c'est le nombre total de bits que contient la mémoire. Elle s'exprime aussi souvent en octet.
- **Le format des données :** c'est le nombre de bits que l'on peut mémoriser par case mémoire. On dit aussi que c'est la largeur du mot mémorisable.
- **Le temps d'accès :** c'est le temps qui s'écoule entre l'instant où a été lancée une opération de lecture/écriture en mémoire et l'instant où la première information est disponible sur le bus de données.
 - **Le temps de cycle :** il représente l'intervalle minimum qui doit séparer deux demandes successives de lecture ou d'écriture.
 - **Le débit :** c'est le nombre maximum d'informations lues ou écrites par seconde.
 - **Volatilité :** elle caractérise la permanence des informations dans la mémoire. L'information stockée est volatile si elle risque d'être altérée par un défaut d'alimentation électrique et non volatile dans le cas contraire.

III.4.2. Différents types de mémoires

➤ RAM (Mémoires vives)

Une mémoire vive sert au stockage temporaire des données. Elle doit avoir un temps de cycle très court pour ne pas ralentir le microprocesseur. Les mémoires vives

Chapitre III: systèmes à microprocesseur

sont en général volatiles, elles perdent leur information en cas de coupure d'alimentation. Certaines d'entre elles, ayant une faible consommation, peuvent être rendues non volatiles par l'adjonction d'une batterie.

Il existe deux grandes familles de mémoires RAM (Random Acces Memory : mémoire à accès aléatoire) :

- **Les RAM statiques** : le bit mémoire d'un RAM statique SRAM est composé d'une bascule. Chaque bascule contient entre 4 et 6 transistors.
- **Les RAM dynamiques** : dans la RAM dynamique DRAM, l'information est mémorisée sous forme d'une charge électrique stockée dans un condensateur.
- **ROM (Mémoires mortes)** : Pour certaines applications, il est nécessaire de pouvoir conserver des informations de façon permanente même lorsque l'alimentation électrique est interrompue. On utilise alors les mémoires mortes ou mémoires à lecture seule (ROM : Read Only Memory). Ces mémoires contrairement aux RAM, ne peuvent être que lues. L'inscription des données en mémoire reste possible elle est appelée programmation. Suivant le type de ROM, la méthode de programmation changera. Il existe donc plusieurs types de ROM.
 - **ROM (tout court)** : Elle est programmée par le fabricant et son contenu ne peut plus être ni modifiée, ni effacée par l'utilisateur, donc toute erreur de programmation est fatale.
 - **Avantages** :
 - Non volatile
 - Mémoire rapide
 - **Inconvénients** :
 - Écriture impossible
 - Modification impossible (toute erreur est fatale)
 - Délai de fabrication (3 à 6 semaines)
 - Obligation de grandes quantités en raison du coût élevé qu'entraîne la production du masque et le processus de fabrication.
 - **PROM** : C'est une ROM qui peut être programmée une seule fois par l'utilisateur (programmable ROM). La programmation est réalisée à partir d'un programmeur

Chapitre III: systèmes à microprocesseur

spécifique. Elle est constituée de fusibles qu'on choisit de griller ou non pendant la phase de programmation.

➤ **Avantages :**

- Claquage en quelques minutes
- Coût relativement faible

➤ **Inconvénients :**

- Modification impossible (toute erreur est fatale)

- **L'EPROM ou UV-EPROM :** C'est une ROM qu'on peut effacer en la soumettant à des rayons UV, et donc on peut bien la reprogrammer.

➤ **Avantages :**

- Reprogrammable et non Volatile

➤ **Inconvénients :**

- Impossible de sélectionner une seule cellule à effacer
- Impossible d'effacer la mémoire
- l'écriture est beaucoup plus lente que sur une RAM (environ 1000x).

- **EEPROM (Electrically EPROM) :** c'est une mémoire programmable et effaçable électriquement, elle est faite à base de transistors SAMOS (Stacked Gate Avalanche Injection Mos Transistor).

➤ **Avantages :**

- Comportement d'une RAM non Volatile.
- Programmation et effacement mot par mot possible.

➤ **Inconvénients :**

- Très lente pour une utilisation en RAM.
- Coût de réalisation.

- **FLASH EPROM :** La mémoire Flash s'apparente à la technologie de l'EEPROM. Elle est programmable et effaçable électriquement comme l'EEPROM.

➤ **Avantages :**

- Comportement d'une RAM non Volatile.
- Programmation et effacement mot par mot possible.

➤ **Inconvénients :**

- Très lente pour une utilisation en RAM.

- Coût de réalisation [7]

III.5. Mémoire cache

L'écart de performance entre le microprocesseur et la mémoire ne cesse de s'accroître. En effet, les composants mémoire bénéficient des mêmes progrès technologique que les microprocesseurs mais le décodage des adresses et la lecture/écriture d'une donnée sont des étapes difficiles à accélérer. Ainsi, le temps de cycle processeur décroît plus vite que le temps d'accès mémoire entraînant un goulot d'étranglement. La mémoire n'est plus en mesure de délivrer des informations aussi rapidement que le processeur est capable de les traiter. Il existe donc une latence d'accès entre ces deux organes.

Depuis le début des années 80, une des solutions utilisées pour masquer cette latence est de disposer une mémoire très rapide entre le microprocesseur et la mémoire. Elle est appelée **cache mémoire**. On compense ainsi la faible vitesse relative de la mémoire en permettant au microprocesseur d'acquérir les données à sa vitesse propre. On la réalise à partir de cellule

SRAM de taille réduite (à cause du coût). Sa capacité mémoire est donc très inférieure à celle de la mémoire principale et sa fonction est de stocker les informations les plus récentes ou les plus souvent utilisées par le microprocesseur. Au départ cette mémoire était intégrée en dehors du microprocesseur mais elle fait maintenant partie intégrante du microprocesseur et se décline même sur plusieurs niveaux.

Le principe de cache est très simple : le microprocesseur n'a pas conscience de sa présence et lui envoie toutes ses requêtes comme s'il agissait de la mémoire principale :

- Soit la donnée ou l'instruction requise est présente dans le cache et elle est alors envoyée directement au microprocesseur. On parle de succès de cache.

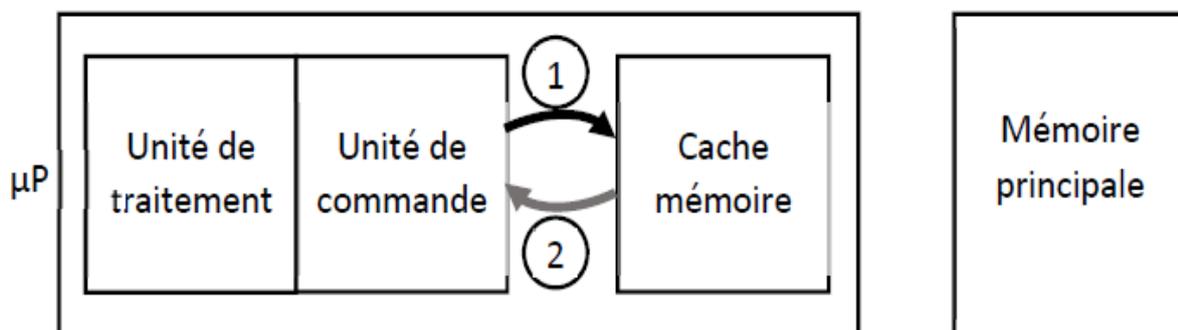


Figure III.5: Succès de la mémoire cache

Chapitre III: systèmes à microprocesseur

- Soit la donnée ou l'instruction n'est pas dans le cache, et le contrôleur de cache envoie alors une requête à la mémoire principale. Une fois l'information récupérée, il la renvoie au microprocesseur tout en la stockant dans le cache. On parle de défaut de cache.

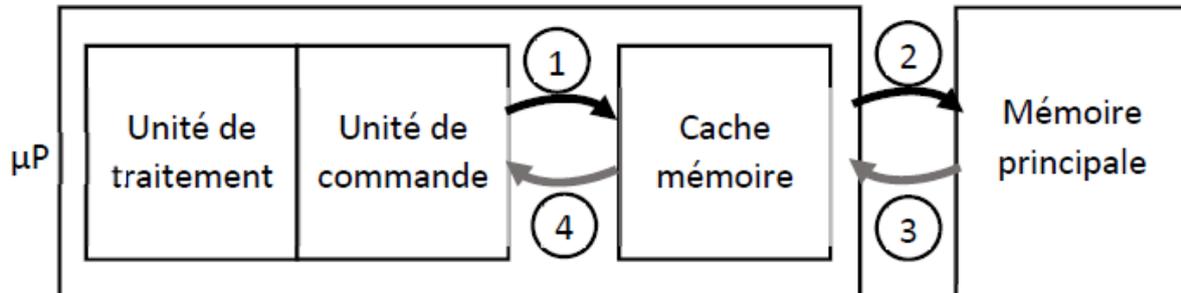


Figure III.6: Défaut de la mémoire cache

III.6. Interfaces d'entrées / sorties

Elles permettent d'assurer la communication entre le microprocesseur et les périphériques tels que : capteur, clavier, moniteur ou afficheur, imprimante, modem

On distingue généralement deux types d'interfaces qui se distinguent par le type de liaison :

III.6.1. liaison parallèle

Dans ce type de liaison, tous les bits d'un mot sont transmis simultanément. Ce type de transmission permet les transferts rapides mais reste limité aux faibles distances de transmissions à cause du nombre important de lignes nécessaires (coût et encombrement) et des problèmes d'interfaces électromagnétiques entre chaque ligne (fiabilité). La transmission est cadencée par une horloge.

III .6.2. Liaison série

Dans ce type de liaison, les bits constitutifs d'un mot sont transmis les uns après les autres sur un seul fil. Les distances de transmission peuvent donc être beaucoup plus importantes mais la vitesse de transmission est plus faible. Sur des distances supérieures à quelques dizaines de mètres, on utilisera des modems aux extrémités de la liaison.

La transmission de données en série peut se concevoir en deux façons :

- Mode synchrone : l'émetteur et le récepteur possèdent une horloge synchronisée qui cadence la transmission. Le flot de données peut être ininterrompu.
- Mode asynchrone : la transmission s'effectue au rythme de la réponse des données. Les caractères envoyés sont encadrés par un signal START et un signal STOP.

III.7. Le microcontrôleur

Les microcontrôleurs sont des circuits logiques qui permettent l'exécution d'un programme dont les actions dépendent de l'état des variables d'environnement du système.

On peut définir un microcontrôleur comme un composant réunissant sur un seul et même silicium un microprocesseur, divers dispositifs d'entrées/sorties et de contrôle d'interruptions ainsi que de la mémoire, notamment pour stocker le programme d'application. Il intègre également un certain nombre de périphériques spécifiques des domaines ciblés (bus série, interface parallèle, convertisseur analogique numérique, ...). Les microcontrôleurs améliorent donc l'intégration et le coût (lié à la conception et à la réalisation) d'un système à base de microprocesseur en rassemblant les éléments essentiels d'un tel système dans un seul circuit intégré. On parle alors de "système sur une puce" (en anglais : "System On chip").

Voici généralement ce que l'on trouve à l'intérieur d'un microcontrôleur :

- Un processeur (C.P.U.),
- Des bus,
- De la mémoire de donnée (RAM et EEPROM),
- De la mémoire programme (ROM, OTPROM, UVPROM ou EEPROM),
- Des interfaces parallèles pour la connexion des entrées / sorties,
- Des interfaces séries (synchrone ou asynchrone) pour le dialogue avec d'autres unités,
- Des timers pour générer ou mesurer des signaux avec une grande précision temporelle.

Le microcontrôleur apparaît donc comme un système extrêmement complet et performant, capable d'accomplir une ou plusieurs tâches très spécifiques, pour lesquelles il a été programmé. Ces tâches peuvent être très diverses, si bien qu'on trouve aujourd'hui des microcontrôleurs presque partout: dans les appareils électroménagers (réfrigérateurs, fours à micro-ondes...), les téléviseurs et magnétoscopes, les téléphones sans fil, les périphériques

Chapitre III: systèmes à microprocesseur

informatiques (imprimantes, scanners...), les voitures (airbags, climatisation, ordinateur de bord, alarme...), les avions et vaisseaux spatiaux, les appareils de mesure ou de contrôle des processus industriels, des stations météorologiques qui permettent la mesure de température, pression, vitesse et direction du vent

Dans notre application nous avons utilisé le microcontrôleur STM32. Les microcontrôleurs STM32 sont développés par la société STMicroelectronics combinant plusieurs caractéristiques telles que la haute performance, le temps réel, la basse tension, le faible coût et consommation, une intégration complète et un développement facile. Actuellement la famille STM32 se divise en deux catégories :

- Les STM32Fxx, F pour fast (rapide) : sont utilisés pour les applications qui donnent priorité à la vitesse d'exécution et dont la consommation en énergie n'est pas une exigence première.
- Les STM32Lxx, L pour low power (faible consommation), sont utilisés dans des domaines où la faible consommation en énergie est prioritaire par rapport à la vitesse d'exécution de l'application.

Les microcontrôleurs STM sont construits autour d'un microprocesseur ARM Cortex-M développé par la société ARM.

III.7.1. ARM Cortex-M :

- **Le cortex A (Application) :** c'est une gamme de processeurs développée pour des tâches nécessitant des calculs complexes.
- **Le cortex M (EMbedded) :** cette gamme est destinée aux microcontrôleurs qui nécessitent une faible consommation d'énergie et un bas prix de vente.
- **Le cortex R (Real-time) :** les processeurs de cette gamme offrent de hautes performances pour les systèmes embarqués temps réel dans lesquels la fiabilité et le déterminisme sont essentiels.

Contrairement aux gammes Cortex A et Cortex R, la gamme Cortex M est un cœur complet devenu un standard dans le domaine des processeurs, il s'appuie sur une architecture système bien définie qui comprend une gestion des interruptions, une horloge système, un système de débogage, et un espace d'adressage mémoire.

L'espace mémoire est divisé en régions distinctes prédéfinies pour le code, les données ; les périphériques et le système. S'appuyant sur une architecture de Harvard. Le Cortex M possède ainsi plusieurs bus, ce qui lui permet d'effectuer des opérations en

Chapitre III: systèmes à microprocesseur

parallèles. Le mode d'accès non aligné aux données lui assure une meilleure utilisation de la mémoire et un accès plus efficace aux registres des périphériques.

Il est important de noter que les processeurs conçus par la société ARM ne sont pas des composants autonomes mais seulement des IP (Intellectual properties) , propriétés intellectuelles).

Ce sont les fabricants, tels que STMicroelectronics, qui réalisent des composants autonomes comme le STM32 en ajoutant au Cortex M de la mémoire et des périphériques après en avoir acheté une licence auprès d'ARM, La figure III.7 montre la relation entre un processeur Cortex M et son cœur Cortex M

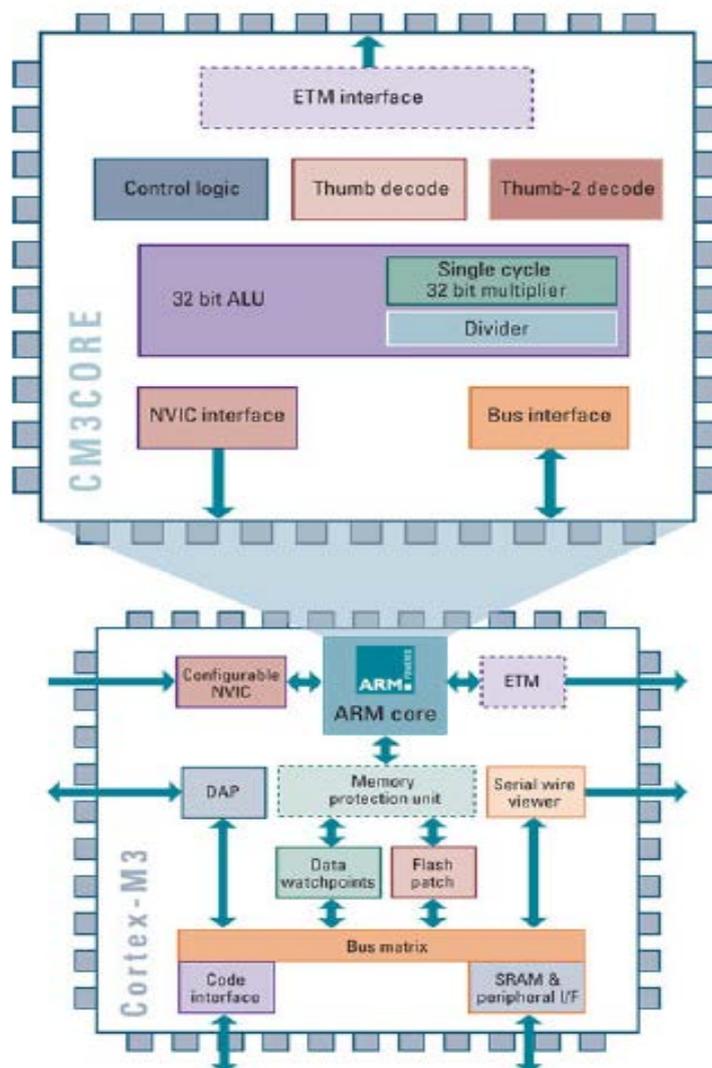


Figure III.7 : la relation entre un processeur Cortex M et son cœur Cortex M

Chapitre III: systèmes à microprocesseur

Un des composants importants du Cortex-M est son NVIC (Nested Vector Interrupt Controller), contrôleur de vecteurs d'interruptions rapprochés), qui présente une structure standard de gestion des interruptions et fournit jusqu'à 240 sources périphériques qui peuvent être chacune dédiée à une interruption. Le temps entre la réception de l'interruption et le début de l'exécution du code correspondant est optimisé en partie grâce à une gestion automatique de la pile, qui se fait par microcode.

A la première vue, les périphériques semblent des classiques des microcontrôleurs : des convertisseurs analogique-numérique, des timers d'usage général, des interfaces I2C, CAN, SPI, USB....., cependant, ces périphériques sont beaucoup plus nombreux et chacun d'eux présente des fonctionnalités supplémentaires.

III.8. Discussion :

Dans ce chapitre nous avons présenté le système à microprocesseur ainsi que son fonctionnement, ensuite nous avons décrit le microprocesseur qui est le composant principal et essentiel de la carte mère et le cœur des systèmes automatiques, il exécute les instructions des programmes, et gère la totalité des procédures et des enchaînements.

Puis nous avons effectué une étude sur les microprocesseurs Cortex M3, qui est entouré par le microcontrôleur STM32, codé sur 32 bits et qui propose néanmoins des modes de veille en vue de réduire la consommation moyenne d'une application.

Chapitre IV:
Implémentation du
microprocesseur ARM dans
une station météorologique

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

IV.1. Préambule

Dans ce chapitre nous avons fait une étude d'une station météorologique. Le processeur utilisé est Cortex-M3 opérant sur des registres de 32 bits, fournit un compromis entre une puissance de calcul appréciable et une consommation réduite ; il propose néanmoins des modes de veille en vue de réduire la consommation moyenne d'une application.

IV.2. Présentation du schéma synoptique de la station météorologique

Une station météorologique est un ensemble de capteurs qui enregistrent et fournissent des mesures physiques et des paramètres météorologiques liés aux variations du climat.

Dans notre cas nous nous sommes fixés comme objectif la réalisation d'une station météorologique équipée d'un capteur de pression, d'une sonde de température, d'un hygromètre, d'un capteur de lumière et d'un microcontrôleur comme le montre la figure suivante :



Figure IV.1 : Vue intérieure d'une station et installation en extérieur

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

Cette application va nous permettre d'aller plus avant dans la découverte de ce microcontrôleur, et pour cela il est nécessaire de bien comprendre et maîtriser les périphériques utilisés. La figure suivante représente le schéma synoptique de la station .

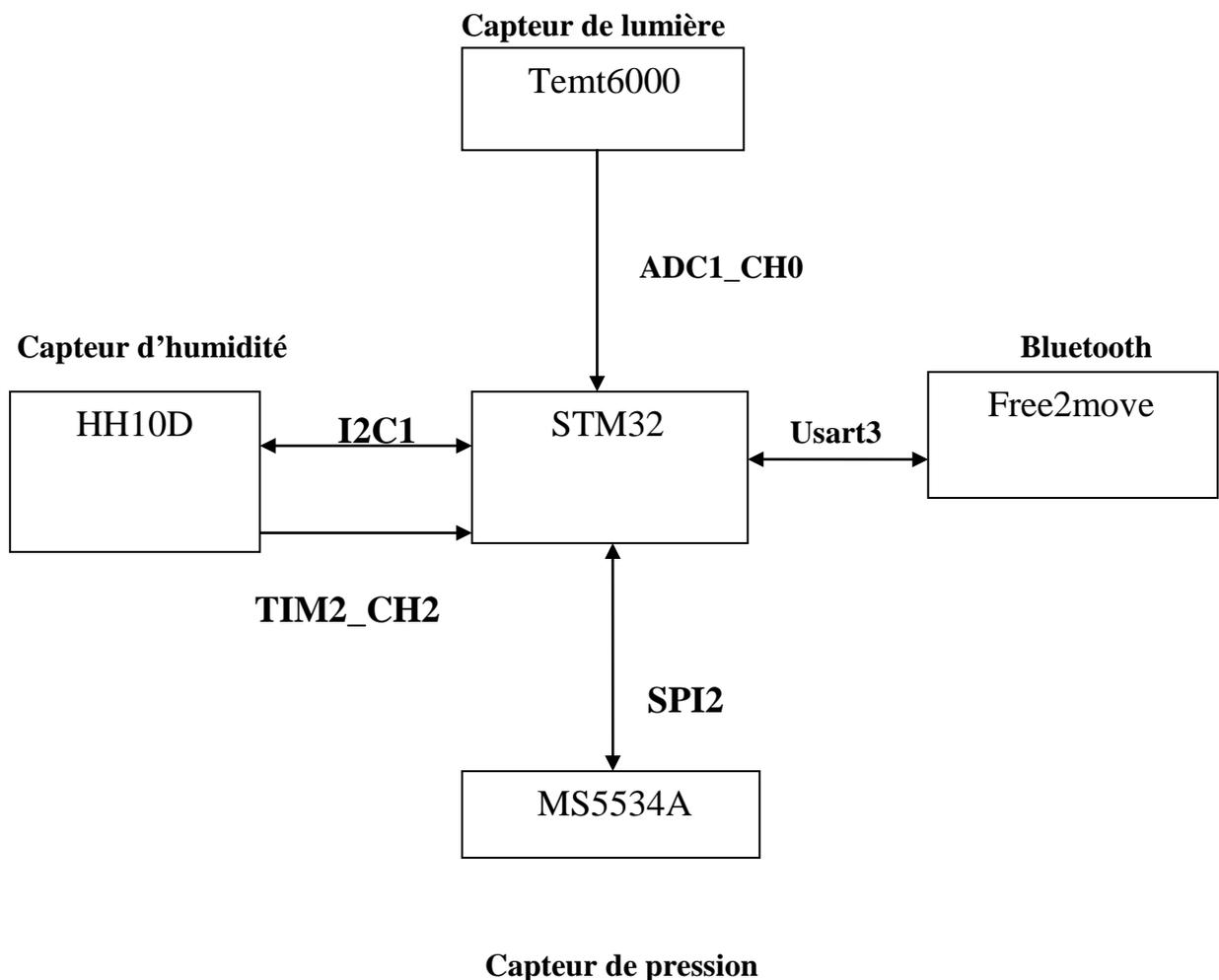


Figure IV.2 : Schéma synoptique d'une station météorologique

IV.3. Présentation des éléments de la station : la station étudiée est composée des capteurs suivants :

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

❖ Capteur d'humidité HH10D

Le HH10D est un système de capteur d'humidité composé d'un capteur d'humidité de type capacitif et d'une puce d'interface convertisseur CF ce capteur fournit un signal périodique proportionnel au pourcentage d'humidité dans l'air.

Le module de capteur d'humidité relative HH10D est équipé d'un capteur d'humidité de type capacitif, d'un condensateur CMOS au convertisseur de fréquence et d'une mémoire EEPROM utilisée pour conserver les facteurs d'étalonnage. Chaque composant est étalonné individuellement, ces informations sont stockées dans une EEPROM accessible en I2C.

En raison des caractéristiques du capteur d'humidité de type condensateur, le système peut réagir très rapidement au changement d'humidité. Chaque capteur est étalonné deux fois dans deux chambres d'humidité précises différentes, deux coefficients uniques liés au capteur sont stockés sur l'EEPROM du module. Les données sont utilisées pour le calcul de l'humidité. L'obtention de la durée de la période d'un signal se fait sur STM32, à l'aide d'un des périphériques timer. Il existe deux modes de capture :

- Le mode input capture qui permet d'obtenir uniquement la durée de la période du signal ;
- Le mode pwm input capture qui fournit en plus la durée de l'état haut ou bas (selon le front choisi pour la période) de ce même signal.

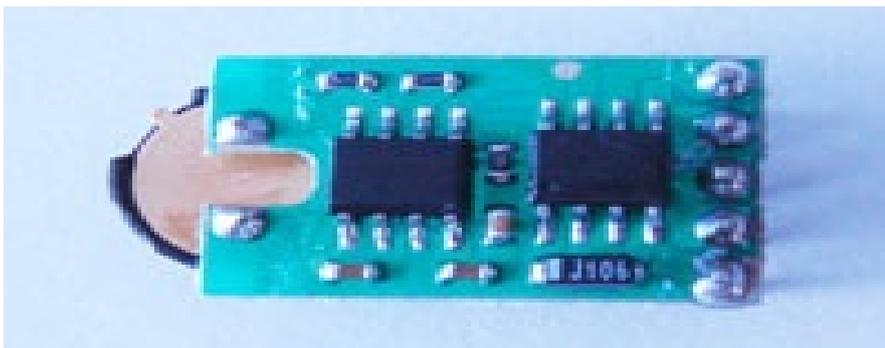


Figure IV.3 : capteur d'humidité

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

➤ Le bus I2C

Le bus I2C (*Inter- Integrated Circuit*) est un bus série synchrone half-duplex permet d'échanger des informations sous forme série . Il est multi-maître et multi-esclaves et n'utilise que deux lignes (et une masse commune) :

- SCL (Serial Clock) : l'horloge commune à tous les équipements,
- SDA (Serial Data) : la ligne portant les données.

Les lignes étant au niveau logique haut quand il n'y a pas de communication, les broches connectées aux lignes SCL et SDA doivent être en open-drain.

Lorsqu'un maître émet sur la ligne SDA, la trame qu'il envoie est décomposée en un bit de démarrage, suivi de l'adresse de l'esclave (7 ou 10 bits) puis d'un bit indiquant si le maître souhaite envoyer une donnée ou en recevoir de l'esclave. À chaque envoi d'un octet de donnée (maître ou esclave) un bit d'acquittement est produit par le récepteur. La communication se termine par un bit de stop.

Dans le cas où il y a plusieurs maîtres, un mécanisme d'arbitrage est ajouté. Ainsi lorsqu'un maître commence à transmettre un message, les autres maîtres ne peuvent pas émettre tant que le bit stop n'est pas arrivé. Les maîtres montrent aussi ce qu'ils émettent afin d'éviter le cas où deux maîtres accèdent simultanément au bus. Ainsi si un maître constate que le niveau est bas alors qu'il a envoyé un bit haut, c'est qu'un autre maître émet. Le maître constatant ce cas interrompt alors immédiatement sa transmission.

Le STM32 dispose d'un périphérique I2C et peut être configuré comme maître et esclave. Le fonctionnement du bus est entièrement pris en considération par le circuit, et ST fournit un ensemble de bibliothèques permettant de facilement le configurer.

❖ Capteur de lumière TEMT6000

Le TEMT6000 est un capteur de lumière ambiante. Ce composant se présente sous la forme d'un transistor NPN (monté en collecteur commun avec une résistance de 10 kΩ) qui fournit une tension proportionnelle à l'intensité lumineuse.

Pour cela, nous allons utiliser l'ADC du STM32, le composant étant connecté sur la broche PA0. Comme son nom l'indique, le capteur de lumière TEMT6000 détectera la luminosité de son environnement. Bien qu'il existe de nombreuses propriétés de la lumière qui

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

peuvent nous aider à catégoriser sa luminosité, le TEMT6000 mesure l'éclairement (mesuré en lux (lx), souvent noté E_v). Ne vous inquiétez pas si l'éclairement est nouveau pour vous, le TEMT6000 est très intuitif à utiliser: plus lumineux = plus de courant, plus sombre = moins de courant. C'est un capteur phototransistor et il peut gérer les tensions des appareils 5V et 3.3V

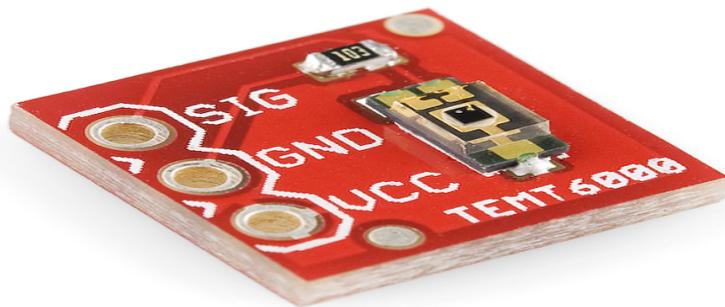


Figure VI.4 : détecteur de lumière ambiante TEMT6000

Pour rendre les mesures de lumière aussi faciles que possible, ce capteur a été conçu dans un circuit diviseur de tension. Le TEMT6000 agit comme l'une des résistances dans le diviseur et à mesure que la lumière le frappe, la tension sur la broche SIG change également. Pour lire cette tension, connectez simplement la broche SIG du TEMT6000 à n'importe quelle broche de conversion analogique-numérique de notre microcontrôleur.

La fonction de chaque broche peut être trouvée dans le tableau suivant :

Symbole	La description
SIG	Tension de sortie du diviseur
GND	GND (0V)
VCC	Tension du capteur (ne dépasse pas 6V)

Tableau IV.1 : Les fonctions des broches du TEMT6000

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

➤ **Le convertisseur ADC**

L'ADC est un convertisseur audio analogique-numérique ce composant ne peut pas être cadencé à plus de 14 MHz, la fréquence d'entrée du résalier étant donnée par APB2 il n'est possible que de faire des divisions par 2, 4, 6, 8.

Ce convertisseur a été conçu pour fournir systématiquement une conversion de qualité de référence dans des conditions « réelles ». Il intègre le système d'horloge.

❖ **Capteur de pression MS5534A**

Est un capteur de pression compensé en température qui communique avec un microcontrôleur au moyen d'un bus synchrone bidirectionnel (deux signaux de données, l'un en provenance et l'autre vers le périphérique, et une horloge). Les commandes à envoyer sont sur 10 bits et les données reçues sont sur 16 bits.

Ce capteur fournit non seulement une information permettant de calculer la pression mais également la température. Afin d'obtenir ces deux informations, il est nécessaire de récupérer un jeu de constantes de calibrage qui permettront ensuite de réaliser les calculs nécessaires pour obtenir des valeurs en °C et en hPa.

La communication peut s'apparenter à du SPI à quelques exceptions près

- Il n'y a pas de Chip-Select, le capteur risque donc de parler alors qu'il n'est pas la cible
- La broche DOUT (MISO) ne passe jamais en haute impédance quand le composant est au repos, imposant donc son potentiel à la ligne. Ceci n'est pas compatible avec la communication multi-esclave
- Le capteur lit les données sur le front montant de l'horloge et le microcontrôleur doit lire les données issues du capteur sur le front descendant. Il n'existe pas de mode SPI adapté à cette situation ;
- Les commandes à lui envoyer sont codées sur 10 bits. La plupart des microcontrôleurs ne peuvent communiquer qu'en 8 ou 16 bits.

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

➤ **Le bus SPI**

Le bus SPI (serial peripheral Interface) est un bus série assurant une communication en full-duplex synchrone. Il est multipoint avec un maître communiquant avec plusieurs esclaves. La communication est réalisée entre le maître et un esclave, le choix de l'esclave se faisant par une sélection directe (donc sans adressage).

La ligne entre le maître et les esclaves est constituée de quatre fils :

- SCLK (Serial Clock): l'horloge du bus qui est produit par le maître,
- MOSI (Master Out Slave In) : la ligne de donnée du maître vers l'esclave,
- MISO (Master In Slave Out) : la ligne de donnée de l'esclave vers le maître,
- SSn (Slave Select n) : la ligne permettant de sélectionner l'esclave destinataire de la communication.

Le principe de communication du SPI repose sur des registres à décalage présents côté maître et côté esclave. La taille du registre peut être configurée (dans le cas du STM32 par exemple ce registre peut-être de 8 ou 16 bits). À chaque période d'horloge, un bit est transféré du maître vers l'esclave et réciproquement de l'esclave vers le maître. Dans le cas d'un registre de 16 bits, il faut donc 16 périodes d'horloge pour effectuer le transfert des données.

Pour qu'une communication fonctionne, il faut que le maître et l'esclave partage le même mode.

Au niveau d'un STM32, un périphérique est dédié au SPI et est facilement configurable à l'aide de quelques registres. La gestion et la lecture des données est ensuite à gérer au niveau de l'applicatif.

➤ **Configuration du SPI**

En vue de communiquer avec le capteur, il est logiquement nécessaire de mettre en place un

Certain nombre de fonctions pour la configuration et la communication en SPI. Nous allons utiliser arbitrairement SPI2 mais le même code sera applicable à tous les autres SPI disponibles.

➤ **Free2Move**

Free2move, spécialiste de la communication sans fil

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

Les modules d'adaptateur série Bluetooth sont souvent sous-estimés en ce qui concerne la connectivité série RS232. En effet, la communication série par un adaptateur série Bluetooth nous donne la connexion série sans fil la plus économique, la plus efficace, la sûre et la plus fiable. Un adaptateur série Bluetooth offre un moyen pratique et facile de communiquer sans fil avec les périphériques série, qu'ils soient anciens ou nouveaux, cet adaptateur série Bluetooth est une unité complète pour la communication RS232 sans fil avec les imprimantes

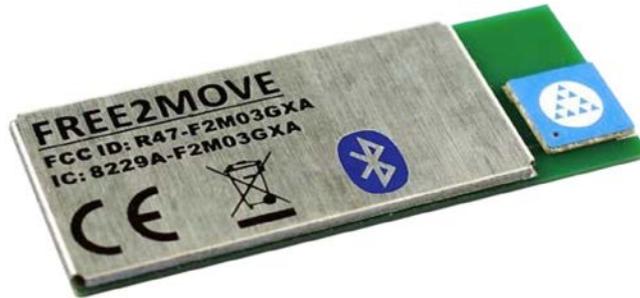


Figure IV.5 : Bluetooth free 2 move

➤ **Liaison série USART**

USART est une abréviation de : Universal Synchronous Asynchronous Receiver Transmitter. C'est un circuit électronique qui permet de mettre en série les octets à transmettre par un modem. Cela signifie que les bits constituant l'octet sont transmis les uns après les autres grâce à deux fils (les bits résultent de la différence de potentiel entre le fil de donnée et le fil de la masse) et non simultanément sur 9 fils (8 fils de donnée et le fil de la masse

➤ **Adaptateur RS-232 à USB**

Le paquet logiciel pour la station météorologique est normalement livré avec le câble RS-232. Si vous souhaitez transmettre les données enregistrées par la station météorologique à un portable ou à un autre mode de stockage mobile, il vous faudra un adaptateur USB (logiciel du driver inclus).

Le STM32 dispose selon les modèles de 4 à 6 USARTs. Sur notre carte, l'USART1 est connecte a un convertisseur USB-série.

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

La configuration de ce périphérique se fait de la façon suivante : Comme tous les périphériques du STM32, il est nécessaire d'activer l'horloge pour l'USART ainsi que pour les broches de communication.

Chaque broche du microcontrôleur dispose de plusieurs fonctions selon les besoins, il est donc nécessaire de configurer les broches



Figure IV.6: Schéma de convertisseur RS232

❖ Les Timers:

- CH Timers

La série CH est une minuterie standard conçue pour un montage sur panneau. Les minuteries CH sont disponibles en plages de 15 secondes, 60 secondes, 3 minutes et 5 minutes et disposent d'un embrayage externe pour permettre une réinitialisation rapide. Ils disposent également d'un interrupteur de commande du moteur pour protéger le moteur à la fin du cycle de temps. Le CH est disponible dans des plages de temps plus longues sur commande spéciale. Les unités permettent une réinitialisation automatique lors de la suppression de l'alimentation et disposent d'un commutateur de charge isolé unipolaire à double jet.



Figure IV.7 : Le Timer série CH

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

- **Le TIM 2**

Est un convertisseur constitue une solution fiable pour de nombreuses application exigeantes telles que les systèmes de transport, les équipements de contrôle industriel.



Figure IV.8: Série TIM2

IV.4. Fonctionnement de la station

Nous avons à notre disposition l'ensemble des pilotes nécessaires à la fois pour l'exploitation du STM32, mais également pour la communication et le traitement des données acquises des capteurs de la station. Après la configuration du STM32, nous configurons l'USART3 sur lequel est connecté le convertisseur Bluetooth-série. Le microcontrôleur STM32 dispose d'un périphérique USART (Full-duplex) avec la possibilité de sélectionner un baud rate, pouvant aller jusqu'à 4,5 Mbps. L'ensemble des composants est ensuite configuré ainsi que le systick pour appliquer une pause entre chaque envoi.

Dans notre boucle infinie, les données sont acquises, formatées et envoyées, toutes les 6 secondes.

Bien entendu, une attente active faite par une fonction Delay n'est pas idéale. Il serait plus judicieux d'exploiter l'un des modes de mise en veille du microcontrôleur afin de réduire la consommation globale de la station

Les mesures météorologiques sont enregistrées à intervalles réguliers et stockées dans L'EEPROM. Les enregistrements sont transférés sur PC via le port série RS232 de la station

Chapitre IV : Etude de l'implémentation du microprocesseur ARM dans une station météorologique

(possibilité d'utiliser un convertisseur USB/Série). Les différents capteurs sont sélectionnés de façon à illustrer l'utilisation des divers périphériques proposés par le STM32. L'acquisition de données se fait par conversion analogique-numérique (TEMT6000), communication numérique synchrone (SPI pour le MS5534A, I2C pour les paramètres du HH10D) et mesure de largeur d'impulsion (HH10D) exploitant les timers. Ces informations sont ensuite transmises pour une liaison sans fil Bluetooth au moyen d'une liaison numérique asynchrone (RS232).

IV.5. Discussion :

Dans ce chapitre, nous avons étudié la station météorologique basée sur le microprocesseur ARM.

Nous avons mis en évidence les capteurs de lumière, d'humidité et de pression, ainsi que les différents bus de liaison.

Conclusion :

Conclusion :

Le travail qui a été effectué dans le cadre de ce mémoire est d'effectuer une étude d'une station météorologique à base du microprocesseur Cortex M3 qui est de type ARM.

Dans un premier temps une présentation générale de l'électronique numérique nous a permis de bien comprendre les notions de base de la logique séquentielle ainsi la logique combinatoire qui sont des éléments essentiels de cette dernière.

Ensuite, nous avons fait une étude sur les systèmes embarqués qui sont aujourd'hui au cœur de la convergence électronique, informatique, télécommunication et réseaux.... Les systèmes embarqués nous entourent et nous assistent au quotidien, ils sont dédiés à une tâche bien précise et se distinguent, tant au niveau des applications, qu'au niveau des performances requises. Leur conception est complexe car elle nécessite de jongler avec des contraintes d'intégration et de fonctionnement (consommation de puissance, surface, débit) souvent très fortes.

Puis nous nous sommes intéressés à étudier les systèmes à microprocesseur, où le microprocesseur est l'élément le plus important. Son apparition a engendré de nombreuses avancées, comme l'augmentation de la vitesse de fonctionnement et de la fiabilité, la réduction de la consommation énergétique....

Nous avons focalisé notre travail sur la réalisation de la station météorologique, où nous avons étudié d'une façon détaillée les différents capteurs de la station à savoir : le capteur de pression et de température MS5534A, le capteur d'humidité HH10D, et le capteur de lumière TEMT6000, ainsi que le microcontrôleur STM32..... Toutefois, pour des raisons de la non disponibilité de certains composants, nous nous sommes limités à la conception théorique de la station.

En perspectives, pour une meilleure caractérisation de l'atmosphère, il serait intéressant de rajouter des capteurs permettant de détecter d'autres paramètres météorologiques, entres autres, les aérosols, la vitesse du vent, le rayonnement solaire...

Références bibliographiques :

Bibliographie

[1] Trends in Embedded-Microprocesseur Design. M. Schlett. Computer Journal. IEEE Computer Society, Août 1998. http://www.cs.utexas.edu/user/dhurger/teaching/cs395ts99/papers/4_embedded.pdf, Août

[2] Antoine Fabrice Jumel. Lionel Morel Tanguy Risset « tanguy.risset@insa-luon.fr » « Conception et programmation de systèmes Embarqués », Juillet

Ouvrages

[3] Alexander NKETSA : « circuit logique programmables mémoires PLD.CPLD et FPGA », paris : Ellipses , 1998

[4] Fabrice Kordon Jérôme Hugues Agusti Canals Alain Dohet « Modélisation et analyse de systèmes embarqués », Paris : Hermes Science publication : Lavoisier, 2013

[5] Francine Krief « Les systèmes embarqués communication, mobilité, sécurité, autonomie » Paris : Hermes : Lavoisier 2008

Sites web

[6] http://www.intel-research.net/Publications/Seattle/062120021245_50.pdf 05/2018

[7] <http://elearn.univ-ouargla.dz>, 08/2018

[8] [http:// www.memioreonlin.com](http://www.memioreonlin.com), 05/2018

[9] [http://www. Jdfineart.com](http://www.Jdfineart.com), 08/2018