

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE



ECOLE NATIONALE POLYTECHNIQUE  
DEPARTEMENT D'ELECTRONIQUE

**Projet de fin d'études**

*EN VUE DE L'OBTENTION DU DIPLOME D'INGENIEUR D'ETAT  
EN ELECTRONIQUE*

**EVALUATION D'UN MICROCONTRÔLEUR EN  
HAUTE TEMPERATURE**

Dirigé par:

Mme L.HAMAMI (ENP)  
M. F.BARBARA (SRPC)

Etudié par:

BOUSSADIA Bilal Adel

**Schlumberger**

*Promotion : Septembre 2006*

# Remerciements

Je remercie dieu le tout puissant qui m'a donné le courage et la volonté de réaliser ce travail.

Je remercie l'entreprise Schlumberger de m'avoir accueillie et permis d'effectuer mon projet.

J'exprime ma profonde gratitude à ma promotrice, Mme *HAMAMI L.*, qui a porté un grand intérêt à mon travail, et qui j'espère ne pas avoir déçu.

J'exprime également ma profonde gratitude à M. *BARBARA F.* pour m'avoir soigneusement orienté durant mon stage.

Je tiens aussi à exprimer toute ma reconnaissance et mes sincères remerciements à tout ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire, et tout particulièrement M. *KOUIDER K.*, M. *GARANDO L.*, M. *SHITIKOV V.*, et Mme *KELLER N.*

# *DEDICACES*

Je dédie ce travail à :

Mes très chers grands-parents : Mémé et Pépé, qui ont toujours été présents dans toutes les étapes de ma vie.

Mes très chers parents, qui ont fait de moi ce que je suis aujourd'hui.

Mes beaux parents, qui me considèrent comme leur fils.

Mes adorables sœurs : Chahrazed, Fouzia, Yasmine, Yasmina et Selma.

Mon très cher frère Toufik.

Mes oncles : Samir, Redouan, Zoubir, et Madjid

Mes tentes : Malika, Saliha, Lila, Salima, Zahida et Faiza

Mes cousins et mes cousines : Salim et Mounia ainsi que l'adorable Rayane, Farid et Mounia, Yacine et Lila, Nabil et Souhila, Riad et Zahida ainsi que Racim, Yanis et Nel, Kamel et Radia, Mourad, Fouzi, Amel, Lounis, Lydia, Sarah et Wassim.

# SOMMAIRE

▪ INTRODUCTION.....	1
▪ CHAPITRE I : PRESENTATION DE L'ENTREPRISE.....	2
I.1 LE GROUPE SCHLUMBERGER.....	3
I.2 SCHLUMBERGER OILFIELD SERVICES.....	3
I.3 SCHLUMBERGER RIBOUD PRODUCT CENTER (SRPC).....	5
I.3.1 L'organisation du centre technologique SRPC.....	6
I.3.2 Le personnel.....	6
I.3.3 La mission du centre.....	6
I.3.4 Les produits et services.....	7
▪ CHAPITRE II : INITIATION À L'ELECTRONIQUE HAUTE TEMPERATURE.....	8
II.1 QU'EST-CE QUE LA HAUTE TEMPERATURE.....	9
II.2 DEFINITION DE L'ELECTRONIQUE HAUTE TEMPERATURE.....	9
II.3 EFFETS DE LA TEMPERATURE SUR LES COMPOSANTS ELECTRONIQUES.....	9
II.3.1 Vieillissement accéléré des composants électroniques : Loi d'Arrhenius.....	10
II.3.2 Déformations thermomécaniques .....	11
II.3.3 La diffusion .....	12
II.3.4 Le latch-up .....	13
II.3.5 L'électromigration.....	13
II.4 L'APPROCHE SCHLUMBERGER .....	16
▪ CHAPITRE III : PLAN DE TRAVAIL ET SELECTION DU MICROCONTRÔLEUR À EVALUER.....	19
III.1 PLAN DE TRAVAIL.....	20
III.1.1 Partie sélection du microcontrôleur (partie théorique) .....	21
III.1.2 Evaluation du microcontrôleur choisi en haute température (partie pratique)...	21
III.2 SELECTION DU MICROCONTRÔLEUR A EVALUER.....	21
▪ CHAPITRE IV : ETUDE DU MICROCONTRÔLEUR MSC1201 Y3.....	27
IV.1 DESCRIPTION ET SCHEMA BLOC.....	28
IV.2 LE CONVERTISSEUR ANALOGIQUE NUMERIQUE $\Sigma\Delta$ 24 bits.....	29
IV.2.1 Structure .....	29
IV.2.2 Registres de configuration et récupération des résultats .....	34
IV.3 LA SOURCE DE COURANT DIGITALE IDAC .....	35
IV.4 LES MEMOIRES.....	35
IV.5 LES PORTS DE COMMUNICATION.....	40
IV.5.1 L' USART (Universal Synchronous / Asynchronous Receiver Transmitter)....	40
IV.5.2 Les ports d'entrées sorties P1 (Port 1) et P3 (Port 3) .....	41
IV.6 LES HORLOGES .....	43
IV.7 LES TIMERS .....	45
IV.8 LES INTERRUPTIONS .....	47
▪ CHAPITRE V : PRESENTATION DE LA CARTE D'EVALUATION .....	51
V.1 PRESENTATION DE LA CARTE D'EVALUATION MSC1201EVM.....	52
V.1.1 Description matérielle.....	52
V.1.2 Description logicielle.....	53

▪ <b>CHAPITRE VI : EVALUATION DU MSC1201Y3 EN HAUTE TEMPERATURE.....</b>	<b>57</b>
VI.1 PROCÉDÉ D’EVALUATION ET MESURES A EFFECTUER.....	58
VI.1.1 Problématique .....	58
VI.1.2 Solutions .....	58
VI.2 ÉVALUATION DU CONVERTISSEUR ANALOGIQUE NUMERIQUE.....	60
VI.2.1 Méthode d’évaluation .....	60
VI.2.2 Mesures á effectuer.....	61
VI.2.3 Travail préliminaire.....	62
VI.2.4 Résultats des mesures effectuées.....	67
VI.2.5 Analyse des résultats .....	69
VI.2.6 Influence de la température sur la référence interne de l’ADC.....	70
VI.2.7 Conclusion.....	73
VI.3 EVALUATION DE LA SOURCE DE COURANT DIGITALE IDAC EN HAUTE TEMPERATURE.....	73
VI.3.1 Méthode d’évaluation .....	73
VI.3.2 Mesures à effectuer .....	73
VI.3.3 Résultats des mesures effectuées .....	73
VI.3.4 Analyse et discussion des résultats.....	74
VI.3.5 Conclusion .....	74
VI.4 EVALUATION DE LA MEMOIRE FLASH EN HAUTE TEMPERATURE .....	74
VI.4.1 Méthode d’évaluation .....	74
VI.4.2 Tests à effectuer.....	75
VI.4.3 Résultats des tests effectués .....	75
VI.4.4 Analyse et discussion des résultats .....	76
VI.4.5 Conclusion .....	77
VI.5 EVALUATION DE L’HORLOGE INTERNE .....	77
VI.5.1 Méthode d’évaluation .....	77
VI.5.2 Mesures à effectuer.....	77
VI.5.3 Résultats des mesures effectuées .....	77
VI.5.4 Analyse et discussion des résultats .....	79
VI.5.5 Conclusion .....	79
VI.6 POBLÈMES RENCONTRÉS DURANT L’ÉVALUATION DU MSC1201Y3.....	79
▪ <b>CONCLUSION GENERALE.....</b>	<b>81</b>

## LISTE DES TABLES

<b>Tableau II.1</b> : Emf/A en fonction de la température.....	15
<b>Tableau II.2</b> : caractéristiques des segments Schlumberger concernés par la haute Température.....	16
<b>Tableau IV.1</b> : Configuration du multiplexeur des entrées analogiques.....	31
<b>Tableau IV.2</b> : Sélection du PGA.....	31
<b>Tableau IV.3</b> : Succession des filtres pour le mode AUTO.....	33
<b>Tableau IV.4</b> : sélection des filtres numériques.....	34
<b>Tableau IV.5</b> : Sélection du mode de calibration.....	34
<b>Tableau IV.6</b> : Partage de la mémoire flash entre Mémoire Programme (PM) et Mémoire Données (DM).....	36
<b>Tableau IV.7</b> : Sélection de l'alimentation du Brownout.....	37
<b>Tableau IV.8</b> : Sélection de l'horloge.....	37
<b>Tableau IV.9</b> : Boot ROM Routines du Msc1201.....	39
<b>Tableau IV.10</b> : Sélection du mode de fonctionnement du Port 0.....	40
<b>Tableau IV.11</b> : Configuration des entrées / sorties.....	41
<b>Tableau IV.12</b> : choix de la grandeur de sortie sur la broche P3.6.....	43
<b>Tableau IV.13</b> : Choix du mode de division de la Clock.....	44
<b>Tableau IV.14</b> : Programmation du diviseur d'horloge.....	45
<b>Tableau IV.15</b> : Modes de fonctionnement des timers.....	46
<b>Tableau IV.16</b> : Les Interruptions du MSC1201Y3.....	47
<b>Tableau VI.1</b> : Différentes configurations de l'ADC.....	61
<b>Tableau VI.2</b> : Valeurs à convertir pour chaque PGA.....	61
<b>Tableau VI.3</b> : ENOB pour des mesures de la valeur 2.0 V à température ambiante (25 °C).....	67
<b>Tableau VI.4</b> : Effet de la température sur la valeur moyenne des mesures en référence interne..	69
<b>Tableau VI.5</b> : Influence de la température sur la référence interne.....	70
<b>Tableau VI.6</b> : Influence de la température sur la mémoire flash.....	76
<b>Tableau VI.7</b> : Influence de la température sur l'horloge interne.....	78

## LISTE DES FIGURES

<b>Figure I.1</b> : Les Geomarkets de Schlumberger Oilfield Services.....	3
<b>Figure I.2</b> : répartition des tâches par segment de Schlumberger Oilfield Services.....	4
<b>Figure I.3</b> : Organisation du centre technologique SRPC.....	5
<b>Figure I.4</b> : Répartition du personnel à SRPC.....	6
<b>Figure II.1</b> : Classification des gammes de températures pour les circuits électroniques.....	9
<b>Figure II.2</b> : Evolution de la courbe d'âge d'un composant électronique en fonction de la température.....	10
<b>Figure II.3</b> : Variation du CTE en fonction de la température.....	11
<b>Figure II.4</b> : Effet de l'expansion thermique sur deux matériaux liés.....	12
<b>Figure II.6</b> : phénomène de latch-up en haute température pour un inverseur CMOS.....	13
<b>Figure II.7</b> : Créations de Voids et Hillocks par effet d'électromigration.....	14
<b>Figure II.9</b> : Effet de la température sur l'électromigration.....	15
<b>Figure II.10</b> : Fiabilité des technologies choisies par Schlumberger en fonction de la température.....	16
<b>Figure II.11</b> : Choix d'un composant électronique selon l'approche Schlumberger.....	18
<b>Figure III.1</b> : Plan du projet .....	20
<b>Figure IV.1</b> : Schéma block du Msc1201Y3.....	28
<b>Figure IV.2</b> : Package et dimensions du Msc1201Y3.....	29
<b>Figure IV.3</b> : Structure de l'ADC.....	30
<b>Figure IV.4</b> : ADC Input Multiplexer.....	30
<b>Figure IV.5</b> : Filtres Numériques.....	33
<b>Figure IV.6</b> : Réponses fréquentielles des filtres numériques.....	33
<b>Figure IV.7</b> : Partage de la SRAM.....	38
<b>Figure IV.8</b> : Contenu de la Direct RAM.....	38
<b>Figure IV.9</b> : Memory Map.....	39
<b>Figure IV.10</b> : Fréquences présentes au sein du microcontrôleur.....	44
<b>Figure V.1</b> : Schéma de la carte d'évaluation MSC1201EVM.....	53
<b>Figure V.2</b> : Environnement du Keil Uvision 3.....	54
<b>Figure V.3</b> : Modes de fonctionnement de TI downloader.....	55
<b>Figure VI.1</b> : Matériel utilisé pour les mesures en haute température.....	59
<b>Figure VI.2</b> : Schéma de principe des mesures en haute température.....	59
<b>Figure VI.3</b> : Interface graphique (Version Démo).....	62
<b>Figure VI.4</b> : Nouvelle interface graphique.....	63
<b>Figure VI.5</b> : Infogramme représentant le programme coté microcontrôleur.....	64
<b>Figure VI.6</b> : Infogramme représentant le programme coté ordinateur.....	66
<b>Figure VI.7</b> : Histogrammes relatifs à différentes mesures de la valeur 2.0 V à température Ambiante.....	68
<b>Figure VI.8</b> : Valeur Théorique de l'ENOB (datasheet).....	68
<b>Figure VI.9</b> : Graphiques représentant les variations de la référence interne en fonction de la température.....	70
<b>Figure VI.10</b> : Visualisation de la référence interne sur l'oscilloscope à différentes Températures.....	72
<b>Figure VI.11</b> : Mesures de l'horloge interne prises à différentes températures.....	78
<b>Figure VI.12</b> : Caractéristique FSYS = f(T) pour le microcontrôleur MSC1201Y3.....	78
<b>Figure VI.13</b> : Dégradation des cartes d'évaluation sous l'effet de la chaleur.....	79

## LISTE DES ABREVIATIONS

°C : Degrés Celsius  
°K : Degré Kelvin  
ACIA:Asynchronous Communication Interface Adapter  
ADC: Analog Discrete Converter  
CAN: Controller Area Network  
CDIP/CERDIP:Ceramic Dual in-Line Package  
CISC:Complex Instruction Set Computer  
CLCC:Ceramic Leadless Chip Carrier  
CMOS : Complementary Metal Oxyde Semiconductor  
CPLD:Complex Programmable Logic Device  
CSI : Chip Select Input  
CSI: Chip Select Input  
CTE: Coefficient Thermique d'Expansion  
DAC: Discrete Analog Converter  
DIP:Dual in-Line Package  
DMA: Direct Memory Access  
DNL: Differential NON-LINEARITY  
Emf : Electromigration Mass Flux  
ENOB:Effective Number of Bits  
EUSART:Enable Universal Synchronous Asynchronous Receiver Transmitter  
eV : electron Volt  
FPGA:Field Programmable Gate Array  
FPLGA:Fine Pitch Land Grid Array  
HT: Haute Temperature (high temperature)  
ICSP:In-Circuit Serial Programming  
IDAC: Current Discrete Analog Converter  
INL: Integral Non-Linearity  
KSPS:kilo. Samples Per Second  
LBGA:Low-profile Ball Grid Array  
LFBGA:Low-profile Fine-pitch Ball Grid Array  
LFCSP:Lead Frame Chip Scale Package  
LQFP:Lead, Low Profile Quad Flat Package  
MCU:MicroController Unit  
MIPS: Million Instructions Per Second  
MQFP:Metric Plastic Quad Flat Package  
MTTF: Mean Time To Fail  
PDIP: Plastic Dual in-Line Package  
PGA:Programmable Gain Amplifier  
PGTQFP:Plastic Thin Quad Flat Package  
PLA: Programmable Logic Array  
PLCC:Plastic Leaded Chip Carrier  
PQFP:plastic Quad Flat Pack  
PSM: Power Supply Monitorig  
PTQFP:Plastic Thin Quad Flat Package  
PTSSOP: Plastic Thin Shrink Small Outline Package  
QFN:Quad Flat No-lead  
RAM: Random Access Memory  
RISC:Reduced Instruction Set Computer



ROM: Read Only Memory  
RTC: Real Time Clock  
RTI: Real Time Interrupt  
RTT: Real Time Timer  
SCI: Serial Communication Interface  
SDIP:Shrink Dual in-Line Package  
SIO : Serial Input Output Port  
SOIC:Small Outline IC  
SOP:Small Outline Package  
SOT:Small Outline Transistor  
SPI: Serial Peripheral Interface  
SPI:Serial Peripheral Interface  
SSOP:Shrink Small Outline Package  
TQFP:Thin Quad Flat Package  
TSSOP:Thin Shrink Small Outline Package  
TVSOP:Thin Very Small Outline Package  
UART:Universal Asynchronous Receiver Transmitter  
USART:Universal Synchronous Asynchronous Receiver Transmitter  
USB: Universal Serial Bus  
WDT: WatchDog Timer

## **INTRODUCTION**

Ce document fait office de projet de fin d'étude pour l'obtention du diplôme d'ingénieur en électronique ; il a été établi à l'issue d'un stage de six mois au sein du centre technologique Schlumberger Riboud Product Center.

Le projet consiste en la qualification d'un microcontrôleur en haute température. Pour ce faire, l'étude se composera de six chapitres, tout d'abord, nous commençons par une brève présentation de l'entreprise Schlumberger, puis nous introduisons la notion de l'électronique haute température et les différents phénomènes qui rendent difficile le développement d'une application électronique en haute température et à la méthodologie suivie par Schlumberger pour remédier au problème, suite à cela nous sélectionnerons un microcontrôleur qui est l'aboutissement d'une étude comparative selon les performances entre différents fabricants, nous continuerons par une description détaillée du microcontrôleur ainsi que les différents équipements utilisés durant cette étude, et enfin nous procéderons à différents tests visant à évaluer les fonctions principales du microcontrôleur.

Notre objectif est d'aboutir à une configuration donnant un fonctionnement optimal du microcontrôleur en haute température.

# CHAPITRE I

## PRESENTATION DE L'ENTREPRISE

## **I.1 LE GROUPE SCHLUMBERGER :**

Tout au long de son histoire, Schlumberger a placé l'innovation au cœur de tous les projets. D'abord spécialisée dans l'exploration pétrolière, la société s'est vite diversifiée pour être présente dans des secteurs comme l'électronique, la gestion des systèmes d'eau, d'air et d'électricité, ainsi que dans les télécommunications et les systèmes d'information.

Cette diversification s'est faite parallèlement à une mondialisation importante et une amélioration des techniques existantes.

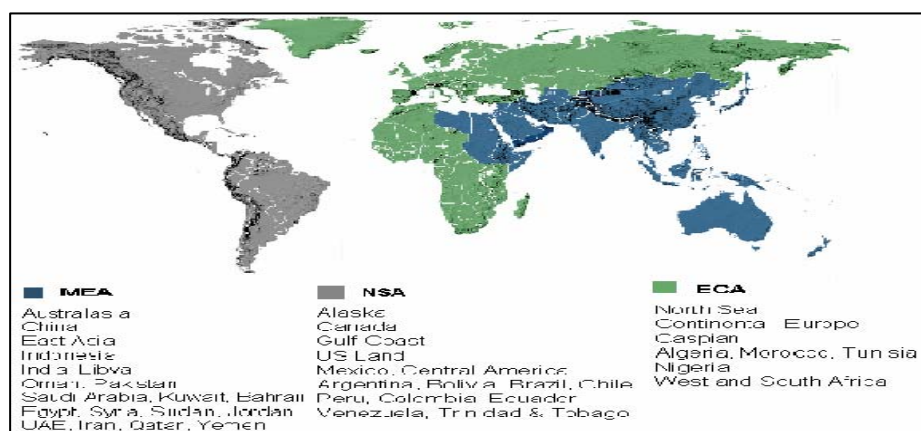
La société Schlumberger est née en 1926. elle est le fruit de la découverte par deux frères alsaciens, Conrad et Marcel Schlumberger, d'un nouveau système de mesures permettant de décrire la nature du sous-sol. Ils ont été les premiers à montrer que des mesures de résistance électrique effectuées en surface permettaient de détecter la présence de dépôts minéraux. En 1927, ils réussissent à prouver la possibilité de la mise en œuvre de ces mêmes principes au fond d'un puits. Ainsi, en utilisant des instruments de mesure à l'extrémité d'un câble conducteur « *wire-line* », ils ont pu identifier les couches géologiques contenant du pétrole. La technique du « *logging* » était née et pendant près de trente ans, la société s'est exclusivement consacrée à cette activité.

Aujourd'hui Schlumberger est une multinationale dotée de deux sièges (Paris et New York) et de sept centres de profit. Le groupe Schlumberger emploie près de 50000 personnes de 140 nationalités différentes dans plus de 100 pays.

## **I.2 SCHLUMBERGER OILFIELD SERVICES :**

Cette branche constitue l'activité historique de Schlumberger et représente environ les deux tiers du chiffre d'affaire du groupe. Elle est le principal fournisseur de services et de technologies à l'industrie pétrolière internationale.

Ce pôle gère ses activités au travers de 27 « Geomarkets » regroupés en trois zones géographiques : Amérique du Nord / Amérique Latine (NSA), Europe / Afrique de l'ouest (ESA) et Moyen Orient / Asie (MEA) comme le montre la *Figure I.1*.

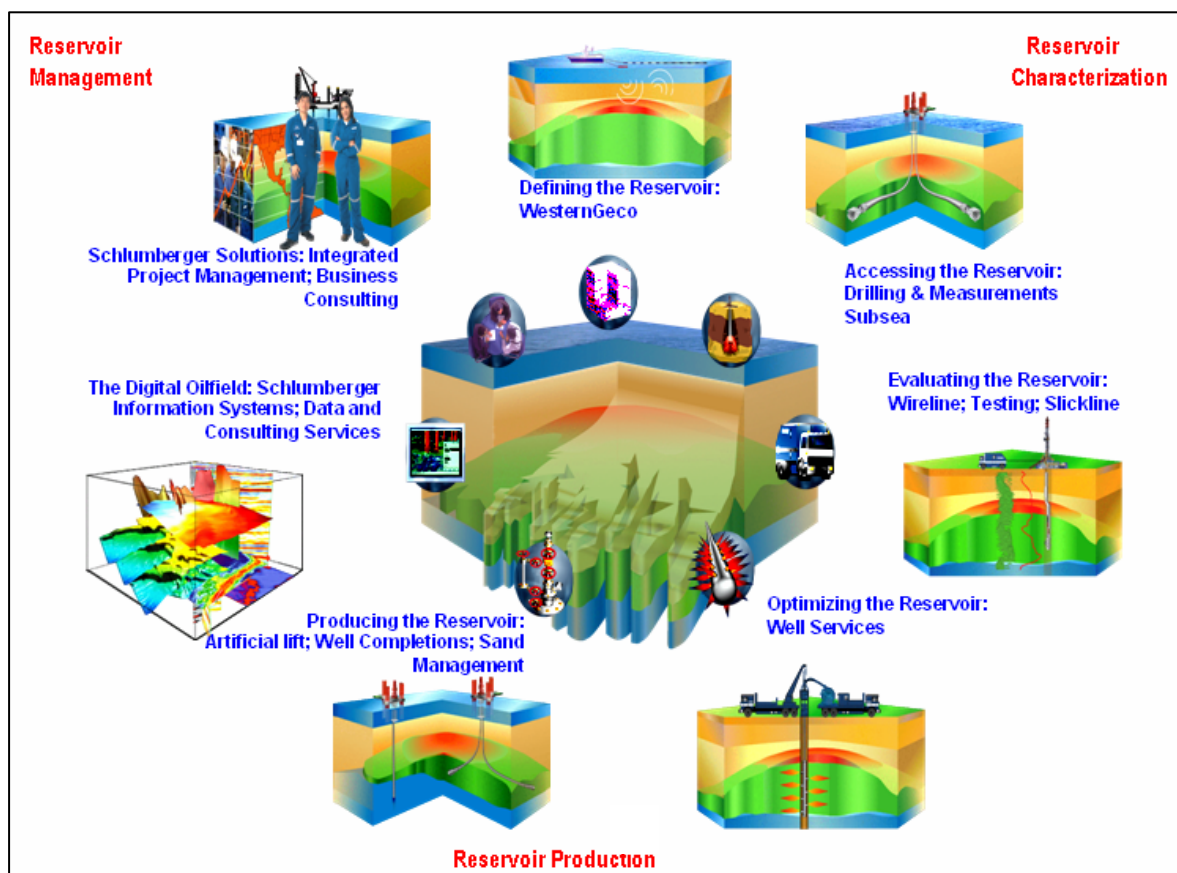


*Figure I.1 : Les Geomarkets de Schlumberger Oilfield Services.*

Il s'organise autour de sept segments technologiques :

- « WesternGeco » service d'acquisition et de traitement de données sismiques.
- « Wire-line » service fournissant l'information nécessaire à la caractérisation du réservoir, à la planification et au contrôle de la construction des puits ainsi qu'à l'évaluation de la production
- « drilling and measurement » service de forages dirigés de mesures et de « logging » en cours de forage.
- « Well services » optimisation de la production, services de cimentation et de stimulation de puits.
- « Well completion and productivity » services d'essais de complétion et de production et d'activation des puits.
- « integrated project management » conseil, gestion des projets et études d'ingénierie mobilisant l'expertise des autres segments technologiques.
- « Schlumberger information solution » solutions stratégiques intégrées comprenant des services de gestion de l'information, des technologies de l'information et une gamme complète de services d'experts.

Ces segments assument totalement tous les services pétroliers depuis l'inspection (recherche de pétrole) jusqu'à la production et la maintenance. La *Figure I.2* résume les services qu'offre chaque segment de Schlumberger Oilfield Services.



*Figure I.2 : Répartition des tâches par segment de Schlumberger Oilfield Services.*

Les segments de Schlumberger Oilfield Services bénéficient de l'assistance de treize centres technologiques et de cinq centres de recherche parmi lesquelles se trouve le centre technologique de Clamart. C'est dans ce centre que j'ai eu la chance de réaliser ce projet.

### **I.3 SCHLUMBERGER RIBOUD PRODUCT CENTER (SRPC):**

Le SRPC situé à Clamart est une filiale Schlumberger qui regroupe trois sociétés :

- Etudes et productions Schlumberger (EPS)
- Etudes et fabrication Dowell Schlumberger (EFDS)
- Services Techniques Schlumberger (STS)

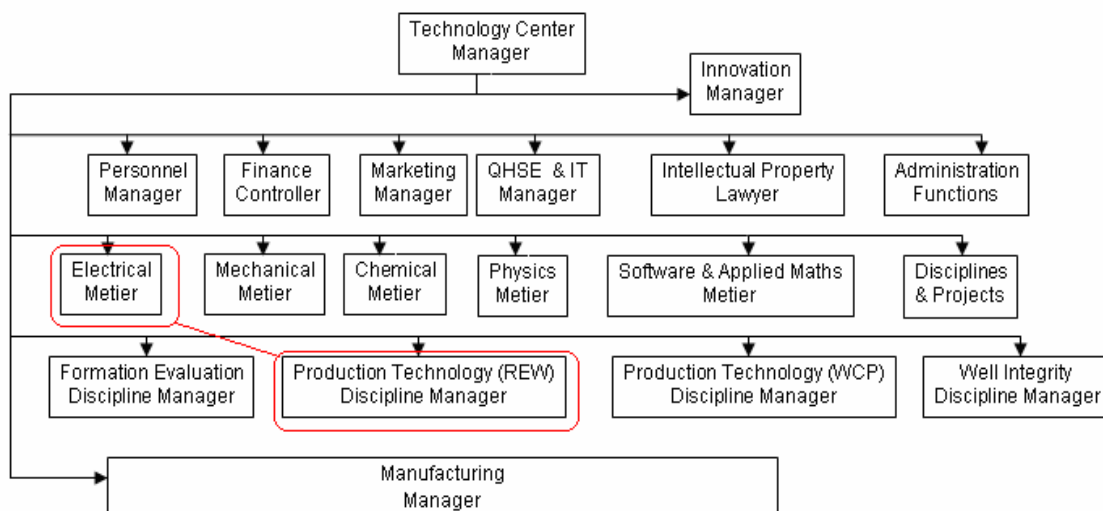
Ces trois centres appartiennent au secteur « Oilfield services » et constituent l'un des centres de production de la branche « Wireline and testing », créée en 1959, EPS Clamart a ensuite fusionné avec le site de Melun en 1989. Entre 1991 et 1994, le site est modernisé et rebaptisé SRPC, en hommage à Jean Riboud, qui fut désigné dirigeant du groupe en 1966 par les frères Schlumberger.

#### **I.3.1 L'organisation d'un centre technologique SRPC:**

Comme le montre l'organigramme ci-dessous le SRPC est constitué d'une partie Etudes et une partie Production.

Dans chaque partie des équipes multidisciplinaires sont formées en fonction des projets en cours : les membres de ces équipes appartiennent à différents métiers : **Mécanique, électrique, physique/chimie** et **software**, ils seront réaffectés à d'autres projets, une fois le travail accompli sur le projet en question. Le chef de métier a un rôle de responsable fonctionnel et de suivi de carrière de personnes appartenant à son métier. En plus de ces départements, il existe des groupes technologiques pour chaque métier.

La partie production travaille sans distinction de segments. Elle se charge de fabriquer, monter, tester et évaluer les produits qui ont été conçus par la partie Etudes. Elle va également piloter la sous-traitance pour tous les segments.



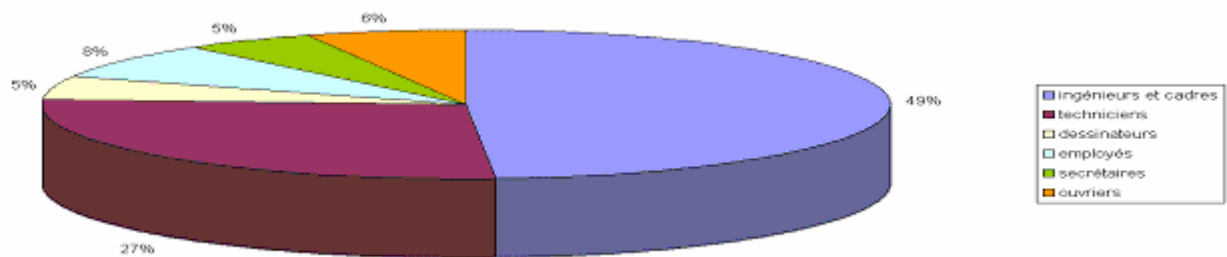
*Figure I.3 : Organisation du centre technologique SRPC.*

**Le Groupe TECHNO ELECTRIC** (Managé par les deux traces rouges sur l'organigramme) se charge de sélectionner les technologies fiables pour les milieux hostiles, et de qualifier les composants pour le métier électrique.

Il se compose de 8 personnes embauchées à plein temps, en plus des prestataires et des stagiaires dont je fais partie. C'est en effet, au sein de ce groupe que j'effectue mon stage de fin d'étude.

### I.3.2 Le personnel :

La répartition par catégorie socioprofessionnelle de SRPC traduit bien l'activité du site et le profil hautement qualifié et très technique de son personnel.



*Figure I.4 : Répartition du personnel à SRPC*

SRPC est avant tout un centre d'études et de recherches pour l'amélioration et la conception des outils de Logging (Wireline and testing) et la mise au point des meilleurs procédés de cimentation des puits (Dowell). Ceci explique donc l'importance de la population d'ingénieurs et de cadres qui sont également chargés de gérer les achats de matériels et de pièces détachées pour les outils assemblés sur le site. Ces mêmes pièces détachées sont essentiellement fabriquées par sous-traitance, ce qui explique le nombre relativement faible d'ouvriers et de techniciens.

Enfin, les dessinateurs sont chargés de concevoir les nouveaux outils et les ensembles électroniques.

### I.3.3 La mission du centre :

SRPC a pour mission de concevoir, développer et fabriquer des produits destinés aux activités de services pétroliers suivants :

- Reservoir Evaluation
- Reservoir Development

Ces produits sont essentiellement fournis à Schlumberger Oilfield Services.

Cependant, certains produits spécifiques sont conçus pour être vendus ou loués directement aux clients tels que des vannes de sécurité, des manomètres permanents et des équipements de mise en production temporaire. De plus, des produits sélectionnés (équipement d'essai de puits et de travail de câble) sont vendus de préférence sous leur forme standard mais peuvent incorporer des modifications limitées pour répondre aux besoins du client.

### **I.3.4 Les produits et services :**

Ce sont la marque Schlumberger, ils peuvent se regrouper selon les grandes familles suivantes, considérées dans le périmètre de certification ISO des activités SRPC :

- Les équipements d'évaluation des réservoirs : équipement de logging, pour l'évaluation en puits ouvert ou tubé.
- Les équipements de développement des réservoirs : équipements de test en surface, équipement de sécurité de tête de puits, systèmes de mise en production et de contrôle de réservoir, solution en cimentation et stimulation.

Ainsi l'entreprise Schlumberger propose des services aux détenteurs des champs pétroliers, elle utilise des outils qu'elle fabrique elle-même au sein des centres technologiques et de production. Ces centres sont équipés de moyens modernes et permettent aux nouveaux venus d'acquérir un certain savoir faire dans des domaines différents.



# CHAPITRE II

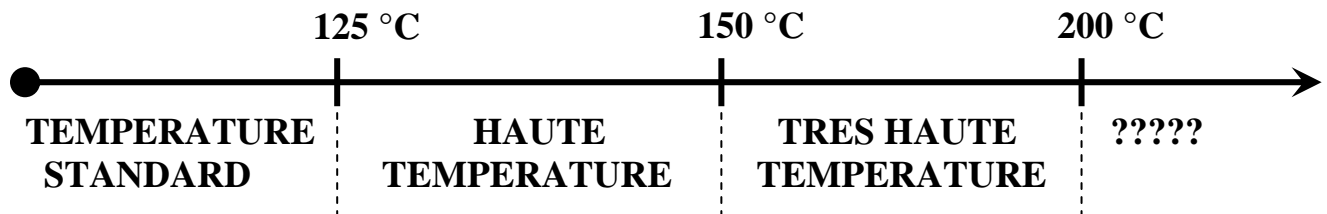
## INITIATION A L'ELECTRONIQUE HAUTE- TEMPERATURE

Mettre en œuvre le plus simple des circuits en haute température n'est pas aussi simple qu'on le croit, car plusieurs phénomènes inattendus peuvent constituer une menace pour les composants électroniques. C'est pour cela que des domaines comme l'industrie militaire, l'industrie automobile et les services pétroliers nécessitent des moyens énormes pour prévenir et combattre les méfaits de la haute température sur les composants électroniques.

Ainsi, on s'intéressera dans ce chapitre à la haute température, à ses dangers et à l'approche Schlumberger visant à choisir ses technologies pour la haute température.

## II.1 QU'EST-CE QUE LA HAUTE TEMPERATURE ? : [1]

Plusieurs classifications de gammes de température pour les composants électroniques ont été établies par différents experts ; ceci a entraîné une confusion lors du tri des applications ou des solutions trouvées. Quelques classifications les considéraient comme dédié à la haute température et d'autres pas. Pour cela chaque segment a établi sa propre classification, pour le domaine des services pétroliers; elle se présente comme suit :



*Figure II.1 : Classification des gammes de températures pour les circuits électroniques.*

On remarque que la haute température est située dans l'intervalle [125, 150] °C qui peut changer en fonction de l'application et des moyens utilisés; ceci reste une notion relative étroitement liée à l'application et à la technologie.

En général les fabricants de composants électroniques certifient leurs produits jusqu'à 85°C pour les applications grand public et 125°C pour les applications spécifiques (militaires, etc.) ; ces températures représentent la frontière inférieure de la haute température.

## II.2 DEFINITION DE L'ELECTRONIQUE HAUTE TEMPERATURE : [2]

L'électronique haute température est une discipline qui opère à des températures qui excèdent celle rencontrée habituellement par l'électronique conventionnelle. Elle est essentiellement basée sur cette dernière, à qui elle apporte des améliorations de design, de package et de propriétés diverses pour minimiser les effets de la haute température.

## II.3 EFFETS DE LA TEMPERATURE SUR LES COMPOSANTS ELECTRONIQUES :

Un système électronique qui fonctionne en haute température est confronté à des phénomènes divers qui peuvent le mettre hors service. Le plus important est le vieillissement accéléré des composants électroniques qui représente un problème majeur, car il ne s'agit pas seulement de faire mettre en service ce système à telle ou telle température mais de certifier aussi un temps minimum de fonctionnement à la température en question.

Le vieillissement est dû à d'autres phénomènes présents, en haute température, appelés mécanismes de défaillances. Les déformations thermomécaniques et la dissipation thermique en sont des exemples et peuvent sérieusement dégrader le package du composant; alors que le latch-up et l'électromigration peuvent porter atteinte au silicium.

**II.3.1 Vieillessement accéléré des composants électroniques : Loi d'Arrhenius [1],[3]**

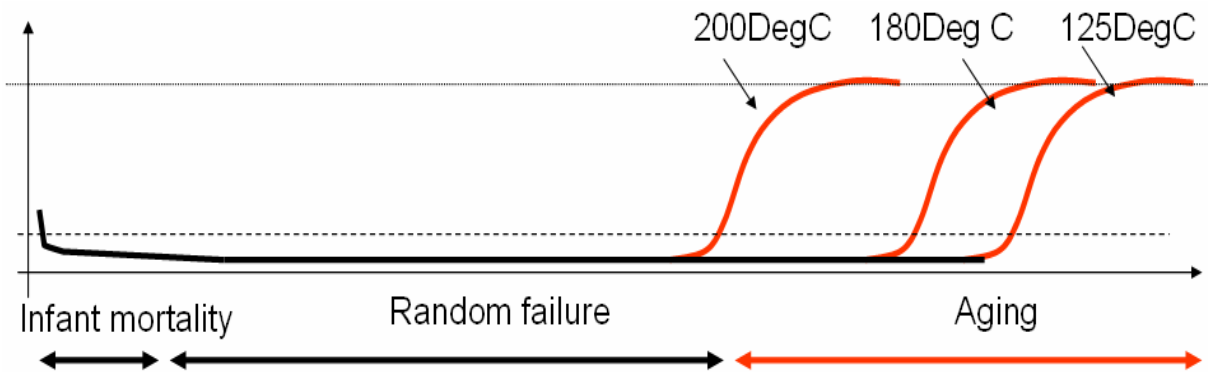
Exposés à la haute température, les composants subissent une usure très importante qui réduit sa durée de vie. Il s'agit d'un vieillissement accéléré qui s'accroît exponentiellement en fonction de la température selon la loi d'Arrhenius :

$$r = A \exp(-E_a/kT)$$

Où **r** : le facteur d'accélération de vieillissement.  
**T** : la température absolue en °K  
**E<sub>a</sub>** : l'énergie d'activation (eV) qui dépend du matériau et du mécanisme de défaillance  
**k** : constante de Boltzmann (8,62 . 10<sup>-5</sup> eV/°K)  
**A** : Une constante.

On remarque que si la température augmente de quelques degrés, le facteur d'accélération augmente très vite ; donc un composant vieillit beaucoup plus vite quand il est à une température élevée. La *Figure II.2* permet d'observer l'évolution de la courbe d'âge d'un composant électronique en fonction de la température.

**Mortality**



*Figure II.2 : Evolution de la courbe d'âge d'un composant électronique en fonction de la température.*

Pour être plus explicite, effectuons un calcul simple en appliquant la loi d'Arrhenius, pour montrer l'impact de la température sur la durée de vie.

Prenons un cas particulier où on a un composant dont l'énergie d'activation **E<sub>a</sub>** = 1 eV reste constante en fonction de la température (Le même mécanisme de défaillance quelque soit la température) alors on peut écrire :

A la température **T<sub>1</sub>** on a la relation :  $r_1 = A \exp(-E_a / kT_1)$  ..... (1)

De même à la température **T<sub>2</sub>** on a :  $r_2 = A \exp(-E_a / kT_2)$  ..... (2)

En divisant l'équation (1) par (2) on obtient le rapport des accélérations **R<sub>12</sub>** :

$$R_{12} = \frac{r_1}{r_2} = \exp\left[ -\frac{E_a}{k} \left( \frac{1}{T_1} - \frac{1}{T_2} \right) \right]$$

Si on calcule  $R_{12}$  et  $R_{13}$  pour les températures  $T_1 = 150 \text{ }^\circ\text{C} = 423 \text{ }^\circ\text{K}$ ,  $T_2 = 200 \text{ }^\circ\text{C} = 473 \text{ }^\circ\text{K}$  et  $T_3 = 225 \text{ }^\circ\text{C} = 498 \text{ }^\circ\text{K}$  ( en supposant que le composant fonctionne à ces températures) on obtient les résultats suivants :

$$R_{12} = 0,055 \text{ et } R_{13} = 0,016$$

Cela veut dire que si le composant en question à une durée de vie  $t_1$  à  $150^\circ$ , il aurait des durées de vie  $t_2 = 0,055 t_1$  à  $200 \text{ }^\circ\text{C}$  et  $t_3 = 0,016 t_1$  à  $225^\circ\text{C}$ .

Prenons un exemple : si ce composant à une durée de vie de 5 ans (48300 heures) à  $150^\circ\text{C}$  alors il peut vivre 2412 heures (100 jours) à  $200^\circ\text{C}$  et 700 heures (29 jours) à  $225^\circ\text{C}$ .

Donc :

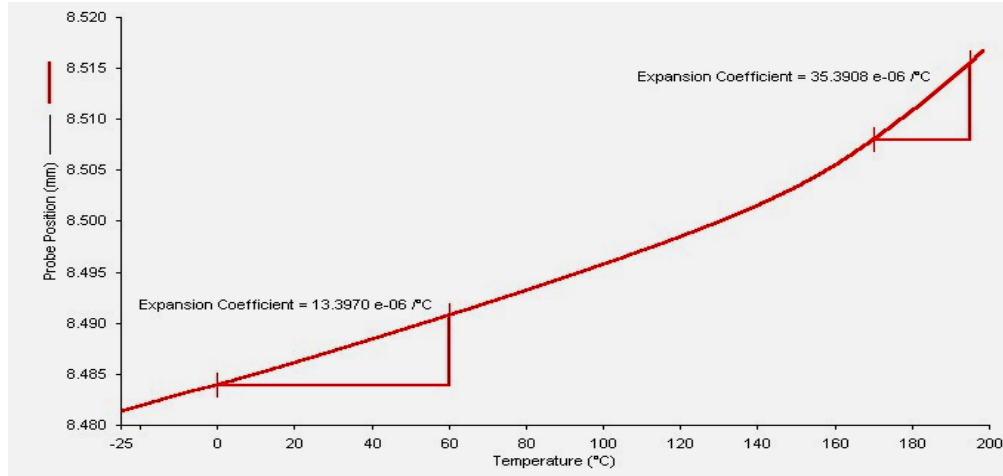
$$150 \text{ }^\circ\text{C} / 5 \text{ ans} \Leftrightarrow 200 \text{ }^\circ\text{C} / 100 \text{ jours} \Leftrightarrow 225 \text{ }^\circ\text{C} / 29 \text{ jours}$$

Ainsi on peut remarquer l'étendue des dégâts de la haute température sur la durée de vie des composants électroniques, en plus dans ce cas on a supposé que le mécanisme de défaillance reste le même donc il n'y a pas superposition de ces phénomènes ce qui n'est généralement pas le cas.

Intéressons nous maintenant aux différents mécanismes de défaillances.

### II.3.2 Déformations thermomécaniques : [1], [4]

La variation de température provoque l'expansion des matériaux; plus la température est élevée plus l'expansion est importante. On voit cela à l'évolution du coefficient thermique d'expansion (CTE) en fonction de la température représentée par la *Figure II.3*



*Figure II.3 : Variation du CTE en fonction de la température*

On remarque que lorsque la température augmente au delà de  $120 \text{ }^\circ\text{C}$  la valeur du coefficient thermique d'expansion augmente de façon importante (trois fois sa valeur à  $25^\circ\text{C}$ ), ainsi en haute température l'expansion est plus élevée et augmente plus vite en fonction de la température.

Deux matériaux différents subissent généralement des expansions différentes puisqu'ils ont des coefficients d'expansion différents (exemples: CTE silicone = 2.8, CTE cuivre = 16.5, CTE aluminium = 23.4). Le problème se pose quand ces deux matériaux sont liés, ils subissent la même expansion mais cela crée un stress mécanique qui entraîne des déformations thermomécaniques et des fissurations comme le montre la *Figure II.4*.

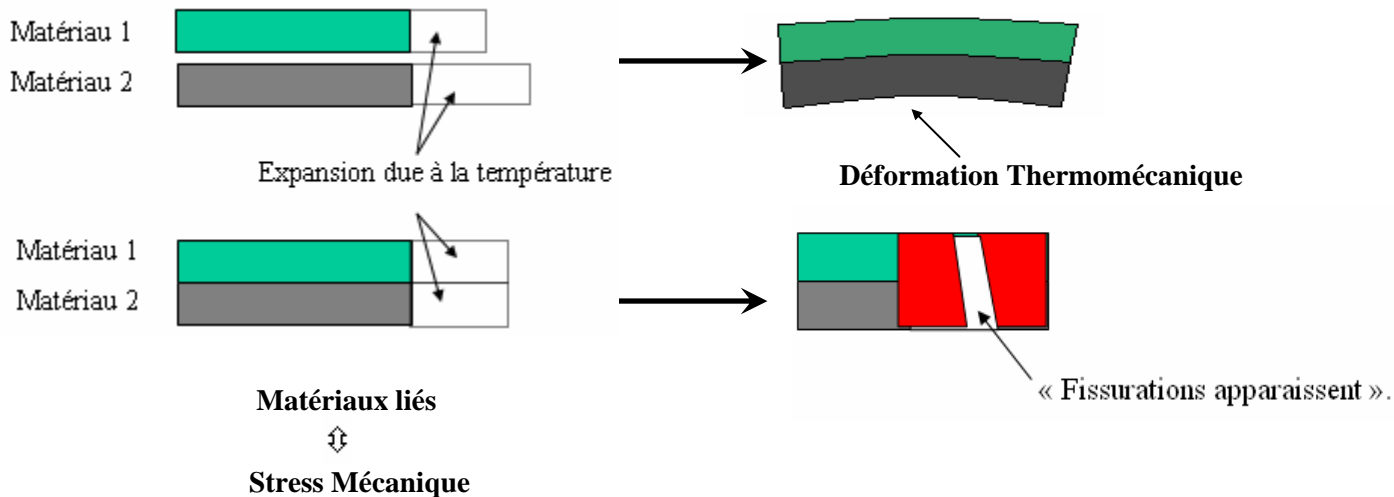


Figure II.4 : Effet de l'expansion thermique sur deux matériaux liés.

Sur un composant électronique les choses se passent exactement de cette façon au niveau du boîtier (package) qui commence par se déformer puis se fissurer (cracks sur le boîtier) à cause de la présence de matériaux différents.

La déformation thermomécanique est un mécanisme de défaillance qui peut être fatal au composant électronique, ceci rend plus délicat la mise en œuvre des applications en haute température.

### II.3.3 La diffusion : [1], [5]

La diffusion est la migration d'une espèce sous le seul effet de l'agitation thermique, cette migration peut fragiliser quelques zones du boîtier d'un composant électronique à cause de la baisse de la concentration de l'espèce en question.

La diffusion suit les lois de **Fick**. Il s'agit de lois empiriques, qui ont reçu une démonstration théorique par **Albert Einstein** avec ses travaux sur le mouvement brownien. **Jean Perrin**, fondateur du CNRS et prix Nobel de physique, fut le premier à mesurer la trajectoire de particules soumises au mouvement brownien et confirma ainsi l'analyse théorique d'Einstein. La première loi de Fick énonce que « **le flux de diffusion est proportionnel au gradient de concentration** »; Cette loi est dérivée de la loi de Fourier sur la conduction de la chaleur.

Mathématiquement, cette loi s'exprime de la manière suivante:

Soit un milieu  $m$  dans lequel se trouve une espèce chimique  $i$ . Soit une surface  $S$ . Si l'on appelle  $\vec{j}_i$  le flux de  $i$  à travers  $S$ , c'est-à-dire la quantité d'espèce qui traverse  $S$  en une seconde (en  $\text{mol}\cdot\text{s}^{-1}$ ), et si  $C_i(x, y, z)$  est la concentration de  $i$  en un point donné, alors la **première loi de Fick** donne :

$$\vec{j}_i = -D_i^m \cdot S \cdot \overrightarrow{\text{grad}}(C_i)$$

$D_i^m$  est le coefficient de diffusion de  $i$  dans  $m$  ; il dépend de la température de manière exponentielle puisque, comme le coefficient d'accélération de vieillissement, il suit la loi d'Arrhenius :

$$D_i^m(T) = D_{i0}^m \cdot e^{-\frac{E}{kT}}$$

Où  $E$  est l'énergie d'activation,  $k$  est la constante de Boltzmann et  $T$  la température absolue.

Ainsi en haute température une variation de quelques degrés peut entraîner une grande diffusion et par conséquent la détérioration du boîtier d'un composant électronique.

### II.3.4 Le latch-up : [1], [6]

Le latch-up est un Phénomène indésirable produit dans un semi-conducteur discret par des courants parasites et qui le « verrouille » à l'état passant. Ce verrouillage peut résulter de l'exposition du circuit à un rayonnement ionisant.

En général, les composants CMOS ont une structure parasite sensible aux latch-ups ; ce phénomène entre en activité lorsque la tension d'entrée est supérieure à la tension d'alimentation  $V_{Input} > V_{dd}$ .

La haute température favorise le phénomène de latch-up vu que la tension des éléments actifs (thyristors, transistors, ...) diminue. La *Figure II.6* représente la dégénérescence de la fonction de transfert d'un inverseur CMOS en haute température à cause des problèmes de latch-up.

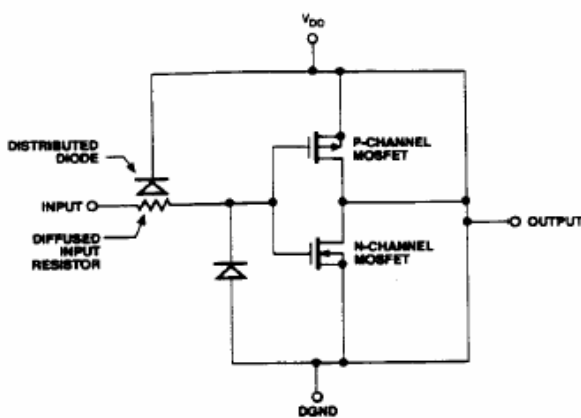
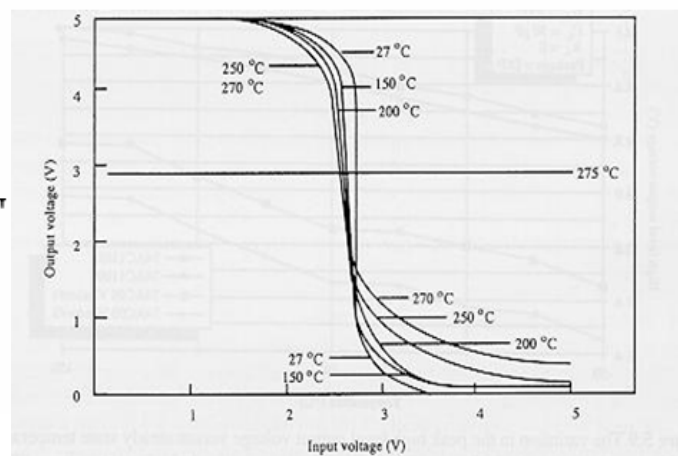


Schéma électrique d'un inverseur CMOS



Inverseur CMOS à différentes températures

*Figure II.6* : phénomène de latch-up en haute température pour un inverseur CMOS.

On peut remarquer qu'à 275°C, l'inverseur ne joue plus son rôle : quelque soit la tension d'entrée il est toujours passant, cela est dû à un latch-up favorisé par l'action de la chaleur sur l'inverseur. Les signes de présence d'un latch-up étaient apparents déjà à 270°C où la tension de sortie de l'inverseur n'est pas nulle pour une entrée de niveau haut, ceci pourrait être considéré comme un latch-up partiel.

Ainsi le latch-up comme mécanisme de défaillance est considérablement favorisé en haute température et peut totalement mettre hors service une application dans ce domaine.

### II.3.5 L'électromigration : [1], [3] et [7]

L'électromigration est le transport d'un ou plusieurs atomes causé par un mouvement gradué des ions dans un conducteur. Ce phénomène n'est significatif qu'en cas d'un transfert conséquent entre les électrons libres et les atomes de métal qui diffuse ce qui implique des densités de courants élevés.

L'électromigration diminue la fiabilité d'un circuit intégré, elle peut provoquer des déplacements d'atomes dans les interconnexions métalliques. Il en résulte des fissures dans les interconnexions (Cracks) ; ceci peut créer des circuits ouverts (*Void*) et des courts circuits (*Hillock*). Ceci conduit à la perte de connexions dans le circuit intégré (Voir *Figure II.7* et *II.8*).

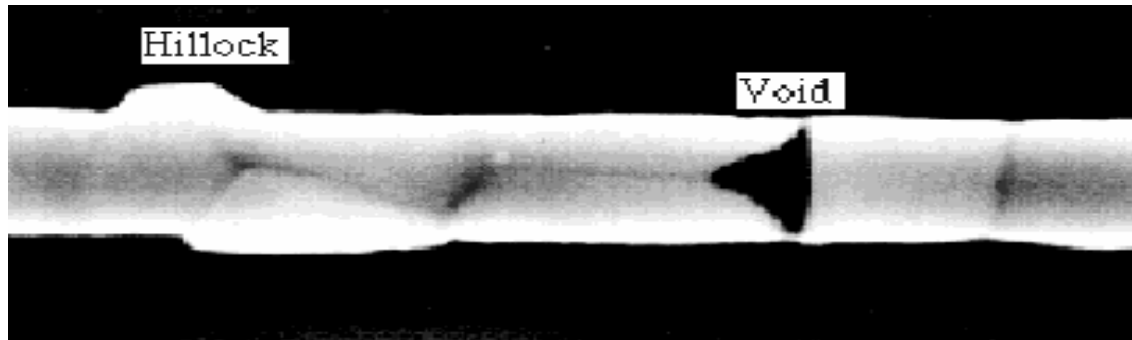


Figure II.7 : Créations de Voids et Hillocks par effet d'électromigration.

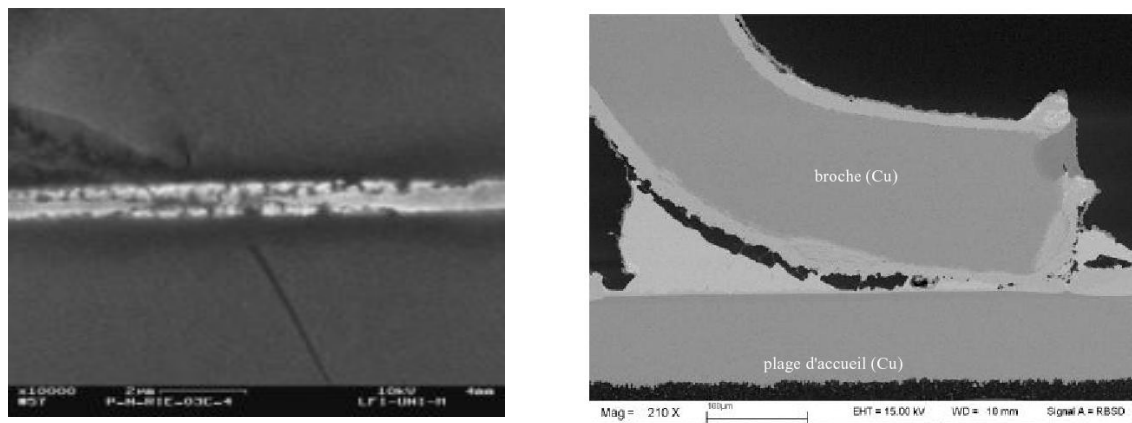


Figure II.8: Interconnexions en cuivre détériorées par électromigration (Cracks)

Dans un conducteur idéal où les atomes sont arrangés dans une structure parfaite, il n'y a pas de collisions électrons - atomes ni d'électromigration. Par contre dans un conducteur réel, l'agitation thermique des atomes provoque des collisions électrons - atomes mais dans la plupart des cas cela ne provoque pas d'électromigration d'atomes car la masse des électrons reste insuffisante pour les déplacer. Cela dit, une grande puissance dans un milieu de température élevée peut apporter le nombre d'électrons suffisant pour déplacer des atomes de métal par un effet d'avalanche puisque chaque atome qui se déplace permet la collision d'autres électrons.

Mathématiquement on peut décrire le phénomène d'électromigration par la relation approximée qui s'écrit comme suit :

$$Emf = -\frac{Na}{kT} \cdot D_0 \cdot \exp\left(\frac{-E_a}{kT}\right) \cdot e \cdot Z^* \cdot \rho \cdot J : \text{Approximation générale d'électromigration.}$$

Où  $Emf$  : le flux de masse des ions (mass flux of ions) ou électromigration.

$Na$  : la densité des atomes.

$D_0$  : coefficient de diffusion (Voir loi de Fick).

$E_a$  : énergie d'activation.

$k$  : constante de Boltzmann.

$T$  : température absolue.

$e \cdot Z^*$  : charge effective.

$\rho$  : résistance spécifique.

$J$  : densité de courant.

On remarque que l'électromigration est fonction de l'expression  $\frac{1}{kT} \exp(-\frac{1}{kT})$ . En supposant que  $E_a = 1\text{eV}$  et  $A = Na \cdot D_0 \cdot (-e) \cdot Z^* \cdot \rho \cdot J$ , traçons alors l'expression  $Emf/A$  dans l'intervalle de température  $[273, 503]^\circ\text{K} \Leftrightarrow [0, 300]^\circ\text{C}$  et analysons l'influence de la température sur l'électromigration.

TEMPERATURE (°K)	Emf / A = 1/kT exp (1/kT)
273	1,49044E-17
283	6,45375E-17
293	2,52545E-16
303	9,0212E-16
313	2,96758E-15
323	9,05918E-15
333	2,58382E-14
343	6,92649E-14
353	1,75446E-13
363	4,21886E-13
373	9,67151E-13
383	2,12166E-12
393	4,46894E-12
403	9,06574E-12
413	1,77609E-11
423	3,36875E-11
433	6,20005E-11
443	1,10952E-10
453	1,93419E-10
463	3,29025E-10
473	5,47023E-10
483	8,90123E-10
493	1,4195E-09
503	2,22118E-09

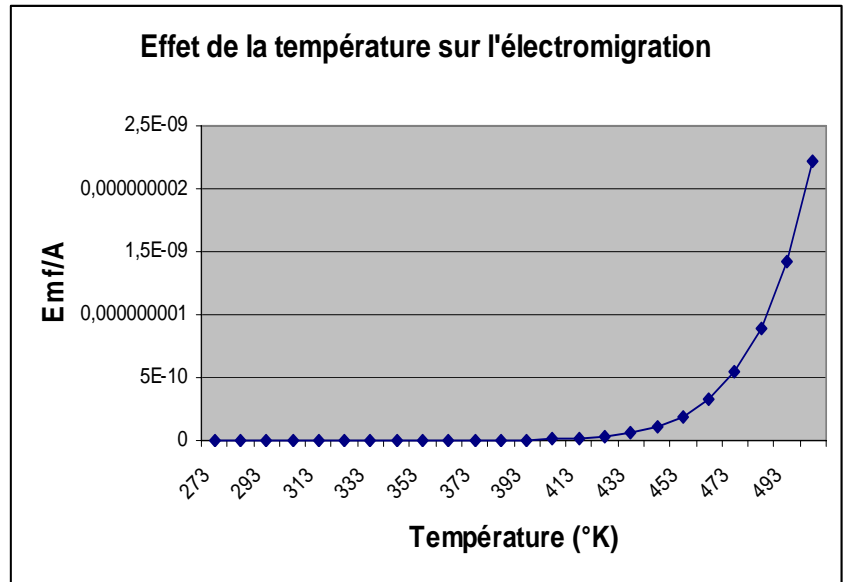


Tableau II.1 : Emf/A en fonction de La température.

Figure II.9 : Effet de la température sur l'électromigration.

On remarque que l'électromigration est proportionnelle à la température, l'augmentation commence à être visible à 160°C (433°K) puis augmente de façon rapide (c'est l'exponentiel qui l'emporte). On peut donc constater que la haute température favorise l'électromigration et multiplie le risque de création des Voids et des Hillocks dans les liaisons intermétalliques. En effet, des experts en électromigration et en électronique haute – température, comme **James Black**, ont pu modéliser le temps moyen de défaillance par électromigration sous la forme d'équation mathématique approximée qui s'écrit comme suit :

$$\frac{1}{MTTF} = C \cdot J^n \cdot \exp\left(-\frac{E_a}{kT}\right) \quad : \text{Approximation du temps de défaillance par électromigration}$$

Où **MTTF** : le temps moyen de défaillance (Mean Time To Failure).

**C** : une constante qui dépend de la géométrie.

**J** : la densité de courant.

**n** : égale à 2 dans des conditions normales.

On remarque bien que le temps moyen de défaillance d'un composant électronique est inversement proportionnel, de façon exponentielle, à la température. Cela veut dire qu'il a le même comportement que le processus de vieillissement à une constante près et qu'une variation de quelques degrés en haute température peut diviser le temps de survie d'un composant de dix fois ou plus.



Ainsi plusieurs mécanismes de défaillances n'ayant aucun impact à température ambiante rentrent en jeu, ils réduisent énormément les chances de réussite d'une application en haute température. Certaines compagnies qui opèrent en haute température ont exprimé le besoin de combattre ces mécanismes de défaillances. Schlumberger répondant à ce critère, découvrons sa démarche vis-à-vis de ce problème.

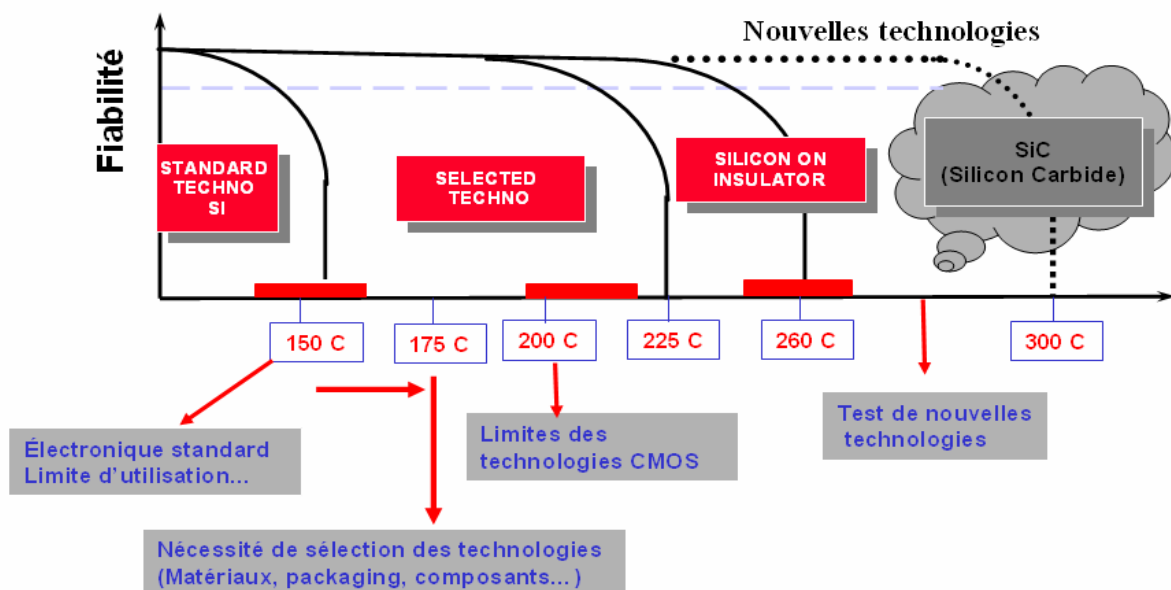
#### II.4 L'APPROCHE SCHLUMBERGER : [1]

Plusieurs segments de Schlumberger ont la haute température comme domaine d'opérations. Pour chaque segment, deux grandeurs sont importantes : la température de fonctionnement (imposé par le milieu) et le temps de fonctionnement à cette température (qualité, fiabilité, rentabilité). Ces segments et leurs caractéristiques sont listés dans le tableau suivant :

Segment	Intervalle de température de fonctionnement	Temps de fonctionnement
Sismique terrestre et marine	[-40, 125] °C	1 an
Logging while drilling	[-40, 150] °C	200/1000 heures
Wire-line	[-40, 175] °C	400 heures
Reservoir monitoring	[150, 225] °C	6 mois
Applications permanentes	150 °C	5 ans

*Tableau II.2 : caractéristiques des segments Schlumberger concernés par la haute température.*

La haute température est un large domaine d'application pour Schlumberger, c'est pour cela qu'elle a déployé de gros moyens pour satisfaire les besoins des services pétroliers, spécialement dans le choix des technologies et des composants électroniques adéquats aux applications en milieu hostile. La *Figure II.10* représente la fiabilité des technologies choisies par Schlumberger en fonction de la température.



*Figure II.10 : Fiabilité des technologies choisies par Schlumberger en fonction de la température.*

Schlumberger choisit des technologies qui existent et teste leur fiabilité en haute température. Pour les températures inférieures à 150°C elle a choisi les technologies standards réservées au grand public à la limite d'utilisation ; cela dit la nécessité de trouver de nouvelles technologies qui fournissent plus de performances et supportent mieux les températures élevées. Elle s'est tournée alors vers une technologie CMOS dédiée aux applications spécifiques (standard Bulk<sup>(1)</sup>) qui a prouvé sa fiabilité jusqu'à 200°C. Mais cette technologie ne satisfaisait pas tout les segments, il a fallu s'assurer de la fiabilité d'autres technologies comme SOI<sup>(2)</sup> (Silicon On Insulator) qui permet d'obtenir des résultats acceptables à des températures qui dépassent les 225°C. Les recherches continuent dans ce domaine vu que Schlumberger espère atteindre les 300°C avec de nouvelles technologies entre autres la technologie SIC<sup>(3)</sup> (Silicon Carbide).

La sélection d'un composant se fait selon deux axes qui se rejoignent comme le montre la *Figure II.11*.

Le premier consiste en un travail de recherche du composant, de sélection de technologie et d'évaluation des différents mécanismes de défaillances qui pourraient intervenir en haute température. Alors que le deuxième est une qualification du composant par rapport à l'application et les performances ; il s'agit tout d'abord de faire une étude de l'application à réaliser et d'établir son design, on passe ensuite à une étape de construction virtuelle et de simulation pour vérifier la faisabilité de l'application. En cas de résultats concluants, on passe à la phase de choix des matériaux utilisés pour prévenir les dangers de la haute température, c'est à cette étape que les deux axes se rejoignent pour insérer la technologie du composant. Ensuite on doit évaluer les mécanismes de défaillances pour tout le système et finir par des tests de qualifications en haute température qui permettent de certifier la fiabilité du composant.

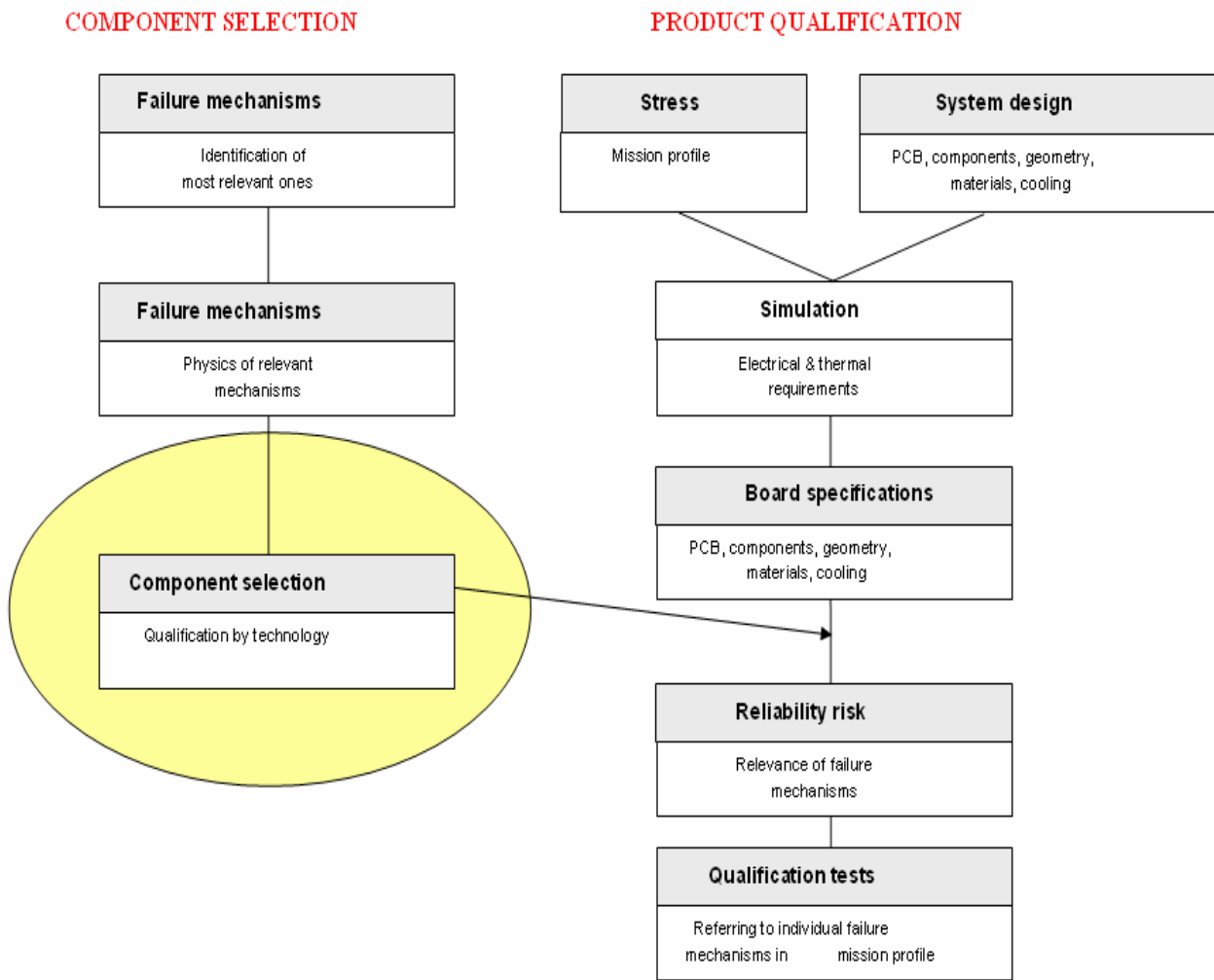
Choisir la technologie du composant nécessite tout de même une certaine expérience dans le domaine de l'électronique haute – température car elle repose sur toutes les études faites au préalable sur les qualifications des différentes technologies en haute température.

---

(1) : Bulk: Standard Si wafer, c'est la technologie classique des semiconducteurs construite sur substrat massif (Bulk), elle généralement sous forme de wafer:(disque circulaire en silice fait à partir d'un seul cristal semiconducteur, il contient généralement plusieurs dizaines de circuits intégrés.

(2) : SOI : Structure en couches qui consiste en de fines couches de silicium (50 nm – 100 µm) créées sur un substrat isolant, généralement du saphir ou du dioxyde de silicium (SiO<sub>2</sub>) (Voir Référence [28]).

(3) : SIC : Composé céramique à base de silicium et de carbone, il est utilisé dans des applications haute Température car il garde un coefficient de température négatif jusqu' 900°C. (Voir Référence [29])



*Figure II.11 : Choix d'un composant électronique selon l'approche Schlumberger*

Ainsi pour assurer la fiabilité des produits qu'utilise Schlumberger, on fait un tri minutieux des composants présents sur le marché grâce à une série de tests qui révèlent les performances ou les défaillances en haute température.

Notre but maintenant, est d'appliquer cette approche pour qualifier un microcontrôleur 8/16 bits afin d'acquérir ne serait ce qu'un peu de ce savoir faire.

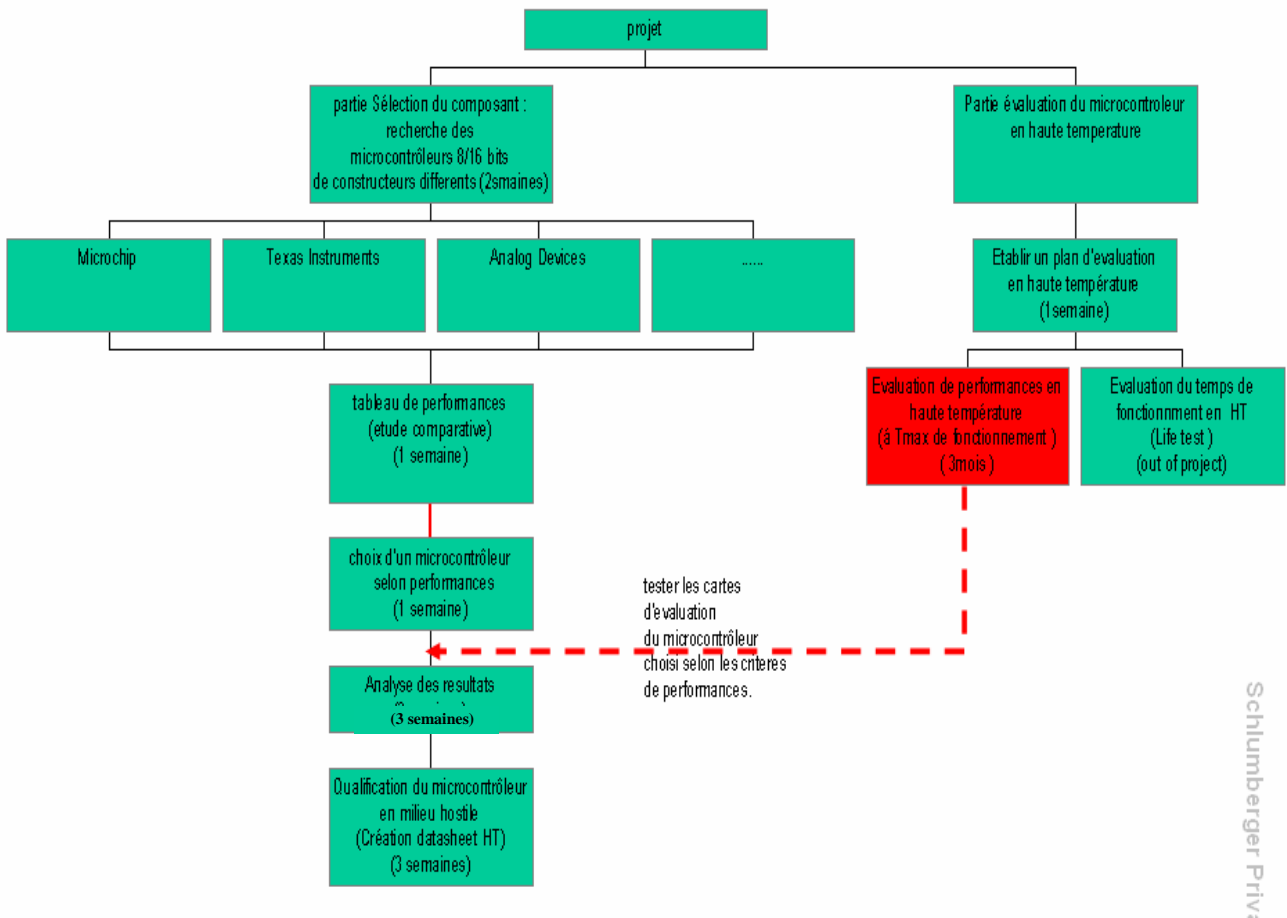
# CHAPITRE III

## PLAN DE TRAVAIL ET SELECTION DU MICROCONTROLEUR A EVALUER

Le but de ce chapitre est de présenter le plan de travail à suivre pour qualifier un microcontrôleur dédié à faire des mesures en haute température dans un premier temps, puis de faire une étude pour choisir le microcontrôleur en question sachant qu’il doit être de faible consommation, pas très encombrant, pas très cher et donnant des résultats précis.

**III.1 PLAN DE TRAVAIL :**

Ce projet consiste à faire un choix d’un microcontrôleur 8 ou 16 bits performant en haute température car il sera destiné au domaine Oilfield, dont les températures peuvent atteindre 150 à 200 °C, ceci peut entraîner des dégradations de fonctionnement vu que la plupart des microcontrôleurs sont certifiés par les constructeurs dans les intervalles de températures standards [- 40, 85 ] C° pour des applications grand public et [- 40, 125 ] C° pour des applications plus spécifiques telles que les applications militaires. Donc les tâches principales consistent à faire une sélection selon les performances d’un microcontrôleur parmi la masse présente sur le marché (partie théorique) puis l’évaluer en haute température (partie pratique) ; ceci revient, dans notre cas, à déterminer la température maximale de fonctionnement fiable et jauger les performances à cette température. L’évaluation d’un composant nécessite aussi de connaître la durée de fonctionnement en haute température (Life test) mais cela ne fait pas partie du projet. La *Figure III.1* représente les différentes tâches de ce projet ainsi que la durée estimée pour chacune d’elles.



*Figure III.1 : Plan du projet*

En se référant au diagramme ci-dessus, le projet est composé de deux parties majeures qu'on se propose d'expliquer comme suit :

### **III.1.1 Partie sélection du microcontrôleur (partie théorique) : (4 à 5 semaines)**

Cette partie consiste à réaliser les tâches suivantes :

- a) Recherche des microcontrôleurs 8/16 bits de constructeurs différents (Microchip, TI, AD, ...)
- b) Etude comparative selon performance (consommation, rapidité, capacité mémoire...) en consultant les datasheets et en notant à chaque fois la température limite donnée par le fabricant.
- c) Choix du meilleur microcontrôleur, selon performance, (vitesse, taille mémoire, périphériques, etc....) susceptible de donner de bons résultats en milieu hostile (haute température).

### **III.1.2 Evaluation du microcontrôleur choisi en haute température (partie pratique): (4 mois)**

Cette partie, plus pratique et plus longue à réaliser, représente le cœur du projet ; elle passe par les étapes suivantes :

- a) Etablir les critères de performances ainsi qu'un plan d'évaluation en milieu hostile (cahier des charges des tests à effectuer).
- b) Faire les tests, en haute température, sur cartes d'évaluation correspondant au microcontrôleur choisi.
- c) Analyser et discuter les résultats obtenus.
- d) Qualifier le microcontrôleur en haute température et établir une datasheet haute température.

### **III.2 SELECTION DU MICROCONTROLEUR A EVALUER : [8], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23].**

Pour choisir le microcontrôleur on procédera par une étude bibliographique qui consiste à balayer la plupart des constructeurs 8 / 16 bits puis faire un tableau comparatif des performances en se référant aux données fournies par les fabricants dans les datasheets. On s'intéressera aux noyaux des microcontrôleurs, à leurs vitesses, leurs résolutions, leurs tailles mémoires, les différents périphériques mis à la disposition de l'utilisateur et les intervalles de températures assurés par les fabricants.

Un Tableau comparatif des différents microcontrôleurs 8 / 16 bits est présenté dans l'Annexe C ; Il faudra donc l'interpréter fabricant par fabricant, s'intéresser aussi aux données de marché et sortir avec un choix judicieux de microcontrôleur assez performant qui peut donner de bons résultats de mesures en haute température.

Interprétons le tableau précédent en étudiant brièvement les microcontrôleurs proposés par chaque fabricant. Au bout de cette étude on devra décider du composant à tester en haute température.

#### a) Analog Devices (AD) :

La compagnie Analog Devices s'est spécialisée dans le traitement de signal par excellence; elle offre une large gamme de circuits intégrés (IC) permettant de traiter des données analogiques, mixtes ou numériques (DSP) distribués sur différents domaines : industrie automobile, instrumentation, ordinateurs, la défense, etc....

En ce qui nous concerne, on s'intéresse au microcontrôleurs de résolutions [8 – 16] bits. Ce fabricant met à la disposition des utilisateurs deux familles de microcontrôleurs de ce type :

- **ADuC70XX** (12 bits) : basée sur un noyau ARM7 TDMI<sup>(1)</sup> cette famille préfère les périphériques à la vitesse car elle contient différents types d'interfaces série et parallèles, et des convertisseurs analogique / numérique et numérique / analogique très performants aussi bien en résolution qu'en vitesse de conversion.
- **ADuC8XX** (12 / 16 bits) : cette famille comporte un noyau 8052<sup>(2)</sup> standard compatible mais plus performant qu'un 8051<sup>(3)</sup>. Elle garde les mêmes caractéristiques que la famille aduc70xx, les périphériques sont plus performants mais la vitesse est moindre.

Ces deux familles assurent un bon fonctionnement dans un intervalle de température assez correct [- 40, 125] °C mais leurs prix sont élevés car elles sont réservées pour des applications militaires.

#### b) Atmel Corporation :

Ce fabricant offre une famille de microcontrôleurs (**AT91XX**) très performants grâce à leur rapidité et à leur taille mémoire. Cependant le manque de périphériques tels que des convertisseurs analogique / numérique impose des données numériques en entrées. Ceci pourrait représenter un handicap pour le domaine oilfield vu qu'on fait principalement de l'instrumentation ; de plus cette famille est certifiée dans un intervalle de température [- 40, 85] °C par le fabricant et risque de ne pas donner de bons résultats en haute température.

#### c) Cygnal Integrated Product (Silicon Laboratories) :

Ce fabricant a choisi d'allier vitesse et diversité de périphériques car il propose des microcontrôleurs 8 bits à base d'un noyau 8051 rapide (la famille **C8051XXX**), ils pourraient s'avérer intéressants à être évalué en haute température sauf que la gamme de température qui nous intéresse [- 55, 125] °C est proposée à un coût onéreux, ce qui permet d'exprimer une certaine réticence quand à son choix.

---

(1): Le processeur ARM est basé sur une architecture RISC 32 bits. Le ARM7TDMI comporte 3 étages pipeline ; et dispose d'un jeu d'instructions sur 16 bits d'où le gain en mémoire. Voir Référence [24].

(2): Le cœur 8052 à été inventé par Intel c'est une amélioration du cœur 8051. Pour plus d'information voir Référence [10].

(3): Le 8051 est un  $\mu$  contrôleur 8 bits inventé par Intel pour plus d'information voir Annexe C et Référence [10]

**d) Dallas Semiconductor Maxim :**

Des microcontrôleurs 8 bits et 16 bits sont proposés par Dallas Semiconductor Maxim; les familles *DS80C320* et *DS80C323* sont à base d'un noyau 8051 et de bus de données 8 bits, leurs avantages sont leur prix et leur faible consommation mais leurs performances en vitesse et en périphériques restent modestes; entre autres l'absence de convertisseurs.

Par contre, les familles *MAXQ3120* et *MAXQ2000* de noyau 16 bits RISC <sup>(1)</sup> sont plus performants au niveau vitesse et interfaces de communication mais restent dédiés au traitement des données numériques, ceci est expliqué par l'absence de convertisseurs analogique / numérique sauf quelques exceptions.

Les microcontrôleurs de Dallas Semiconductor n'opèrent que dans l'intervalle de température [- 40, 85] °C, en plus, leurs kits d'évaluation sont restreints à l'intervalle [0, 70] C° ce qui rend difficile leur évaluation en haute température.

**e) Freescale Semiconductor :**

Freescale met à la disposition des utilisateurs des microcontrôleurs 8 bits basés sur la famille noyaux 68HC05/08/11 de Motorola. Il est vrai que la gamme de package, de périphériques et de convertisseurs est assez impressionnante mais les performances en vitesse restent limitées en plus les microcontrôleurs de Motorola sont pour la plupart obsolètes ; donc il n'est pas du tout intéressant de les évaluer en haute température.

**f) Fujitsu :**

Même si elle est plutôt focalisée software, FUJITSU propose néanmoins des microcontrôleurs 8 bits et 16 bits; les plus connus sont les familles *MB95XXX* et *MB89XXX* pour les 8 bits et *MB90XXX* pour les 16 bits. Ces microcontrôleurs peuvent avoir des tailles mémoire assez importantes et suffisamment d'interfaces série, mais cela influe sur la vitesse qui est relativement réduite. En plus ces microcontrôleurs sont certifiés jusqu'à 85°C seulement. Ainsi l'étude de ces familles en haute température ne paraît pas apporter de bons résultats.

**g) Infineon :**

Infineon propose des familles de microcontrôleurs 8 bits de noyau 8051 ainsi que des microcontrôleurs 16 bits basés sur une architecture pipeline. Les premiers sont les familles *C500/C800* basées sur une architecture standard et *XC800* basée sur une architecture haute performance; les seconds sont les familles *C166* et *XC166* à 4 et 5 étages pipeline respectivement. Ces microcontrôleurs sont performants au niveau de la taille mémoire et des périphériques; surtout la nouvelle famille *XC166* qui propose un convertisseur analogique/numérique à résolution programmable et une vitesse de traitement assez satisfaisante mais cela dit sa consommation est élevée par rapport à ses concurrents. Deux gammes de température sont disponibles [- 40, 85] °C et [- 40, 125] °C mais la dernière affiche des coûts plutôt élevés.

---

(1): L'architecture RISC est une architecture à jeu d'instruction réduit (Reduced Instruction Set Computer). Son principe est de restreindre le jeu d'instructions seulement à quelques instructions, vu qu'elles ne sont généralement pas toutes utilisées, en plus elle leur impose le même nombre de cycles d'exécution => possibilité d'une structure pipeline pipeline. Pour plus d'information voir Référence [25].



**h) Lattice :**

Ce fabricant plutôt spécialiste des circuits programmables tels que les CPLDs et les FPGAs, propose des microcontrôleurs 8 bits d'architectures différentes. On trouve d'abord la famille **DF68XX** à base d'un noyau DF68-8 bit-IP, elle offre une vitesse de traitement intéressante et une interface SPI pour la communication série seulement; les autres périphériques sont absents notamment les convertisseurs ADC et DAC. Lattice propose aussi la famille **R8051** de noyau 8051 et d'une architecture RISC, elle offre une vitesse supérieure à la famille précédente mais la taille mémoire et la disponibilité des périphériques sont d'autant plus médiocres car on ne trouve même pas une mémoire programme interne et les interfaces se limitent à un émetteur / récepteur asynchrone (UART). La troisième famille **DP8051**, basée sur un noyau 8051 et une architecture pipeline, est sans doute la plus performante. Elle offre de multiples interfaces série; mais l'absence de convertisseurs et la RAM insuffisante peuvent être un handicap pour les utilisateurs et les designers.

**i) Microchip :**

Depuis 1999 Microchip cherche à être le leader dans le domaine des microcontrôleurs, c'est pour cela qu'elle a adopté une stratégie productive basée principalement sur des microcontrôleurs 8 et 16 bits d'architecture RISC ; ce qui lui a permis non seulement d'envahir le marché mondial mais aussi de multiplier ses ventes de microcontrôleurs par deux. Ainsi Microchip donne le choix à l'utilisateur vu les différentes gammes de microcontrôleurs 8 bits et 16 bits qu'elle propose. Par exemple pour les 8 bits, on va des familles les plus anciennes et les moins performantes telle que **PIC10FXXX** aux familles les plus complètes aux niveaux: vitesse, taille mémoire et disponibilité des périphériques telle que la famille **PIC18FXXX**; de plus on trouve entre ces deux familles des familles intermédiaires telles que **PIC12FXXX** et **PIC16FXXX** qui complètent la liste et permettent une liberté de choix selon les besoins. De même pour les microcontrôleurs 16 bits, les **dsPIC** une famille de microDSP très performants, se subdivise en plusieurs groupes le plus performant étant **dsPIC30F601X** et le moins performant est **dsPIC30F201X**; et entre les deux on trouve trois groupes qui sont **dsPIC30F301X**, **dsPIC30F401X** et **dsPIC30F501X** de plus en plus performants respectivement. Tout les microcontrôleurs de Microchip adoptent une architecture RISC haute performance, ils mettent à la disposition de l'utilisateur plusieurs périphériques tels que des ADC 10, 12 et 16 bits haute précision et différentes interfaces séries; ainsi que des bus CAN. Une multitude de packages est disponible; allant de 6 à 128 pins. La plupart de ces microcontrôleurs peuvent fonctionner dans l'intervalle de température [- 40, 125] °C, ce qui est très intéressant pour le domaine oilfield.

**j) National Semiconductor :**

Ce fabricant propose une famille 8 bits COP8XXXX de noyau COP8<sup>TM</sup> et une autre 16 bits **CR16MAS9** de noyau CR16B RISC (propre à National Semiconductor) . Ces deux familles présentent des propriétés mémoires plus que satisfaisantes et les périphériques tels que les ADC et les interfaces série sont disponibles, surtout pour les microcontrôleurs 16 bits, mais leur vitesse de traitement est relativement faible et la résolution des convertisseurs est seulement de 8 ou 10 bits.

Ces microcontrôleurs sont certifiés pour les gammes de température [0, 70 ] °C et [-40, 125] °C pour les microcontrôleurs 8 bits et [-40, 85] °C et [-40, 125] °C pour les microcontrôleurs 16 bits, les premiers coûtent beaucoup trop cher ce qui écarte la possibilité d'évaluer ce genre de microcontrôleurs; les seconds sont en début de production puisque la

plupart des microcontrôleurs sont à l'état de prototype, donc ils n'ont pas encore prouvé leur fiabilité.

#### **k) NEC Electronics :**

NEC offre la famille **KO/X 2**, c'est des microcontrôleurs 8 bits d'architecture CISC <sup>(1)</sup>, leur taille mémoire est satisfaisante et les périphériques disponibles mais pas assez performants puisqu'on trouve seulement un convertisseur 10 bits bien insuffisant pour faire des mesures de haute précision.

#### **l) Renesas :**

Ce fabricant propose les familles suivantes :

- La famille **740** : des microcontrôleurs 8 bits dont la taille mémoire est satisfaisante mais la vitesse est relativement faible et les convertisseurs qu'il propose sont de faible résolution. Ils fonctionnent généralement dans un intervalle de température [-20, 85] °C.
- La famille **H8** : ce sont des microcontrôleurs basés sur le noyau H8 de Renesas; leur avantage est la disponibilité des périphériques comme les convertisseurs ainsi que les interfaces séries mais ces microcontrôleurs restent lents et leurs tailles mémoire sont insuffisantes. Ils sont certifiés dans l'intervalle de température [-20, 75] C°, ce qui reste éloigné des températures de tests désirés.

#### **m) Texas Instruments :**

Ce fabricant est plus spécialisé DSP que microcontrôleurs, cela dit l'expérience acquise sur DSP a permis à TI de fabriquer des microcontrôleurs très performants appelés Mixed Signal Processors qui sont des microcontrôleurs alliant les signaux analogiques et numériques et sont principalement dédiés à l'instrumentation. En ce qui nous concerne les microcontrôleurs qui nous intéressent sont le **MSC1201** et le **MSP430** de résolution 8 et 16 bits respectivement. Ils sont connus pour leur très faible consommation. Les **MSP430** sont d'architecture RISC alors que les **MSC1201** sont à base du noyau 8051. Ces deux familles offrent des tailles mémoires et des vitesses satisfaisantes ainsi que des convertisseurs de grande précision surtout pour le **MSC1201** puisque sa résolution est de 24 bits, il n'est pas encombrant et son coût est celui d'un composant grand public ; en plus il est certifié jusqu'à 125°C ce qui fait de lui le candidat idéal pour des tests en haute température.

#### **n) Toshiba :**

Deux familles de microcontrôleurs 8 bits (**TMP86XXX**, **TMP88CX40**) et une autre 16 bits (**TMP91XX27**) sont proposées par ce fabriquant ; ils offrent des tailles mémoire satisfaisantes mais leurs vitesses et la résolution de leurs convertisseurs ne font pas l'unanimité.

Cette brève description des différents microcontrôleurs présents sur le marché nous donne une idée sur la difficulté de faire un choix vue la diversité des produits. Dans notre cas, le choix se fait de lui-même puisqu'on voit vraiment que le **MSC1201**<sup>(1)</sup> se détache du lot en satisfaisant les critères qu'on s'était fixés. Il ne reste qu'à espérer qu'il donne de bons résultats en haute température.

---

(1) : Il y'a deux versions du MSC1201, on prendra la MSC1201Y3 car elle propose un mémoire flash plus Conséquente.

# CHAPITRE IV

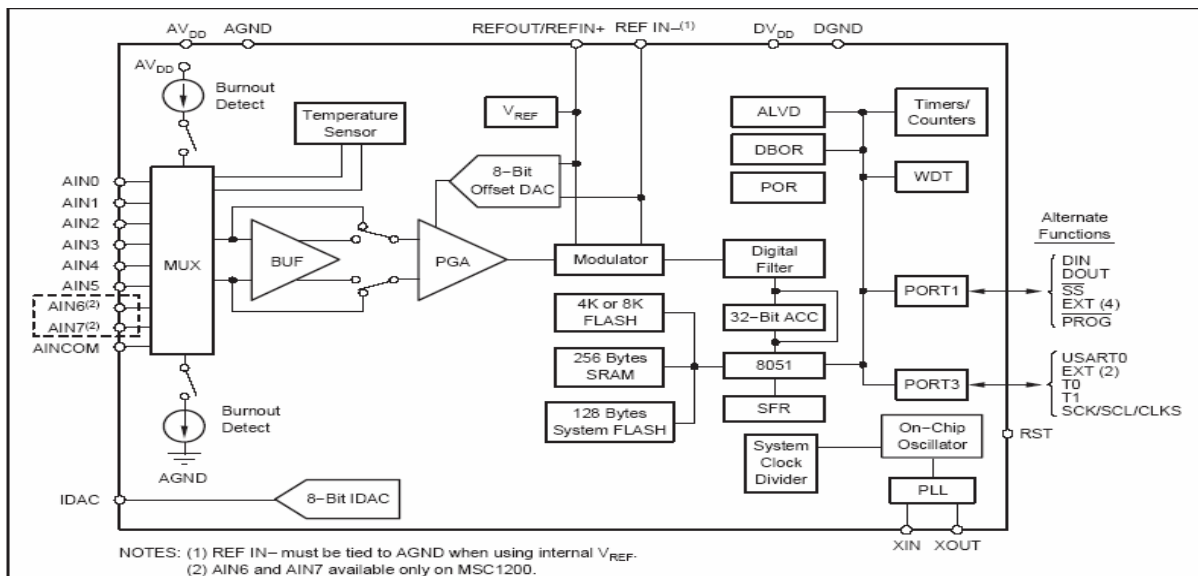
## ETUDE DU MICROCONTROLEUR MSC1201Y3

Maintenant qu'on a choisi le microcontrôleur à évaluer, étudions le de façon plus détaillée afin d'établir les différents programmes de tests. La description suivante est totalement inspirée de la documentation technique du MSC1201Y3 [8].

#### IV.1 DESCRIPTION ET SCHEMA BLOC :

Le microcontrôleur MSC1201Y3 est un composant qui joint des signaux analogiques et numériques connu sous le terme anglais " mixed signal device " cela va de soit vu la performance du convertisseur analogique numérique qu'il comporte : c'est un ADC **sigma – delta** <sup>(1)</sup> de 24 bits qui permet différentes configurations grâce à la possibilité de sélection de son gain d'amplification ( PGA : programmable gain amplifier) de 1 à 128, l'offset, l'état du buffer, la tension de référence ainsi que la fréquence d'acquisition.

Le composant est aussi muni d'une mémoire SRAM de 256 octets, d'une mémoire flash programme/données de 8 Ko (qui peut être programmé plus d'un million de fois) et d'une belle collection de périphériques tels qu'un SPI (serial port Interface), un I<sup>2</sup>C ( port série deux fils "Philips"), un USART (universal synchronous / Asynchronous receiver transmitter), 16 entrées/sorties digitales ( P1 et P3 :deux ports de 8 bits chacun), deux timers 16bits, plusieurs modes d'horloge ( interne, externe, PLL) et différents filtres numériques (sinc<sup>2</sup>, sinc<sup>3</sup>, etc....).(Voir *Figure IV.1* )



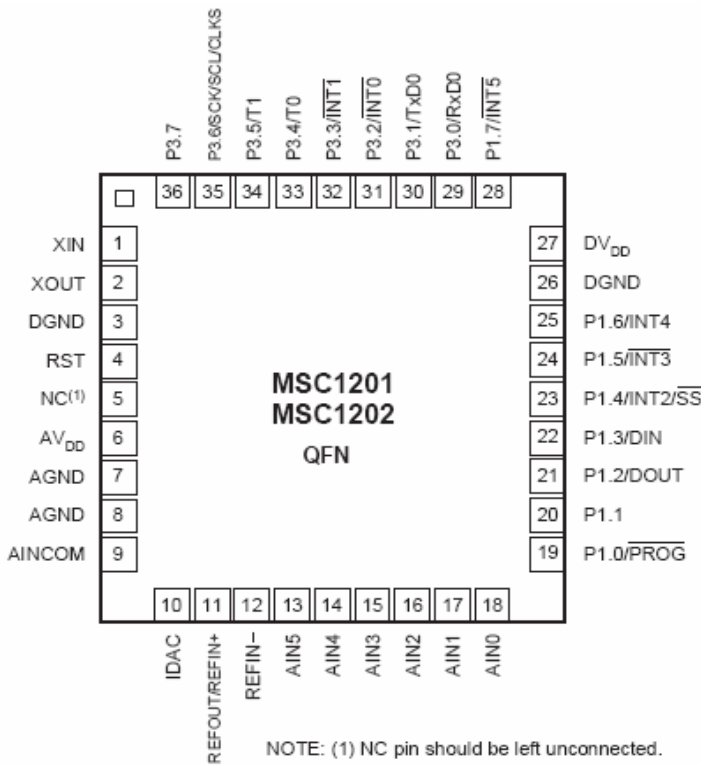
*Figure IV.1 : Schéma bloc du MSC1201 Y3*

Le MSC1201 a un noyau **8051<sup>(2)</sup> optimisé**, les instructions s'exécutent exactement de la même façon que pour un 8051 standard mais à fréquences d'horloges égales il exécute une instruction trois fois plus vite que ce dernier ; autrement dit, il offre les mêmes performances à 33 MHz qu'un 8051 standard à 82,5 MHz cela est dû à son design qui lui assure des plus grandes performances et une consommation moindre. Il est disponible en package **QFN-36<sup>(3)</sup>**: **36 Pins Quad Flat No lead** et son rapport taille performance est plus que satisfaisant ce qui le rend très facile à intégrer (Voir *Figure IV.2*). Ce microcontrôleur est dédié aux mesures haute résolution dans les domaines des émetteurs smart, du contrôle des processus industriels, chromatographie, l'instrumentation, etc.

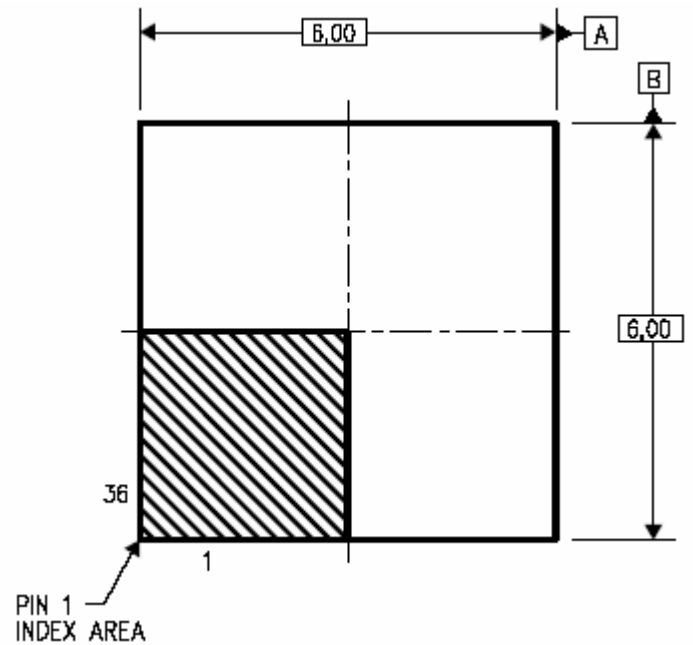
(1) : les convertisseurs delta sigma sont des convertisseurs haute précision car ils sont basés sur le principe du suréchantillonnage. Voir Annexe D et Référence [9].

(2) : Le noyau 8051 standard est expliqué dans la Référence [10].

(3) : C'est package à 36 broches avec un boîtier plastique sans plomb. Voir Annexe E.



a) Le package



b) Les dimensions

Figure IV.2: Package et dimensions du MSC1201 Y3

## IV.2 LE CONVERTISSEUR ANALOGIQUE NUMERIQUE $\Sigma$ - $\Delta$ 24 bits :

### IV.2.1 Structure :

L'ADC a une structure assez complète qui permet au programmeur d'intervenir à plusieurs niveaux pour changer la configuration, la fréquence d'acquisition, les entrées analogiques, l'offset en changeant les valeurs de certains registres internes comme le montre la **Figure IV.3**

La chaîne de conversion se compose des éléments suivants :

- Un **multiplexeur** (ADC Input Multiplexer) pour sélectionner les entrées analogiques de mesure, ceci permet d'obtenir jusqu'à 6 canaux d'entrées différentielles (Voir **Figure IV.4**).

Le réglage du multiplexeur se fait grâce à un registre interne qui se présente comme suit :

#### ADC Input Multiplexer (ADMUX)

	7	6	5	4	3	2	1	0	Reset Value
SFR D7h	INP3	INP2	INP1	INP0	INN3	INN2	INN1	INN0	01h

Les quatre bits de poids le plus fort (INP 3-0) permettent de sélectionner l'entrée positive et ceux de poids le plus faible (INN 3 -0) permettent de sélectionner l'entrée négative suivant les **Tableaux IV.1**

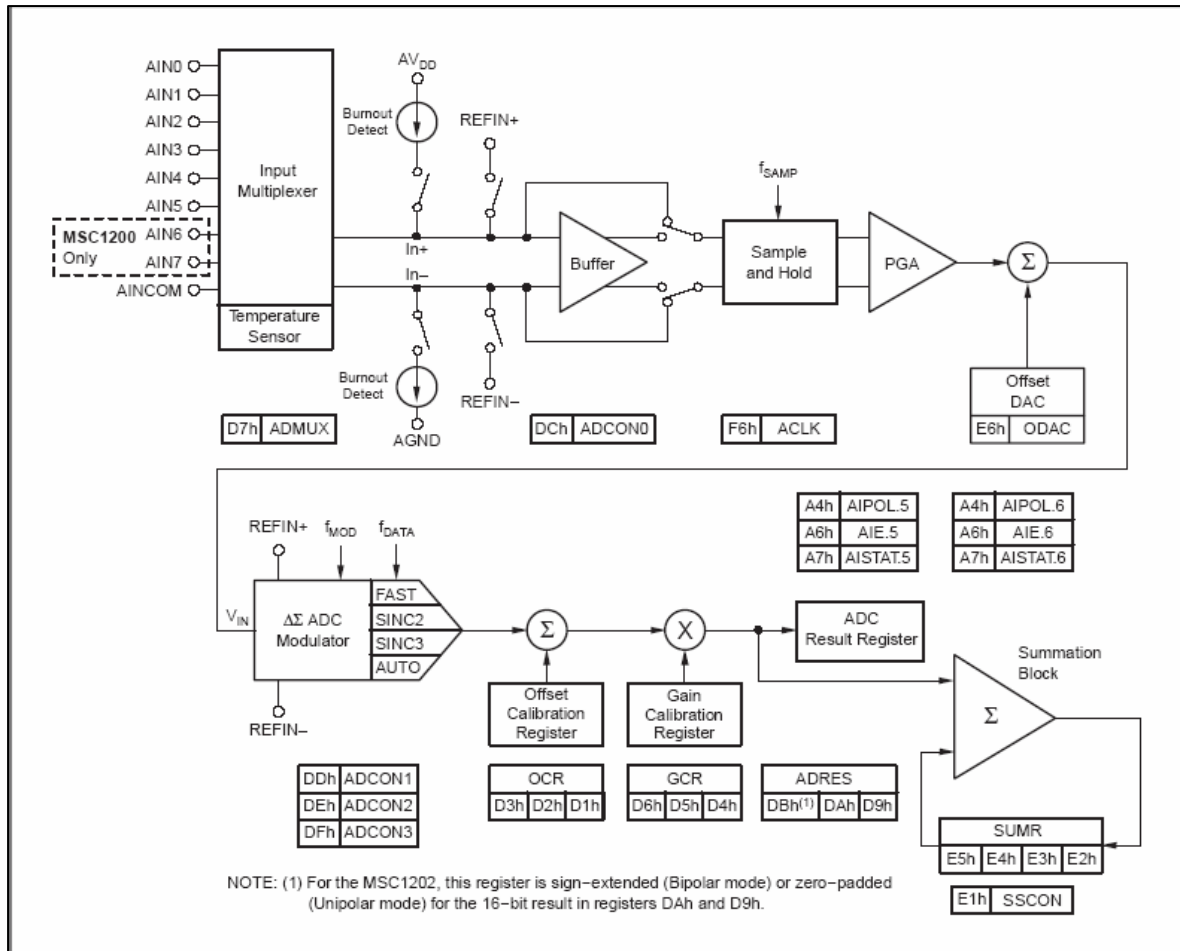


Figure IV.3 : Structure de l'ADC

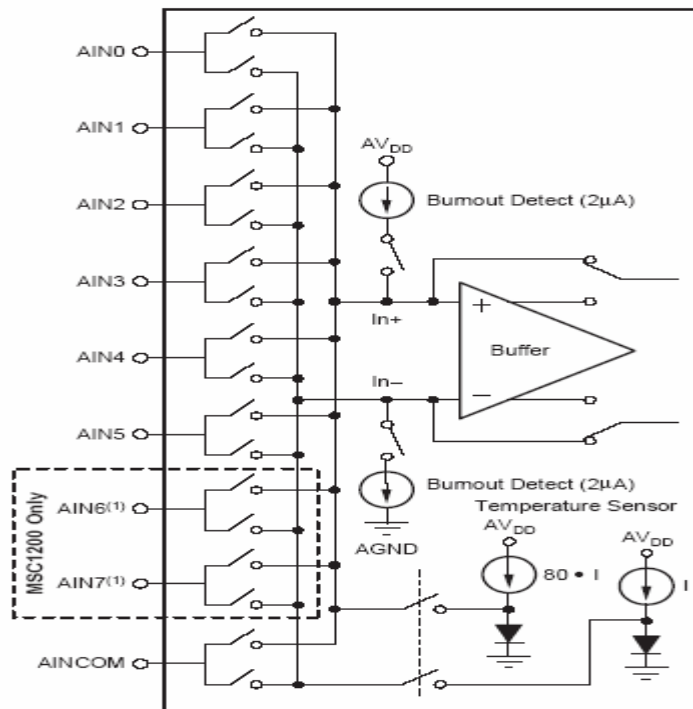


Figure IV.4 : ADC Input Multiplexer

INP3	INP2	INP1	INP0	POSITIVE INPUT
0	0	0	0	AIN0 (default)
0	0	0	1	AIN1
0	0	1	0	AIN2
0	0	1	1	AIN3
0	1	0	0	AIN4
0	1	0	1	AIN5
0	1	1	0	AIN6 (MSC1200 only; for the MSC1201/02, this pin is internally tied to REFIN-)
0	1	1	1	AIN7 (MSC1200 only; for the MSC1201/02, this pin is internally tied to REFIN-)
1	0	0	0	AINCOM
1	1	1	1	Temperature Sensor (requires ADMUX = FFh)

INN3	INN2	INN1	INN0	NEGATIVE INPUT
0	0	0	0	AIN0
0	0	0	1	AIN1 (default)
0	0	1	0	AIN2
0	0	1	1	AIN3
0	1	0	0	AIN4
0	1	0	1	AIN5
0	1	1	0	AIN6 (MSC1200 Only)
0	1	1	1	AIN7 (MSC1200 Only)
1	0	0	0	AINCOM
1	1	1	1	Temperature Sensor (requires ADMUX = FFh)

*Tableau IV.1 : Configuration du multiplexeur des entrées analogiques*

Donc si on veut faire une acquisition entre AIN0 (positif) et AINCOM (négatif) on attribue au registre ADMUX la valeur 0x08 (08<sub>h</sub>).

- Un **Burnout Detect** qui permet de détecter les circuits ouverts (fullscale) et les court circuits (signal faible) en mettant le bit **BOD** à 1 dans le registre **ADCON0** ( Registre de configuration de l'ADC ).

- Un **Buffer** (ADC Input Buffer) : certes, il réduit l'intervalle de tension d'entrée mais il permet d'avoir un courant plus conséquent car quand il est désactivé la résistance d'entrée est élevée (7MΩ/PGA) ; c'est pour cela que l'utilisation du buffer est toujours conseillée. Pour l'activer il faut mettre à 1 le bit **EBUF** du registre **ADCON0**.

- Un **amplificateur de gain programmable** (PGA) : le choix du gain d'amplification entre 1 et 128 permet d'augmenter la résolution des mesures puisqu'avec un PGA = 1 et une référence VREF = 2.5 V la plus petite tension qu'on peut mesurer est 1.5 μV alors qu'avec un PGA=128 et VREF = 19 mV (équivalent à VREF = 2.5V puisque 2.5/128 = 0.019) on peut mesurer des tension allant jusqu'à 75 nV ; on augmente la résolution mais la qualité de la mesure est amoindrie vu que le nombre de bits effectifs ENOB est plus faible.

Le choix du PGA est possible grâce à trois bits **PGA 2-0** du registre **ADCON0** suivant le tableau suivant :

PGA2	PGA1	PGA0	GAIN
0	0	0	1 (default)
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

*Tableau IV.2 : Sélection du PGA*



- Un **Offset DAC** ( source digitale d'une tension d'offset) qui permet de déplacer le centre de l'intervalle de mesure de la moitié de la tension de référence dans les deux sens en augmentant ou en diminuant. Cela est possible en introduisant la valeur adéquate dans le registre **ODAC**. Le bit de poids le plus fort de ce registre est le bit signe ce qui permet de déplacer l'intervalle dans un sens ou dans l'autre.

- Un **Modulateur** de fréquence deuxième ordre qui tourne à la fréquence  $F_{mod}$  ; cette dernière est calculée grâce aux relations suivantes :

$$F_{mod} = \frac{F_{CLK}}{(ACLK + 1).64} = \frac{F_{ACLK}}{64} \quad \text{Où } F_{CLK} : \text{ est la fréquence d'horloge du MCU.}$$

$ACLK$  : un registre interne de 0 à 127.

$$F_{ACLK} = F_{CLK} / (ACLK + 1).$$

Le Modulateur permet de fixer la fréquence d'acquisition  $F_{DATA}$  car elle s'écrit comme suit :

$$F_{DATA} = \frac{F_{mod}}{\text{Decimation Ratio}} \quad \text{Où } F_{mod} : \text{ Fréquence du modulateur.}$$

Decimation Ratio: un rapport entre 1 et 2047.

Le rapport de décimation (Decimation Ratio) peut être réglé grâce au registre ADCON2 et ADCON3 où le premier représente l'octet de poids le plus faible et le deuxième l'octet de poids le plus fort mais seulement les trois derniers bits sont actifs.

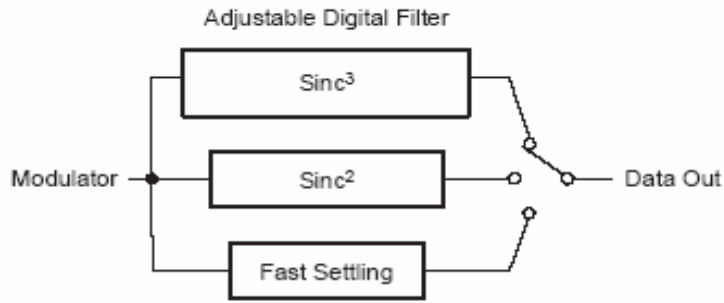
- Des **filtres numériques** qui permettent de réduire le bruit, on peut utiliser le filtre rapide (fast : 1cycle de conversion), le filtre sinc<sup>2</sup> (2 cycles de conversion) ou le filtre sinc<sup>3</sup> (3 cycles de conversion) (Voir *Figure IV.5*); les réponses de ces filtres en fonction de la fréquence de conversion sont représentées dans la *Figure IV.6*.

Même si le choix du filtre numérique Sinc<sup>3</sup> entraîne une conversion plus lente, il reste néanmoins le plus performant car son premier lobe secondaire est plus faible en amplitude que pour les autres filtres. Mais si on veut plus de qualité au niveau réduction de bruit le Msc1201Y3 propose un mode automatique (Auto) qui fait suivre deux filtres rapides par un filtre Sinc<sup>2</sup> puis un filtre Sinc<sup>3</sup> ; donc une conversion dure au moins quatre cycles  $T_{Data}$  comme le montre le *Tableau IV.3* .

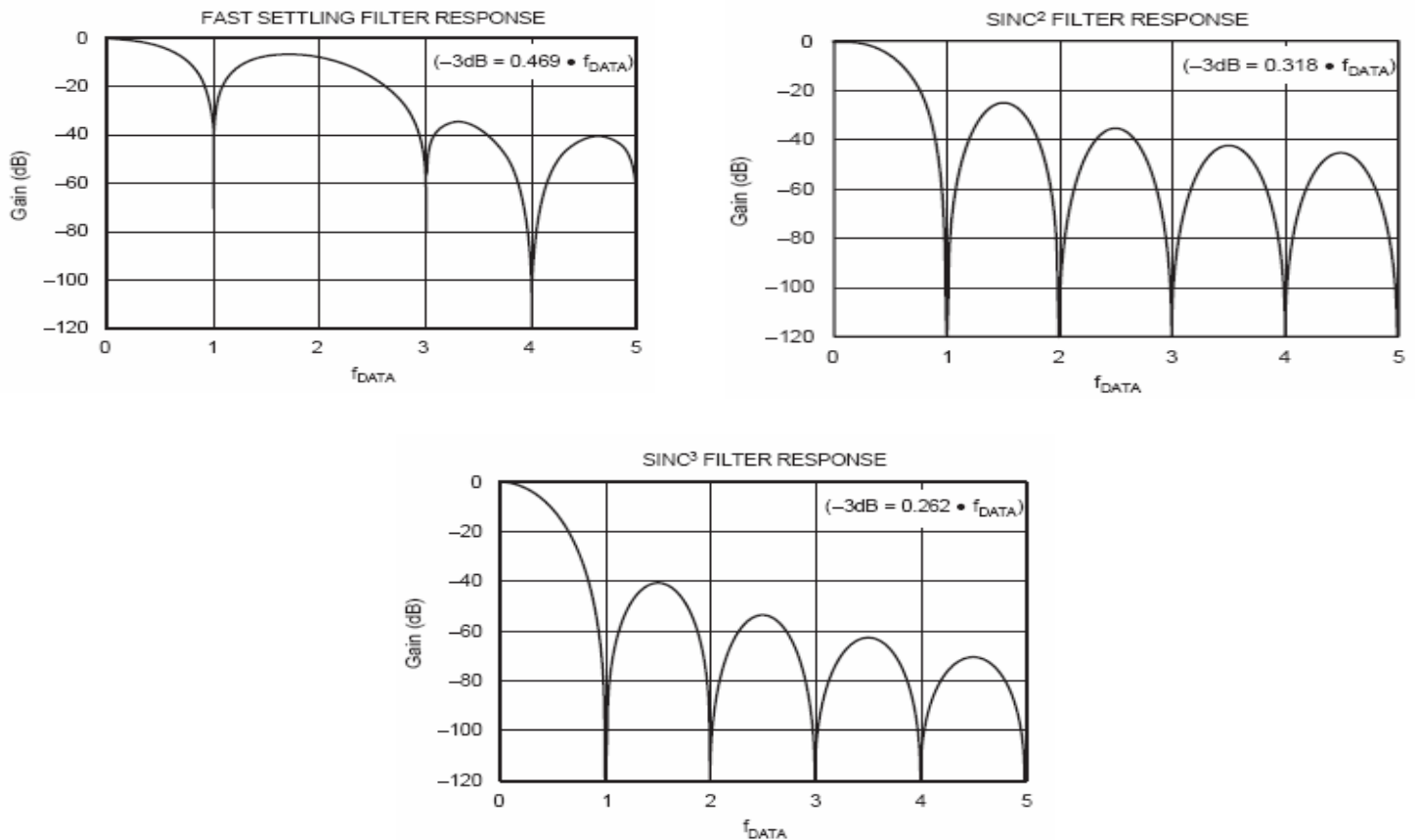
**AUTO MODE FILTER SELECTION**

CONVERSION CYCLE			
1	2	3	4+
Fast	Fast	Sinc <sup>2</sup>	Sinc <sup>3</sup>

*Tableau IV.3 : Succession des filtres pour le mode AUTO*



*Figure IV.5 : Filtres Numériques*



*Figure IV.6 : Réponses fréquentielles des filtres numériques*

On peut sélectionner le filtre numérique désiré grâce aux deux bits **SMI-0** dans le registre **ADCON1** suivant le tableau ci-dessous :

SM1	SM0	SETTLING MODE
0	0	Auto
0	1	Fast Settling Filter
1	0	Sinc <sup>2</sup> Filter
1	1	Sinc <sup>3</sup> Filter

*Tableau IV.4 : sélection des filtres numériques*

- Des **calibrations** qui permettent de corriger les erreurs de PGA et d'offset. Ceci est possible grâce aux bits **CAL2-0** du registre **ADCON1** selon le tableau suivant :

CAL2	CAL1	CAL0	CALIBRATION MODE
0	0	0	No Calibration (default)
0	0	1	Self-Calibration, Offset and Gain
0	1	0	Self-Calibration, Offset only
0	1	1	Self-Calibration, Gain only
1	0	0	System Calibration, Offset only (requires external signal)
1	0	1	System Calibration, Gain only (requires external signal)
1	1	0	Reserved
1	1	1	Reserved

*Tableau IV.5 : Sélection du mode de calibration*

Le seul inconvénient reste qu'une calibration a besoin de 7 cycles de conversion  $T_{Data}$  pour faire les corrections, donc une calibration de gain et d'offset prendra 14 cycles de conversions.

- un système de sélection de **tension de référence** VREF qui donne le choix à l'utilisateur entre deux modes de fonctionnement : en référence interne ou en référence externe.

- En référence interne, la tension de référence est générée à l'intérieur du microcontrôleur Le MSC1201Y3 permet d'avoir soit 2.5V ou 1.25V en référence interne ; le choix se fait en mettant le bit **VREFH** du registre **ADCON0** à zéro pour  $VREF = 1.25V$  ou à un pour  $VREF=2.5V$ .
- En référence externe, la tension de référence est fournie par une source extérieure entre les pins REFIN+ et REFIN- du microcontrôleur , la tension maximale acceptée par le composant est 2.5 V et il est préférable qu'elle soit stable et à faible bruit. Le bit **EVREF** du registre **ADCON0** permet de choisir entre un fonctionnement en référence interne ou externe : s'il est à 0 la référence externe est sélectionnée par contre s'il est à 1 c'est la référence interne qui l'est; il suffit seulement de choisir laquelle utiliser.

### V.1.2 Registres de configuration et récupération des résultats :

Maintenant qu'on a décrit la chaîne de conversion et que l'on a plus d'informations sur la configuration de chaque élément, on présente quelques registres de configuration de l'ADC pour avoir surtout l'ordre des bits expliqués précédemment dans le registre afin de programmer aisément.

- Le registre **ADCON0** :

	7	6	5	4	3	2	1	0	Reset Value
SFR DCh	—	BOD	EVREF	VREFH	EBUF	PGA2	PGA1	PGA0	30h

Tous les bits ont été expliqués précédemment il faut juste faire attention à l'ordre des bits et le code correspondant à chaque configuration.

- Le registre **ADCON1**:

	7	6	5	4	3	2	1	0	Reset Value
SFR DDh	OF_UF	POL	SM1	SM0	—	CAL2	CAL1	CAL0	00h

Il n'y a que les bits **OF\_UF** et **POL** qui n'ont pas été expliqués au préalable ; le premier (**OF\_UF** : Overflow / Underflow) se met à 1 en cas de dépassement de capacité ou de données manquantes, le deuxième (**POL** : polarity) permet de choisir la polarité de la mesure soit bipolaire (et on le met à 0) soit unipolaire (et on le met à 1).

Les résultats de la conversion se présentent sur 24 bits ; pour les récupérer il faut aller les chercher dans trois registres qui représentent l'octet de poids faible, l'octet de poids moyen et l'octet de poids fort respectivement. Ces registres sont **ADRESL** (ADC Results Low byte); **ADRESM** (ADC Results Middle byte) ET **ADRESH** (ADC Result High byte) respectivement. Pour obtenir le résultat entier en Volt il faut faire les décalages adéquats pour avoir le nombre (code) correct puis le multiplier par l'échelle correspondant à la tension de référence utilisée.

### IV.3 LA SOURCE DE COURANT DIGITALE IDAC :

Le MSC1201 Y3 héberge une source de courant digitale de 8 bits IDAC, elle est qualifiée par le terme digitale car elle est générée en interne et son intensité peut être réglée par la simple programmation d'un registre interne dans notre cas c'est **IDAC** ; cette source peut être utilisée pour des mesures ratiométriques<sup>(1)</sup> par exemple. L'IDAC a sa propre référence interne qui n'a aucune relation avec la tension de référence de l'ADC et son intensité est régie par l'équation suivante :

$$IDAC_{OUT} \mu A \approx IDAC \cdot 3.9 \mu A \quad \text{à } 25 \text{ } ^\circ C$$

Où  $IDAC_{OUT}$  : est le courant de sortie en  $\mu A$ .

**IDAC** : est un registre interne 8 bits dont la valeur est dans l'intervalle [0-255].

3.9  $\mu A$  : est la résolution de l'IDAC.

Ainsi la valeur maximale que peut débiter cette source est 1mV environ.

### IV.4 LES MEMOIRES :

Le MSC1201 contient trois types de mémoires : une mémoire flash de 8 KO, une mémoire vive statique SRAM ( Scratchpad Random Access Memory) de 256 O ; et une ROM de démarrage ( Boot ROM).

- La mémoire flash est utilisée aussi bien pour les données que pour les programmes, cependant l'exécution d'un programme en mémoire flash ne peut se faire que s'il est dans la

(1) : mesures ratiométriques : mesures de rapports

partie mémoire programme c'est pour cela que le partage de la mémoire flash en mémoire programme et mémoire données est programmable ; il est fait grâce au registre de configuration hardware **HCR0** dans la partie Configuration memory **DFSEL** (Voir *Tableau IV.6*). Ce registre ainsi que deux autres registres de configuration hardware **HCR1** et **HCR2** ne peuvent être changé au sein d'un programme directement car ils résident dans la mémoire de configuration. Mais ils sont ajoutés au projet sous forme de fichier; en général c'est le fichier **HWCfr.a51** qui est ajouté et le changement des paramètres dans ce fichier permet de configurer pratiquement tout le hardware du MSC1201.

HCR0	MSC120xY2		MSC120xY3	
DFSEL	PM	DM	PM	DM
00	2kB	2kB	4kB	4kB
01	2kB	2kB	6kB	2kB
10	3kB	1kB	7kB	1kB
11 (default)	4kB	0kB	8kB	0kB

*Tableau IV.6 : Partage de la mémoire flash entre Mémoire Programme (PM) et Mémoire Données (DM) .*

- La mémoire de configuration se compose de 128 octets divisés en deux parties équivalentes de 64 octets : une partie basse qui se localise dans l'intervalle d'adresses (8000<sub>h</sub>-803F<sub>h</sub>) et une partie haute qui a pour adresses l'intervalle (8040<sub>h</sub>-807F<sub>h</sub>). La partie basse autorise la lecture et l'écriture, elle héberge les registres de configuration hardware **HCR0**, **HCR1** et **HCR2** qui se sont localisés par les adresses 803F<sub>h</sub>, 803E<sub>h</sub> et 803D<sub>h</sub> respectivement et ils se présentent sous la forme suivante :

#### Hardware Configuration Register 0 (HCR0)

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
CADDR 3Fh	EPMA	PML	RSL	EBR	EWDR	1	DFSEL1	DFSEL0

Avec : **EPMA** : Activer l'Accès à la mémoire programme (Enable Program Memory Access). Mis à 1 ce bit permet l'accès total à la mémoire Programme; dans le cas contraire l'accès ne peut avoir lieu qu'après un effacement total de la mémoire programme.

**PML** : Protection de la mémoire programme (Program Memory Lock)  
Quand ce bit est à 0 le programmeur a le droit de lecture et d'écriture dans la mémoire programme par contre s'il est à 1 le programmeur n'a que l'autorisation de lire.

**RSL** : Mis à 0 ce bit autorise la programmation par RESET SECTOR qui est une autre méthode de programmer la mémoire programme en la mettant à jour et en lançant le moniteur et les routines de programmation. Cependant mis à 1 il a le même effet que **PML** mais seulement sur 4 KO.

**EBR** : Activer la ROM de démarrage (Enable Boot ROM). À 1 ce bit active la ROM de démarrage, à 0 il la désactive.

**EWDR** : Mis à 0 ce bit autorise le RESET par watchdog timer ; dans le cas contraire il le désactive.

**DFSEL1-0**: Ces bits permettent le partage de la mémoire flash entre mémoire programme et mémoire données comme spécifié précédemment.

#### Hardware Configuration Register 1 (HCR1)

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
CADDR 3Eh	DBSEL3	DBSEL2	DBSEL1	DBSEL0	1	DDB	1	1

Où **DBSEL 3-0** : Sélectionner l'alimentation du brownout select (Digital Supply Brownout Select). Les valeurs de l'alimentation du brownout dépendent de la fréquence d'horloge et de l'alimentation globale du microcontrôleur. Cela dit les valeurs listés ci-dessous sont nominales et pour les hautes fréquences d'horloge la variation peut être de 10% autour de la valeur nominale.

<b>DBSEL</b>	<b>Tension nominale</b>
0000	4.6 V
0001	4.2 V
0010	3.8 V
0011	3.6 V
0100	3.3 V
0101	3.1 V
0110	2.9 V
0111	2.7 V
1000	2.6 V
De 1001 à 1111	Reserved

*Tableau IV.7 : Sélection de l'alimentation du Brownout.*

**DDB** : Désactiver la détection du Brownout digitale (Disable Digital Brownout Detection). Quand ce bit est à 1 la détection est activée dans le cas contraire elle est désactivée.

#### Hardware Configuration Register 2 (HCR2)

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
CADDR 3Dh	0	0	0	0	0	CLKSEL2	CLKSEL1	CLKSEL0

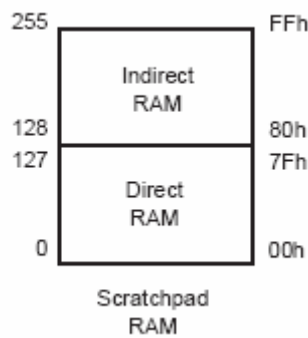
Où **CLKSEL 2-0** : le Clock Select . La sélection de l'horloge se fait comme suit :

<b>CLKSEL</b>	<b>CLK MODE</b>
De 000 à 010	Réservés
011	Horloge Externe
100	PLL haute fréquence
101	PLL basse fréquence
110	Oscillateur interne haute fréquence
111	Oscillateur interne basse fréquence

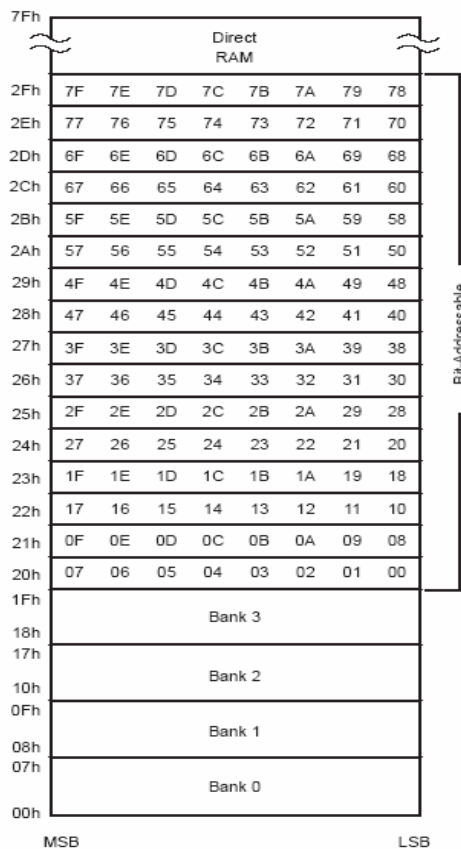
*Tableau IV.8 : Sélection de l'horloge.*

Par contre la partie haute de la mémoire de configuration n’autorise que la lecture car on y trouve des informations telles que le numéro de série du composant (Serial Number) écrit sur 4 octets à partir de l’adresse 8050<sub>h</sub>.

- La SRAM est elle aussi partagée en deux parties égales de 128 octets : Indirect RAM et Direct RAM (Voir *Figure IV.7*). La RAM Indirecte, si on peut l’appeler ainsi, ne peut être adressée qu’indirectement et contient des données à usage général ; alors que la RAM directe est adressée directement et contient quelques cases mémoires à usage spécifique et quatre banques mémoires qui contiennent les registres de travail (Working Registers) **R0 – R7** (Voir *Figure IV.8*) ; ainsi à chaque fois qu’une instruction qui implique les registres de travail est exécutée, l’accès direct à la Direct RAM se fait systématiquement.



*Figure IV.7 : Partage de la SRAM*



*Figure IV.8 : Contenu de la Direct RAM*

- La ROM de démarrage (Boot ROM) contrôle les opérations pendant la programmation à travers le port série ; de plus des routines qui ont été établies pour faciliter la programmation y sont stockées (Boot ROM Routines). Ces Routines sont listées dans le tableau ci-dessous

HEX ADDRESS	ROUTINE	C DECLARATIONS	DESCRIPTION
F802	sfr_rd	char sfr_rd(void);	Return SFR value pointed to by CADDR <sup>(1)</sup>
F805	sfr_wr	void sfr_wr(char d);	Write to SFR pointed to by CADDR <sup>(1)</sup>
FBD8	monitor_isr	void monitor_isr() interrupt 6;	Push registers and call cmd_parser
FBDA	cmd_parser	void cmd_parser(void);	See application note SBAA076, <i>Programming the MSC1210</i> , available at www.ti.com.
FBDC	put_string	void put_string(char code *string);	Output string
FBDE	page_erase	char page_erase(int faddr, char fdata, char fdm);	Erase flash page
FBE0	write_flash	Assembly only; DPTR = address, ACC = data	Flash write <sup>(2)</sup>
FBE2	write_flash_chk	char write_flash_chk(int faddr, char fdata, char fdm);	Write flash byte, verify
FBE4	write_flash_byte	void write_flash_byte(int faddr, char fdata);	Write flash byte <sup>(2)</sup>
FBE6	faddr_data_read	char faddr_data_read(char faddr);	Read byte from Configuration Memory
FBE8	data_x_c_read	char data_x_c_read(int faddr, char fdm);	Read xdata or code byte
FBEA	tx_byte	void tx_byte(char);	Send byte to USART0
FBEC	tx_hex	void tx_hex(char);	send hex value to USART0
FBEF	putc	void putc(void);	send "x" to USART0 on R7 = 1
FBF0	rx_byte	char rx_byte(void);	Read byte from USART0
FBF2	rx_byte_echo	char rx_byte_echo(void);	Read and echo byte on USART0
FBF4	rx_hex_echo	char rx_hex_echo(void);	Read and echo hex on USART0
FBF6	rx_hex_dbl_echo	int rx_hex_dbl_echo(void);	Read int as hex and echo: USART0
FBF8	rx_hex_word_echo	int rx_hex_word_echo(void);	Read int reversed as hex and echo: USART0
FBFA	autobaud	void autobaud(void);	Set USART0 baud rate after CR <sup>(3)</sup> received
FBFC	putspace1	void putspace1(void);	Output 1 space to USART0
FBFE	putcrlf	void putcrlf(void);	Output CR, LF to USART0

Tableau IV.9 : Boot ROM Routines du MSC1201.

Après avoir présenté toutes les mémoires séparément il est normal de les regrouper dans la Memory MAP ( Schéma de la mémoire) qui se présente comme suit :

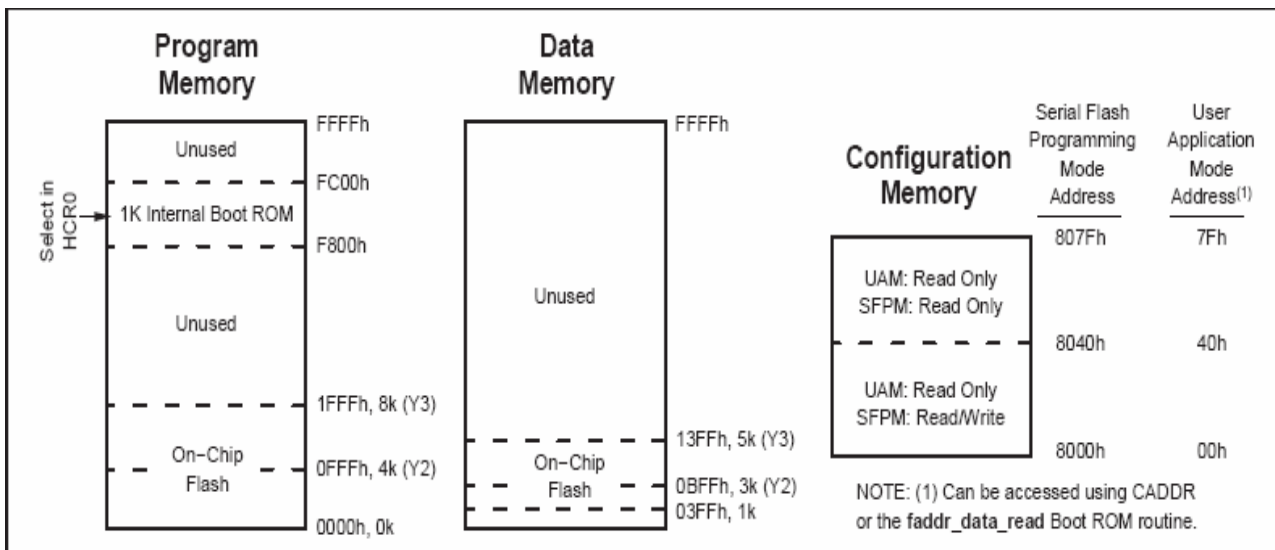


Figure IV.9 : Memory Map



## IV.5 LES PORTS DE COMMUNICATION :

Le MSC1201Y3 met à la disposition de l'utilisateur trois ports série : un USART, un SPI et un I<sup>2</sup>C plus deux ports d'entrées/sorties de 8 bits chacun appelés P1 et P3. On s'intéressera dans cette partie principalement à l'USART<sup>(1)</sup> (Port 0) et les deux ports P1 et P3, on ne s'intéressera pas vraiment au SPI et I<sup>2</sup>C dans ce cas car ils n'interviendront pas de façon majeure dans notre application qui vise à évaluer ce microcontrôleur en haute température.

### IV.5.1 L' USART (Universal Synchronous / Asynchronous Receiver Transmitter) :

L'USART est une liaison série synchrone/ Asynchrone ; elle est synchrone si l'émetteur et le récepteur disposent de la même référence d'horloge pour communiquer (envoi d'informations qui permettent d'éliminer le déphasage entre l'émetteur et le récepteur). Dans le cas contraire la communication est asynchrone.

Sur la carte Msc1201EVM l'USART est directement relié à un connecteur RS232 et sa configuration se fait grâce au registre **SCON0** de la manière suivante:

#### Serial Port 0 Control (SCON0)

	7	6	5	4	3	2	1	0	Reset Value
SFR 98h	SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0	00h

Où **SM 0-2** : définissent le mode de fonctionnement de l'USART ( Serial Port 0 Mode).

MODE	SM0	SM1	SM2	FUNCTION	LENGTH	PERIOD
0	0	0	0	Synchronous	8 bits	12 pCLK <sup>(1)</sup>
0	0	0	1	Synchronous	8 bits	4 pCLK <sup>(1)</sup>
1	0	1	0	Asynchronous	10 bits	Timer 1 Baud Rate Equation
1	0	1	1	Asynchronous-Valid Stop Required <sup>(2)</sup>	10 bits	Timer 1 Baud Rate Equation
2	1	0	0	Asynchronous	11 bits	64 pCLK <sup>(1)</sup> (SMOD = 0) 32 pCLK <sup>(1)</sup> (SMOD = 1)
2	1	0	1	Asynchronous with Multiprocessor Communication	11 bits	64 pCLK <sup>(1)</sup> (SMOD = 0) 32 pCLK <sup>(1)</sup> (SMOD = 1)
3	1	1	0	Asynchronous	11 bits	Timer 1 Baud Rate Equation
3	1	1	1	Asynchronous with Multiprocessor Communication <sup>(3)</sup>	11 bits	Timer 1 Baud Rate Equation

(1) pCLK will be equal to tCLK, except that pCLK will stop for Idle mode.  
(2) RI\_0 will only be activated when a valid STOP is received.  
(3) RI\_0 will not be activated if bit 9 = 0.

**Tableau IV.10** : Sélection du mode de fonctionnement du Port 0

Pour les modes 1,2 et 3 un bit START et un bit STOP sont ajoutés à chaque trame de 8 ou 9 bits (cela dépend du bit RI\_0).

**REN\_0** : Autoriser la réception (Receive Enable). Quand il est à 1, ce bit autorise la réception en activant l'accès au registre de réception ; par contre s'il est à 0 il interdit l'accès au registre de réception et empêche la réception de données au moyen de l'USART.

**TB8\_0** : ce bit définit l'état du 9<sup>ème</sup> bit transmis par le port 0 pour les modes 2 et 3.

**RB8\_0** : ce bit identifie l'état du 9<sup>ème</sup> bit réceptionné par le port 0 pour les modes 2 et 3 complètement vide après une phase de transmission ( toute la donnée est envoyée).

(1) : Pour comprendre le fonctionnement de l'USART il faut consulter l'Annexe F.

**TI\_0** : (Transmitter Interrupt Flag). Ce bit indique si le buffer du port 0 est complètement vide après une phase de transmission ( toute la donnée a été envoyée).

**RI\_0** : (Receiver Interrupt Flag). Ce bit indique si un octet, au moins, de données a été reçu dans le buffer du port 0.

Les Données sont envoyées et réceptionnées dans le Buffer 8 bits de l'USART **SBUF0**. Ainsi pour envoyer des données en langage C il faut juste affecter **SBUF0** à une variable et la lire, une lecture directe à partir du registre n'est pas possible et réciproquement pour envoyer une donnée on affecte une donnée au buffer. On peut cependant utiliser deux routines stockées dans la Boot ROM : **Tx\_Byte** et **Rx\_Byte** pour l'émission et la réception respectivement.

#### IV.5.2 Les ports d'entrées/sorties P1( Port 1) et P3 ( Port 3) :

Les port 1 et 3 sont des entrées / sorties à usage général à priori mais ils peuvent être utilisés pour déclencher des interruptions externes ou effectuer des mesures de grandeurs propres au microcontrôleur comme l'horloge interne par exemple.

Chaque pin de ces ports peut être configurée en entrée, en sortie ou en standard 8051 et selon sa configuration, il devient soit à usage général ou spécifique. Configurer les ports se fait en programmant les registres **P1DDRL** et **P1DDRH** concernant P1, de même, **P3DDRL** et **P3DDRH** concernant P3. Ces registres se présentent sous la forme suivante :

##### Port 1 Data Direction Low (P1DDRL)

	7	6	5	4	3	2	1	0	Reset Value
SFR AEh	P13H	P13L	P12H	P12L	P11H	P11L	P10H	P10L	00h

##### Port 1 Data Direction High (P1DDRH)

	7	6	5	4	3	2	1	0	Reset Value
SFR AFh	P17H	P17L	P16H	P16L	P15H	P15L	P14H	P14L	00h

On remarque que **P1DDRL** permet de configurer les quatre premières pins (de poids faible) et **P1DDRH** les quatre autres pins (de poids forts) selon le tableau suivant :

PiH	PiL	
0	0	Standard 8051
0	1	CMOS Output
1	0	Open Drain Output
1	1	Input

Avec **i** un entier entre 0 et 7.

Tableau IV.11 : Configuration des entrées / sorties

La configuration du port 3 se fait exactement de la même manière que le port 1, il faut juste remplacer **P1DDRL** par **P3DDRL** et **P1DDRH** par **P3DDRH**.

Il faut maintenant connaître la fonction de chaque pin en étudiant les registres **P1** et **P3**.

### Port 1 (P1)

	7	6	5	4	3	2	1	0	Reset Value
SFR 90h	$\overline{P1.7}$ $\overline{INT5}$	P1.6 INT4	$\overline{P1.5}$ $\overline{INT3}$	P1.4 INT2/ $\overline{SS}$	P1.3 DIN	P1.2 DOUT	P1.1	$\overline{P1.0}$ $\overline{PROG}$	FFh

Où **P1.7-0** : dans ce cas le port a le comportement d'entrées / sorties classique.

$\overline{INT5}$  : Interruption externe 5; elle est déclenchée par un front descendant sur cette pin si elle est autorisée.

**INT4** : Interruption externe 4; elle est déclenchée par un front montant sur cette pin si elle est autorisée.

$\overline{INT3}$  : Idem que  $\overline{INT5}$  sur la pin P1.5

**INT2/  $\overline{SS}$**  : Idem que **INT4** sur la pin P1.4 et peut être utilisée pour sélectionner le mode esclave de l'SPI (Slave Select).

**DIN** : Réception de données des ports SPI et I<sup>2</sup>C ( Serial Data In ).

**DOUT** : Transmission de données à travers les ports SPI et I<sup>2</sup>C ( Serial Data Out ).

$\overline{PROG}$  : (Program Mode). Quand ce bit est à 0 en power on le composant entre en mode Programmation.

### Port 3 (P3)

	7	6	5	4	3	2	1	0	Reset Value
SFR B0h	P3.7	P3.6 SCK/SCL/CLKS	P3.5 T1	P3.4 T0	$\overline{P3.3}$ $\overline{INT1}$	$\overline{P3.2}$ $\overline{INT0}$	P3.1 TXD0	P3.0 RXD0	FFh

Où **P3.7-0** : Idem que **P1.7-0**.

**SCK/SCL/CLKS** : Grâce à ce bit on peut mesurer les différentes fréquences d'horloge utilisées par le microcontrôleur; la sélection se fait en programmant le registre PASEL Comme suit :

### $\overline{PSEN/ALE}$ Select (PASEL)

	7	6	5	4	3	2	1	0	Reset Value
SFR F2h	PSEN4	PSEN3	PSEN2	PSEN1	PSEN0	0	0	0	00h

**PSEN4-0** : Ces 5 bits permettent de définir la sortie P3.6 selon le tableau Suivant :

PSEN4	PSEN3	PSEN2	PSEN1	PSEN0	Sortie P3.6
0	0	0	0	0	General Purpose I/O
0	0	0	0	1	SYSCLK
0	0	0	1	1	Internal PSEN
0	0	1	0	1	Internal ALE
0	0	1	1	1	F <sub>OSC</sub> (buffered X <sub>IN</sub> oscillator clock)
0	1	0	0	1	Memory WR
0	1	0	1	1	T0 Out
0	1	1	0	1	T1 Out
0	1	1	1	1	F <sub>MOD</sub>
1	0	0	0	1	SYSCLK/2
1	0	0	1	1	Internal PSEN/2
1	0	1	0	1	Internal ALE/2
1	0	1	1	1	F <sub>OSC</sub>
1	1	0	0	1	Memory WR/2
1	1	0	1	1	T0 Out/2
1	1	1	0	1	T1 Out/2
1	1	1	1	1	F <sub>MOD</sub> /2

*Tableau IV.12 : choix de la grandeur de sortie sur la broche P3.6*

**T1** : (Timer / Counter 1 External Input). Une transition de 1 à 0 sur cette broche incrémente le timer 1.

**T0** : (Timer /Counter 0 External Input). Idem que **T1**.

$\overline{INT1}$  : Interruption Externe 1. Un front descendant ou un niveau bas sur cette pin déclenche l'interruption externe 1.

$\overline{INT0}$  : Interruption Externe 0. Idem que pour  $\overline{INT1}$ .

**TXD0** : (Serial Port 0 Transmit). Cette broche transmet les données séries du port 0 pour les modes 1, 2, 3 et l'horloge de synchronisation pour le mode 0.

**RXD0** : (Serial Port 0 Receive). Cette pin permet de recevoir les données séries du port 0 pour les modes 1, 2, 3 et permet un transfert bidirectionnel pour le mode 0.

#### **IV.6 LES HORLOGES :**

Le MSC 1201Y3 offre cinq modes d'horloges différents :

- Horloge Externe (External Clock Mode).
  - Oscillateur Interne Haute Fréquence (Internal RC Oscillator High-Frequency Mode).
  - Oscillateur Interne Basse Fréquence (Internal RC Oscillator Low-Frequency Mode).
  - PLL Haute Fréquence (PLL High-Frequency).
  - PLL Basse Fréquence (PLL Low-Frequency).
- } Dans le cas d'un cristal (quartz).

Le choix d'un de ces modes se fait grâce au registre **HCR2** comme expliqué précédemment. Lorsqu'un mode d'horloge est choisi, toutes les fréquences indispensables pour le fonctionnement des périphériques, timers, ADC, etc. se calculent à partir de ce mode comme le montre la *Figure IV.10*.

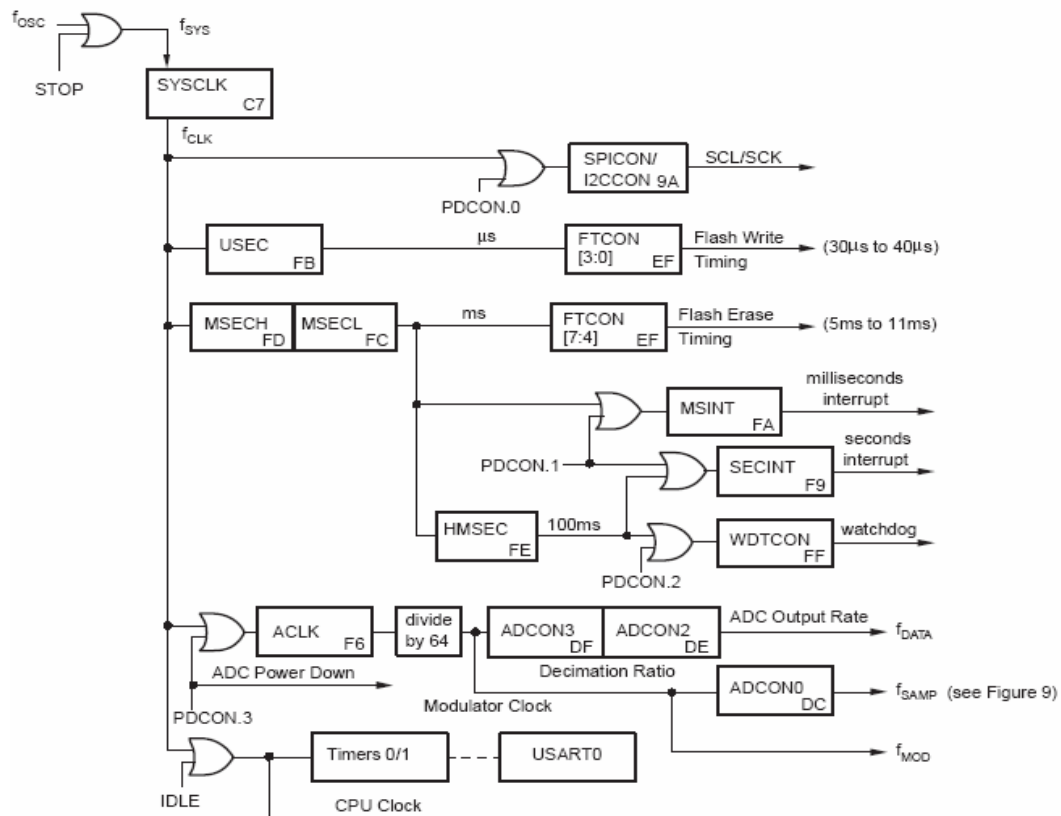


Figure IV.10 : Fréquences présentes au sein du microcontrôleur

Comme le montre la figure ci-dessus la fréquence primaire, correspondant au choix du mode d’horloge, peut être divisée par 2, 4, 8, 16, 32, 1024, 2048 et 4096 en programmant le registre **SYSCLK** de la manière suivante :

**System Clock Divider (SYSCLK)**

	7	6	5	4	3	2	1	0	Reset Value
SFR C7h	0	0	DIVMOD1	DIVMOD0	0	DIV2	DIV1	DIV0	00h

Où **DIVMOD1-0** : permettent de choisir le mode de division ( Clock Divide Mode) suivant le tableau ci-dessous :

<b>DIVMOD</b>	<b>Mode de Division</b>
00	<u>Mode Normal</u> : Pas de division.
01	<u>Mode Immédiat</u> : la division se fait immédiatement et ne s’arrête qu’après un IDLE mode ou intervention du programmeur.
10	<u>Mode Retardé</u> : Ce mode est conditionné par une interruption spéciale du timer (Milliseconds Interrupt); le début ou l’arrêt de la division se fait après chaque interruption.
11	<u>Mode Manuel</u> : même chose que le mode immédiat sauf que l’arrêt de division ne se fait que par écriture du programmeur.

Tableau IV.13 : Choix du mode de division de Clock.

**DIV2-0** : Ces trois bits permettent de choisir le diviseur de la fréquence d'horloge; comme le montre le tableau suivant :

<b>DIV</b>	Diviseur	Nouvelle Fréquence $F_{CLK}$
000	Division par 2	$F_{CLK} = F_{SYS} / 2$
001	Division par 4	$F_{CLK} = F_{SYS} / 4$
010	Division par 8	$F_{CLK} = F_{SYS} / 8$
011	Division par 16	$F_{CLK} = F_{SYS} / 16$
100	Division par 32	$F_{CLK} = F_{SYS} / 32$
101	Division par 1024	$F_{CLK} = F_{SYS} / 1024$
110	Division par 2048	$F_{CLK} = F_{SYS} / 2048$
111	Division par 4096	$F_{CLK} = F_{SYS} / 4096$

**Tableau IV.14** : Programmation du diviseur d'horloge

Les registres, en relation avec l'horloge, les plus importants qu'on va utiliser lors de la programmation dans le but d'évaluer le microcontrôleur tels que **ACLK**, **HCR2**, etc.... ont déjà été traités précédemment.

#### **IV.7 LES TIMERS :**

Comme cité au début de cette étude du composant, le MSC1201Y3 contient deux timers de 16 bits chacun appelés Timer 0 et Timer 1. Pour accéder ou modifier son contenu, chaque timer a deux registres représentant l'octet de poids fort **THi** (i = 0 ou 1) et l'octet de poids faible **TLi**(i=0 ou 1). On obtient donc au sein du microcontrôleur quatre registres : deux pour le Timer 0 : (**TL0**, **TH0**) et deux pour le Timer 1 : (**TL1**, **TH1**).

La configuration des deux timers se fait grâce à deux registres internes **TCON** et **TMOD**; le premier permet de contrôler le fonctionnement des timers alors que le deuxième permet de sélectionner leurs modes de fonctionnement.

##### **Timer/Counter Control (TCON)**

	7	6	5	4	3	2	1	0	Reset Value
SFR 88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00h

Où **TF1** : (Timer1 Overflow Flag). Ce bit se met à 1 quand le Timer 1 est en dépassement de capacité ( maximum de comptage).

**TR1** : (Timer 1 Run Control). Ce bit permet d'autoriser (état haut) ou d'interdire ( état bas) les opérations du Timer 1.

**TF0** : Ce bit est pour le Timer 0 ce qu'est **TF1** pour le Timer 1.

**TR0** : Ce bit est pour le Timer 0 ce qu'est **TR1** pour le Timer 1.

**IE1** : ( Interrupt 1 Edge Detect). Ce bit se met à 1 quand un front ou un niveau déclenchant l'interruption externe 1 est détecté.

**IT1** : ( Interrupt 1 Type Select). Ce bit permet de choisir entre un niveau bas ( IT1=0) et un front descendant ( IT1=1) comme évènement déclencheur de l'interruption externe 1 quand elle est autorisée.

**IE0** : Ce bit est pour l'interruption externe 0 ce qu'est **IE1** pour l'interruption externe 1.

**IT0** : Ce bit est pour l'interruption externe 0 ce qu'est **IT1** pour l'interruption externe 1.

**Timer Mode Control (TMOD)**

	7	6	5	4	3	2	1	0	Reset Value
SFR 89h	TIMER 1				TIMER 0				00h
	GATE	C/T	M1	M0	GATE	C/T	M1	M0	

Où les quatre bits de poids fort permettent de choisir le mode de fonctionnement du Timer1 et les quatre bits de poids faible permettent identiquement de choisir le mode de fonctionnement du Timer0 ; ainsi :

**GATE** : ( Timer Gate control). Ce bit permet de définir les conditions d'incréméntation des timers. Car quand il est mis à 0; l'incréméntation des timers se fait quand **TRi** = 1 indépendamment de  $\overline{INTi}$  . Par contre quand il est à 1, l'incréméntation se fait juste lorsque **TRi**=1 et  $\overline{INTi}$  = 1 ( i = 0 ou 1).

$\overline{C/T}$  : (Timer / Counter Select). Mis à 0, ce bit permet de synchroniser l'incréméntation du timer avec l'horloge interne. Alors qu'à 1 le timer est incréménté par des fronts sur la pin **Ti** quand **TRi**=1 ( i = 0,1).

**MI-0** : ( Timer Mode Select). Ces bits permettent de sélectionner le mode de fonctionnement des timers selon le tableau suivant :

M1	M0	MODE
0	0	Mode 0: 8-bit counter with 5-bit prescale.
0	1	Mode 1: 16 bits.
1	0	Mode 2: 8-bit counter with auto reload.
1	1	Mode 3: Timer 1 is halted, but holds its count.

**Tableau IV.15** : Modes de fonctionnement des timers.

Les timers peuvent aussi être sensibles à une horloge en quartz ( cas du mode PLL) et ils s'incrémentent pour des divisions de la fréquence du cristal; cela grâce au registre **CKCON** qui se présente sous la forme suivante :

**Clock Control (CKCON)**

	7	6	5	4	3	2	1	0	Reset Value
SFR 8Eh	0	0	0	T1M	T0M	MD2	MD1	MD0	01h

Où **T1M** : (Timer 1 Clock Select). Ce bit permet de sélectionner la division de la fréquence du cristal qui pilote le Timer 1. Quand il est à l'état bas le Timer 1 est sensible à une division par 12 de la fréquence du cristal alors que quand il est à l'état haut le Timer 1 est sensible à une division par 4 du cristal.

**T0M** : Ce bit est au Timer 0 ce qu'est **T1M** au Timer 1.

**MD2-0** : (Stretch MOVX Select). Ces bits permettent de sélectionner le temps ajouté à l'accès d'une instruction externe d'écriture en mémoire. Mais pour avoir un accès rapide à la mémoire flash il vaut mieux mettre ces bits à 000<sub>b</sub>.

## IV.8 LES INTERRUPTIONS :

Le MSC1201 Y3 met à la disposition du programmeur deux types d'interruption :

- Des interruptions internes qui se déclenchent par programmation (le bit qui définit l'événement déclencheur de l'interruption est activé par programmation).
- Des Interruptions externes qui se déclenchent par la détection d'un front ou un niveau logique sur une broche entrée/sortie (Port 1 ou Port 3).

Les interruptions doivent être déclarées dans le programme suivant la syntaxe suivante :

**Void Nom\_IT Interrupt Num\_IT using 1**

Où Nom\_IT : est le nom de l'interruption choisie par le programmeur.

Num\_IT : est le numéro d'interruption qui décrit l'adresse du vecteur d'interruption ; ce numéro est très important, il est calculé à partir de la relation :

**Adresse\_VI<sub>H</sub> = Num<sub>8</sub>\_TI \* 8 + 3.**

Les interruptions ont un ordre de priorité bien défini (Voir le *Tableau IV.16*). Les interruptions auxiliaires sont des interruptions système c'est donc les plus prioritaires alors que les interruptions externes sont généralement les moins prioritaires sauf exception.

Lors de l'exécution d'une interruption quelconque, une interruption plus prioritaire qui se déclenche est prise en compte; après son exécution, le microcontrôleur revient à l'interruption initiale. Par contre si une interruption moins prioritaire est déclenchée, elle est immédiatement ignorée et le microcontrôleur continue à exécuter l'interruption en cours.

INTERRUPT/EVENT	INTERRUPT		PRIORITY	FLAG	ENABLE	PRIORITY CONTROL
	ADDR	NUM				
AV <sub>DD</sub> Low Voltage Detect	33h	6	High 0	ALVDIP (AIPOL.1) <sup>(1)</sup>	EALV (AIE.1) <sup>(1)</sup>	N/A
Count (SPI/I <sup>2</sup> C)	33h	6	0	CNTIP (AIPOL.2) <sup>(1)</sup>	ECNT (AIE.2) <sup>(1)</sup>	N/A
I <sup>2</sup> C Start/Stop	33h	6	0	I2CIP (AIPOL.3) <sup>(1)</sup>	EI2C (AIE.3) <sup>(1)</sup>	N/A
Milliseconds Timer	33h	6	0	MSECIP (AIPOL.4) <sup>(1)</sup>	EMSEC (AIE.4) <sup>(1)</sup>	N/A
ADC	33h	6	0	ADCIP (AIPOL.5) <sup>(1)</sup>	EADC (AIE.5) <sup>(1)</sup>	N/A
Summation Register	33h	6	0	SUMIP (AIPOL.6) <sup>(1)</sup>	ESUM (AIE.6) <sup>(1)</sup>	N/A
Seconds Timer	33h	6	0	SECIP (AIPOL.7) <sup>(1)</sup>	ESEC (AIE.7) <sup>(1)</sup>	N/A
External Interrupt 0	03h	0	1	IE0 (TCON.1) <sup>(2)</sup>	EX0 (IE.0) <sup>(4)</sup>	PX0 (IP.0)
Timer 0 Overflow	0Bh	1	2	TF0 (TCON.5) <sup>(3)</sup>	ET0 (IE.1) <sup>(4)</sup>	PT0 (IP.1)
External Interrupt 1	13h	2	3	IE1 (TCON.3) <sup>(2)</sup>	EX1 (IE.2) <sup>(4)</sup>	PX1 (IP.2)
Timer 1 Overflow	1Bh	3	4	TF1 (TCON.7) <sup>(3)</sup>	ET1 (IE.3) <sup>(4)</sup>	PT1 (IP.3)
Serial Port 0	23h	4	5	RI_0 (SCON0.0) TI_0 (SCON0.1)	ES0 (IE.4) <sup>(4)</sup>	PS0 (IP.4)
External Interrupt 2	43h	8	6	IE2 (EXIF.4)	EX2 (EIE.0) <sup>(4)</sup>	PX2 (EIP.0)
External Interrupt 3	4Bh	9	7	IE3 (EXIF.5)	EX3 (EIE.1) <sup>(4)</sup>	PX3 (EIP.1)
External Interrupt 4	53h	10	8	IE4 (EXIF.6)	EX4 (EIE.2) <sup>(4)</sup>	PX4 (EIP.2)
External Interrupt 5	5Bh	11	9	IE5 (EXIF.7)	EX5 (EIE.3) <sup>(4)</sup>	PX5 (EIP.3)
Watchdog	63h	12	10 Low	WDTI (EICON.3)	EWDI (EIE.4) <sup>(4)</sup>	PWDI (EIP.4)

*Tableau IV.16 : Les Interruptions du MSC1201Y3.*

Pour qu'elle soit exécutée (prise en compte) une interruption doit être autorisée (activée); Ceci est possible en programmant des registres spéciaux tels que : **AIE** (Auxiliary Interrupt Enable) pour les interruptions les plus prioritaires; **IE** (Interrupt Enable) et **EIE** (Extended Interrupt Enable). Ces registres se présentent sous la forme suivante :



**Auxiliary Interrupt Enable (AIE)**

	7	6	5	4	3	2	1	0	Reset Value
SFR A6h	ESEC	ESUM	EADC	EMSEC	EI2C	ECNT	EALV	0	00h

Où **ESEC** : (Enable Second System Timer Interrupt). Cette interruption est la moins prioritaire parmi les interruptions auxiliaires ; elle est masquée quand ce bit est à état bas et autorisée lorsqu'il est à état haut.

**ESUM** : (Enable Summation Interrupt). Cette interruption est activée quand ce bit est égal à 1 et masquée quand il est égal à 0.

**EADC** : ( Enable ADC Interrupt). Cette Interruption Concerne l'ADC ; elle est activée pour EADC = 1 et masquée pour EADC = 0 .

**EMSEC** : ( Enable Millisecondes system Timer Interrupt). Cette interruption permet des signaux des timers de quelques millisecondes; Cette interruption est activée quand ce bit et à 1 et elle est masquée dans le cas contraire.

**EI2C** : ( Enable I<sup>2</sup>C Start / Stop Bit). Quand EI2C = 0 cette interruption est maquée dans le cas contraire elle est autorisée.

**ECNT** : ( Enable Serial Bit Count Interrupt). Comme pour les interruptions Précédentes, cette interruption est activée lorsque ce bit est à l'état haut ; elle est masquée dans le cas contraire.

**EALV** : ( Enable Analog Low Voltage Interrupt). Cette Interruption se déclenche pour une faible tension analogique, elle est activée pour EALV = 1 et masquée dans le cas contraire.

**Interrupt Enable (IE)**

	7	6	5	4	3	2	1	0	Reset Value
SFR A8h	EA	0	0	ES0	ET1	EX1	ET0	EX0	00h

Où **EA** : (Global Interrupt Enable). Ce bit peut masquer les toutes les interruptions sauf les interruptions auxiliaires. Les interruptions sont masquées quand il est à l'état Bas et activées quand il est à l'état Haut.

**ES0** : (Enable Serial Port 0 Interrupt). Ce bit permet de masquer l'interruption de l'USART quand il est à 0 et de l'autoriser quand il est à 1. Le déclenchement de l'interruption se fait par les flags **RI\_0** ou **TI\_0**.

**ET1** : (Enable Timer 1 Interrupt). Ce bit autorise l'interruption du timer quand il est à 1; celle-ci se déclenche en mettant le Flag **TRI** à 1.

**EX1** : ( Enable External Interrupt 1). Quand il est à 0 ce bit masque l'interruption externe 1; par contre quand il est à 1 il l'autorise. Cette dernière est déclenchée par un niveau bas ou un Front descendant sur la broche **INT1** .

**ET0** : (Enable Timer 0 Interrupt). Ce bit est pour le Timer 0 ce qu'est **ET1** pour le Timer 1.

**EX0** : ( Enable External 0 Interrupt). Idem que **EX1**.

**Extended Interrupt Enable (EIE)**

	7	6	5	4	3	2	1	0	Reset Value
SFR E8h	1	1	1	EWDI	EX5	EX4	EX3	EX2	E0h

Où **EWDI** : Ce bit active ( mis à 1) ou masque ( mis à 0) l'interruption du watchdog Timer.

**EX5** : Mis à 1 ce bit autorise l'interruption externe 5.

**EX4** : Idem que **EX5** pour l'interruption externe 4.

**EX3** : Idem que **EX5** pour l'interruption externe 3.

**EX2** : Idem que **EX5** pour l'interruption externe 2.

Quand une interruption est déclenchée, un Flag est mis à 1; l'interruption continuera à s'exécuter continuellement tant que ce flag est à 1 d'où la nécessité de le remettre à 0 à la fin du sous programme d'interruption.

On peut Aussi modifier l'ordre de priorité des interruptions ( sauf les Interruptions auxiliaires) en rendant quelques interruptions plus prioritaires que d'autres par la programmation des registre **IP** et **EIP** comme suit :

**Interrupt Priority (IP)**

	7	6	5	4	3	2	1	0	Reset Value
SFR B8h	1	0	0	PS0	PT1	PX1	PT0	PX0	80h

Où **PS0** : ( Serial Port 0 Interrupt). Quand il est à l'état bas, ce bit n'a aucune incidence sur l'ordre de priorité naturel. Par contre quand il est à 1 l'interruption correspondante devient prioritaire.

**PT1** : ( Timer 1 Interrupt ) Idem que **PS0** mais pour l'interruption du Timer 1.

**PX1** : ( External Interrupt 1 ) Idem que **PS0** mais pour l'interruption externe 1.

**PT0** : ( Timer 0 Interrupt ) Idem que **PS0** mais pour l'interruption du Timer 0.

**PX0** : ( External Interrupt 1 ) Idem que **PS0** mais pour l'interruption externe 1.

**Extended Interrupt Priority (EIP)**

	7	6	5	4	3	2	1	0	Reset Value
SFR F8h	1	1	1	PWDI	PX5	PX4	PX3	PX2	E0h

**PWDI** : ( Watchdog Timer Interrupt ) Idem que **PS0** mais pour l'interruption du watchdog Timer.

**PX5** : ( External Interrupt 5 ) Idem que **PS0** mais pour l'interruption externe 5.

**PX4** : ( External Interrupt 4 ) Idem que **PS0** mais pour l'interruption externe 4.

**PX3** : ( External Interrupt 3 ) Idem que **PS0** mais pour l'interruption externe 3.

**PX2** : ( External Interrupt 2 ) Idem que **PS0** mais pour l'interruption externe 2.

Il est conseillé de garder l'ordre naturel de priorité afin d'éviter les problèmes de masquage inattendu de certaines interruptions.

Après Cette brève description des éléments essentiels du MSC1201Y3 nous allons présenter la carte d'évaluation correspondante sur laquelle nous allons effectuer les tests en haute température.

# CHAPITRE V

## PRESENTATION DE LA CARTE D'EVALUATION

Les tests en haute température seront effectués sur des cartes d'évaluation fournies par les fabricants. Celle qui héberge le microcontrôleur MSC1201Y3 est la carte MSC1201EVM qu'on va présenter dans ce qui suit :

## **V.1 PRESENTATION DE LA CARTE D'EVALUATION MSC1201EVM :**

Toute communication PC hôte – matériel ne peut être réalisée sans adaptations matériel et protocolaire (logicielle). C'est pour cela que des descriptions matérielle et logicielle de la carte d'évaluation utilisée sont indispensables pour la détermination des tests à effectuer et de la meilleure manière de les réaliser. Ainsi il faut s'intéresser aux périphériques hardwares (ADC, DAC, Mémoires) et aux interfaces (SPI, I2C, UART, RS232) d'une part et à l'environnement de développement d'autre part.

### **V.1.1 Description matérielle :**

(L'Annexe A montre le schéma électrique de la carte MSC1201 EVM).

Cette carte intègre le microcontrôleur MSC1201 Y3 muni d'un convertisseur analogique numérique 24 bits de grande précision qui permet d'aller jusqu'à 22 bits effectifs. Ce convertisseur est considéré comme un périphérique privilégié à tel point que le fabricant présente la MSC1201EVM comme une carte à base d'un convertisseur 24 bits supporté par un microcontrôleur de noyau 8051.

Ainsi on trouve, sur cette carte, 10 entrées analogiques qui sont en liaison directe avec le convertisseur analogique numérique ; ainsi on trouve :

- Six entrées de mesure (AIN0-AIN5); ces dernières sont multiplexées donc il n'y a qu'une seule entrée utilisée à la fois.
- Une entrée GND qui est la masse de la carte.
- Une entrée AINCOM utilisée généralement comme la masse; elle peut reliée au GND grâce à un jumper si la mesure prend comme référence la masse de la carte. Cette entrée peut être utilisée comme entrée de mesure en cas de nécessité mais il est généralement conseillé de la mettre à la masse.
- Deux entrées de mesure non multiplexées (AIN+ et AIN-) qui permettent de faire des mesures en mode différentiel (mesurer la différence de potentiel entre deux entrées actives)

On trouvera aussi une sortie analogique IDAC qui peut être utilisée comme générateur de courant digital de faible intensité ( $I_{max} = 1mA$ ); il est généralement utilisé par le programmeur

comme réaction à un test pour activer ou désactiver un équipement adjacent.

En ce qui concerne les ports de communication, la carte a tout ce qu'il faut pour satisfaire le designer puisqu'elle propose pas moins de quatre ports séries différents:

- Un RS232, classique, qui communique directement avec l'USART (port 0) du microcontrôleur.
- Un port SPI/ I2C (port 1) qui permet d'utiliser soit un SPI trois fils soit un I2C deux fils (un port à la fois), le choix d'un port ou l'autre se fait grâce à une configuration bien spécifique de certains jumpers (JMP9, 10,13).
- Un port USB 2.0 qui est lié au microcontrôleur grâce à un adaptateur USART/USB appelé TUSB3410 de TI , Ce port est vraiment intéressant car c'est une version rapide et de faible consommation comparé au port RS232 par exemple, présent sur la carte.

La carte peut aussi être alimentée en deux modes :

- Un mode standard avec une tension de 5V à fournir.
- Un mode faible voltage avec une tension 3.3V à fournir.

Le choix de l'un ou l'autre des modes se fait grâce à un jumper (JMP2)

Bien sûr, comme toutes les cartes d'évaluation munies d'un ADC on a des points test pour mesurer la référence interne, des entrées pour référence externe, des switches de RESET et de HALT, des LEDs de contrôle, ainsi que tout ce qui est indispensable pour le fonctionnement normal de n'importe quelle carte d'évaluation.

La Figure V.1 représente un schéma de la carte précisant tous les composants présents.

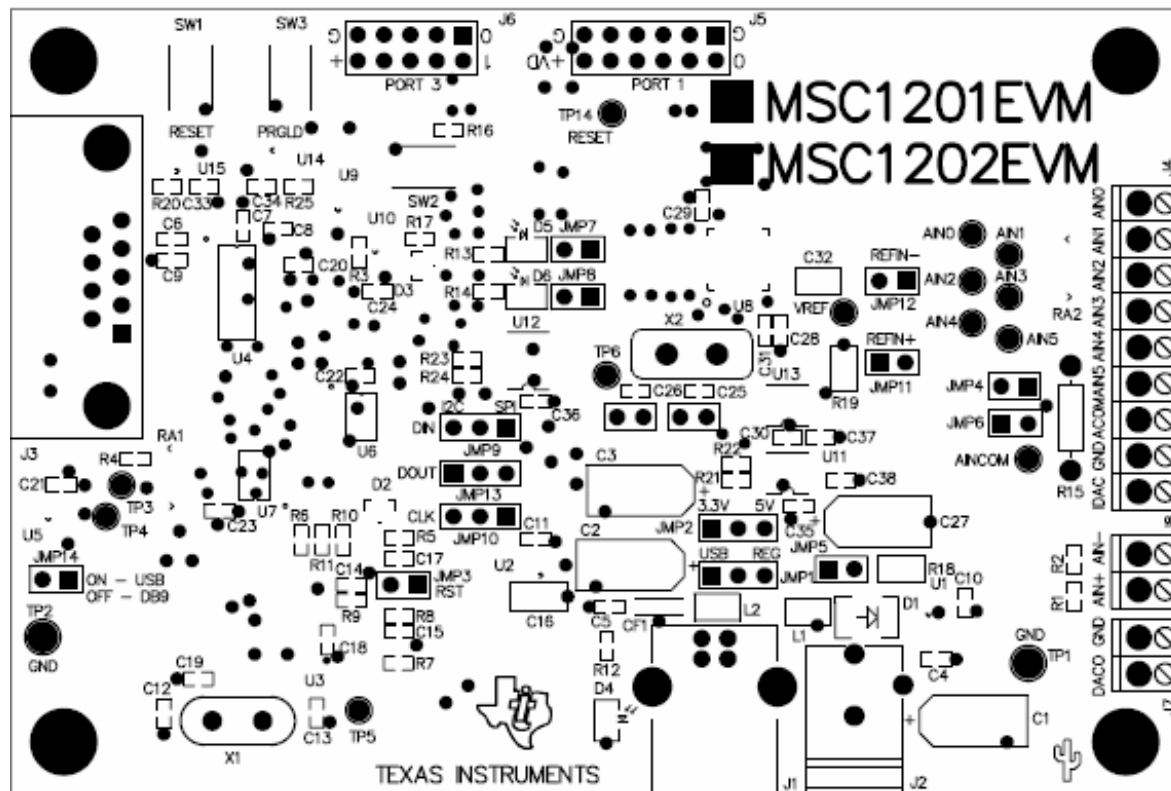


Figure V.1: Schéma de la carte d'évaluation MSC1201EVM.

### V.1.2 Description Logicielle :

Le microcontrôleur est programmé puis géré par deux logiciels différents respectivement:

#### a) Keil Uvision 3:

Ce logiciel permet d'établir un programme en langage C qui est présenté sous forme de projet. Tout est élaboré autour d'un fichier source **prog.c** auquel on ajoute des bibliothèques telles que : REG1200.h, stdio.h, rom1201.a51, etc....

L'environnement de Keil Uvision 3 est facilement maîtrisable, il ressemble à tous les environnements de programmation standard. La configuration du logiciel avant de commencer à programmer est très importante notamment le choix du composant à programmer car ce logiciel assume la programmation de tous les microcontrôleurs de la famille MSC12xx.

L'environnement de programmation se présente comme suit :

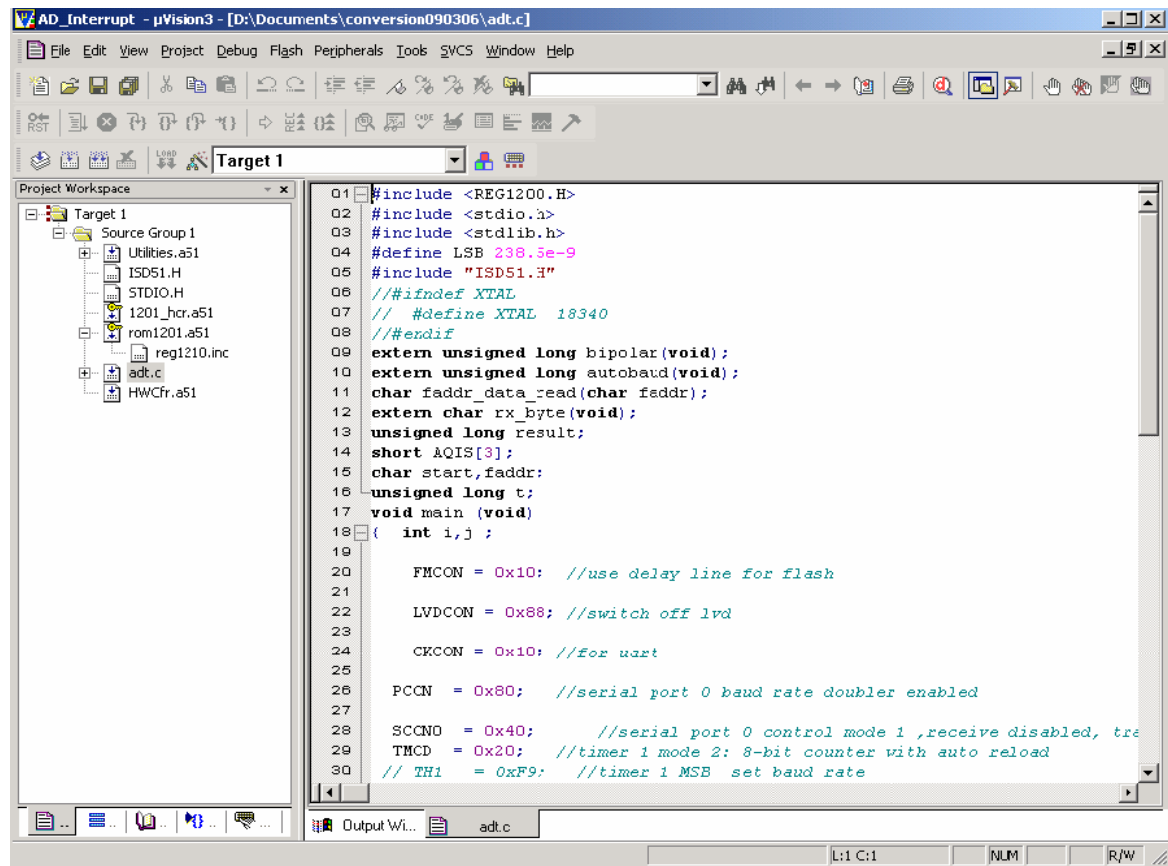


Figure V.2 : Environnement du Keil Uvision 3

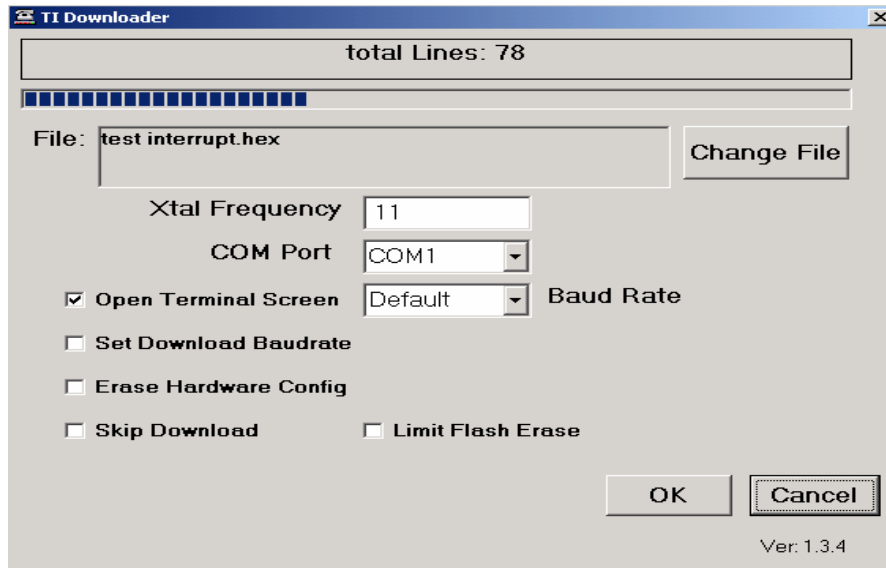
Après la compilation du programme (build project) un fichier **prog.hex** est créé : c'est le fichier en langage machine chargeable sur la mémoire programme (programme flash). Or Keil Uvision 3 n'assume pas le chargement ; c'est un autre logiciel qui le fait.

### **b) TI downloader:**

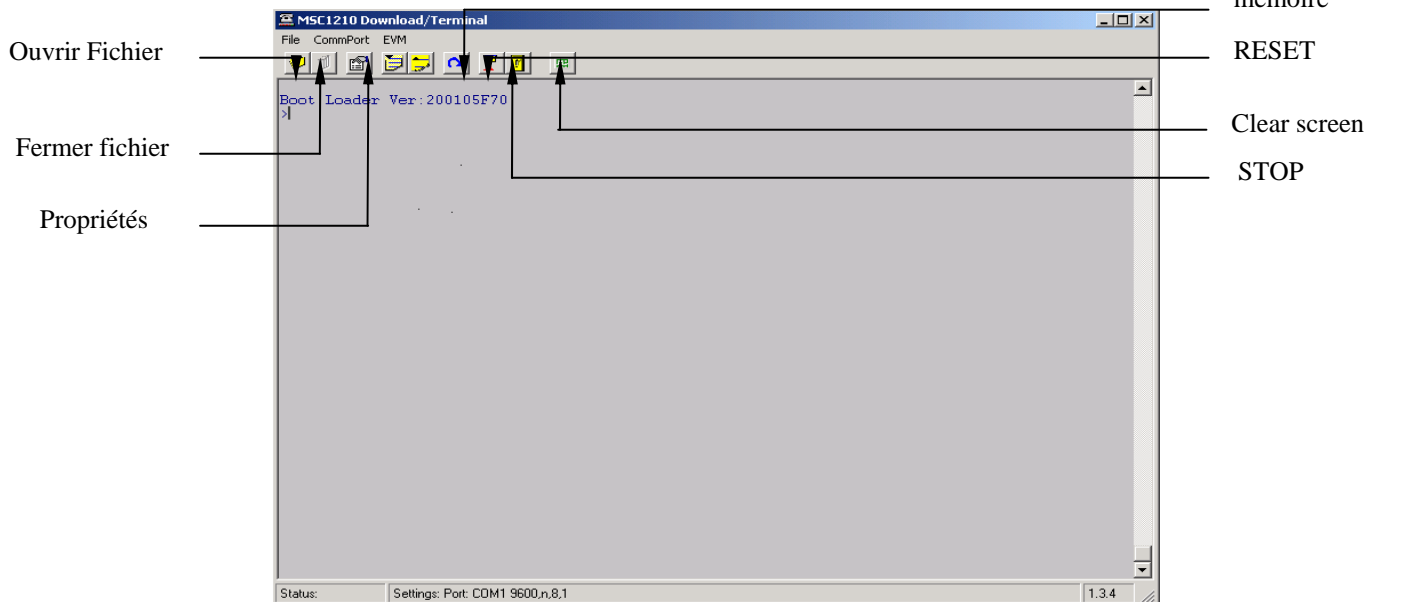
TI downloader est un logiciel proposé par Texas Instrument pour programmer la mémoire Flash (programme) de la famille des microcontrôleurs Msc12xx. Pour cela, un fichier de format **prog.hex** ou **prog.bin** est indispensable car ce logiciel n'assume que les programmes en code machine. Aussi TI downloader joue le rôle d'une interface de communication graphique entre l'utilisateur et le microcontrôleur une fois que le programme est chargé en mémoire car il met à la disposition de l'utilisateur des touches virtuelles de RESET, de STOP ainsi que l'affichage des données récupérées (par l'ADC par exemple) et la sauvegarde de ces dernières dans un fichier; Ce logiciel a été élaboré avec visual basic c'est pour cela qu'il offre une telle aisance d'utilisation mais au niveau performance ce logiciel reste limité vue les mesures qui devront être effectuées dans le but de l'évaluation du MSC1201; ainsi on peut envisager la possibilité d'utiliser ou même d'élaborer une autre interface graphique avec des outils plus performants qui permettent de dessiner par exemple les résultats obtenus ou faire différents calculs pour assurer un niveau de performance et une transparence satisfaisants pour l'utilisateur. Malgré cela l'utilisation de TI downloader reste indispensable pour pouvoir programmer le composant.

L'interface graphique TI downloader se présente sous deux formes :

- Une phase de chargement pendant laquelle l'utilisateur ne peut intervenir (seulement réglage de baud rate ou de l'espace mémoire flash à effacer...) (Voir *Figure V.3*)
- Une phase d'exécution qui se présente sous la forme d'un écran qui s'ouvre à l'utilisateur et lui donne la main pour lancer et arrêter l'exécution du programme, récupérer les données envoyées par le microcontrôleur, envoyer des données au microcontrôleur moyennant une réception prévue au préalable durant la programmation. (Voir *Figure VI.3*)



a) TI downloader durant le chargement



b) TI downloader après chargement (terminal de communication)

*Figure V.3 : Modes de fonctionnement de TI downloader.*



Ainsi la combinaison de ces deux logiciels permet de programmer et de communiquer avec la carte d'évaluation MSC1201 EVM, et pour plus de facilité d'utilisation Texas Instrument a ajouté la possibilité d'appeler le TI downloader directement à partir de Keil Uvision 3 moyennant une bonne configuration de ce dernier.

Après cette brève description de la carte d'évaluation MSC1201EVM, et l'acquisition de certaines connaissances de base sur le microcontrôleur qu'elle héberge, passons au cœur du projet qui est l'évaluation de ce microcontrôleur en haute température. Nous nous intéresserons à la méthode d'évaluation ainsi qu'aux résultats obtenus.

# CHAPITRE VI

## EVALUATION DU MSC1201Y3 EN HAUTE- TEMPERATURE

On s'intéressera dans cette partie à la qualification du MSC1201Y3 pour des températures bien supérieures à 125 °C spécifiée par le fabricant vu que dans un puit de pétrole les températures vont des fois jusqu'à 175 °C et plus, ce qui nous incitera à utiliser une méthode d'évaluation adéquate pour laquelle différents tests sont effectués.

## **VI.1 PROCÉDÉ D'EVALUATION :**

L'évaluation d'un microcontrôleur est principalement basée sur *les dégradations* de ces différentes *fonctions* à des *températures supérieures à celles spécifiées par le fabricant*. Cela veut dire qu'il faut tester le plus de fonctions possibles et aller vers les températures les plus élevées possibles.

### **VI.1.1 Problématique :**

On est confronté à deux problèmes :

- Le premier, étant la technique qui consiste en la façon de chauffer le composant sur des cartes d'évaluation vu qu'elles ne sont pas dédiées à ces opérations et risquent de fondre.
- Le deuxième, plus en relation avec la fiabilité des mesures, réside dans le nombre de mesures qu'il faut effectuer pour chaque fonction afin d'avoir des résultats acceptables. Vu que le microcontrôleur a plusieurs fonctions à tester et que les mesures deviennent vite exhaustives il faudra détecter à partir de quel moment la détérioration entraîne des résultats erronés; ce qui nous incitera à réaliser un plan de mesures pour protéger la carte d'évaluation.

### **VI.1.2 Solutions :**

Le premier problème est vite réglé, grâce au matériel et procédés mis à notre disposition par Schlumberger, puisque l'idée repose sur le chauffage du composant seul, à l'aide d'une machine permettant de régler la température de l'air qu'elle débite. Ce dernier sort à grande vitesse dans un tuyau dont l'embout sera appliqué sur le composant; Le reste de la carte sera protégé par une matière isolante à la chaleur et la température sera contrôlée grâce à un thermomètre ( thermocouple) dont le détecteur sera fixé sur le microcontrôleur. (Voir les *Figures VI.1* et *VI.2*).

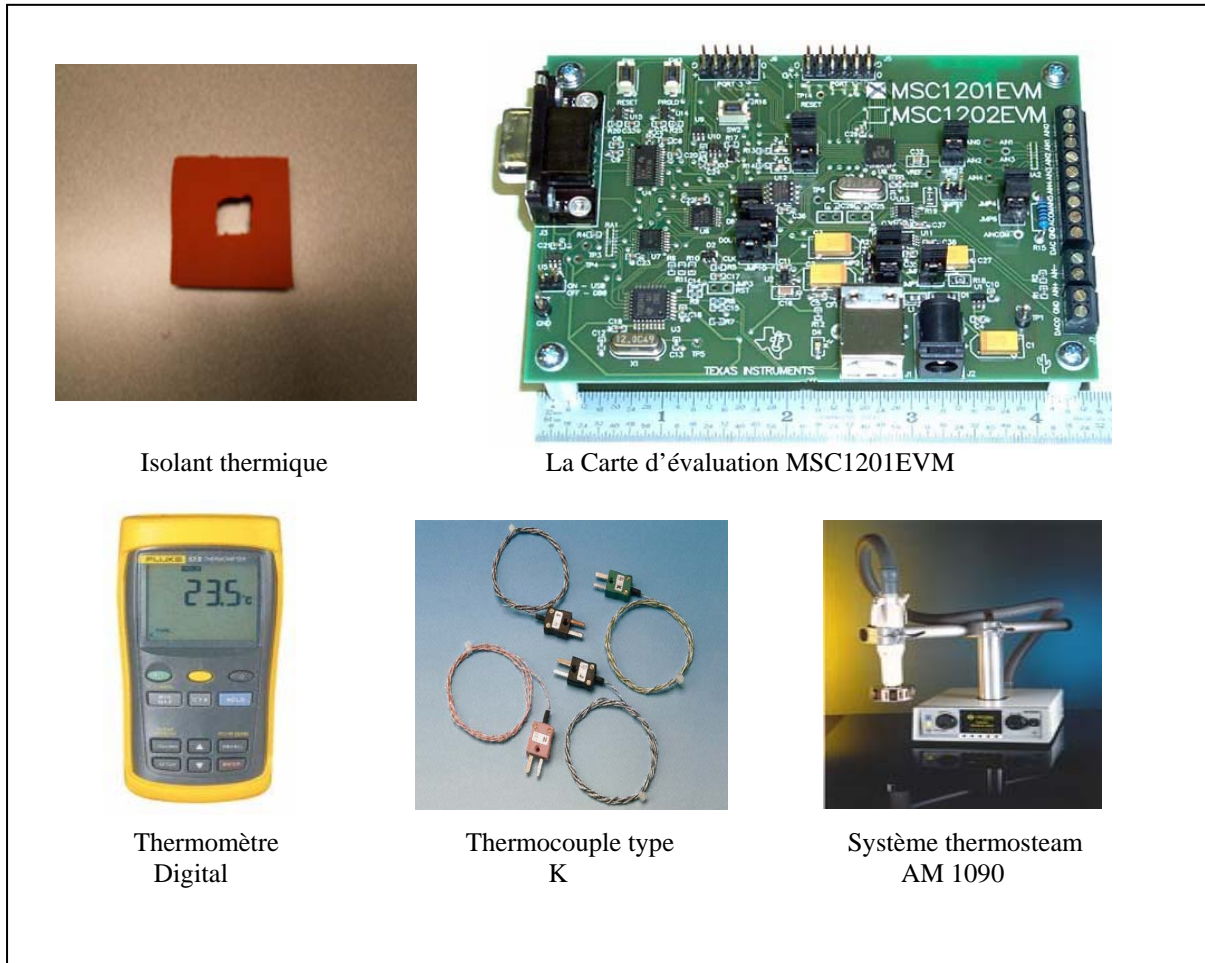


Figure VI.1 : Matériel utilisé pour les mesures en haute température.

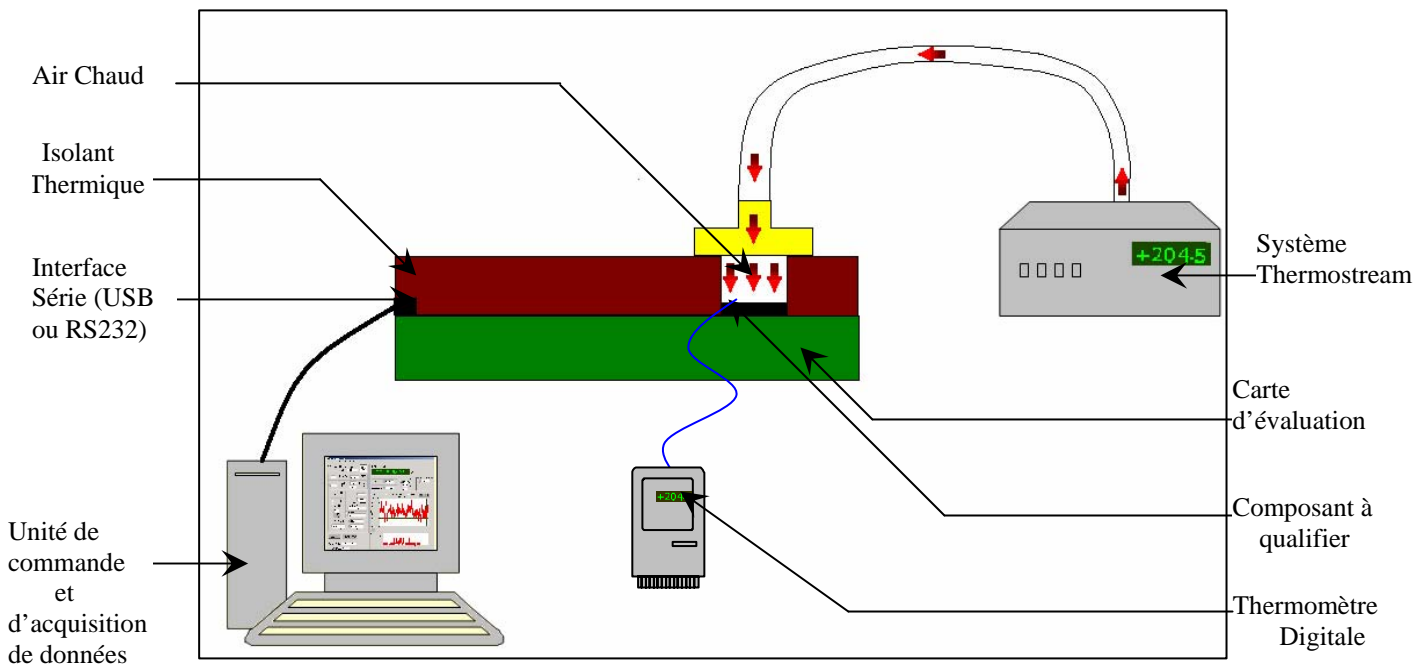


Figure VI.2 : Schéma de principe des mesures en haute température.

Le système thermostream (Thermotemp AM1090) est une machine qui permet de régler la température de l'air qu'elle fournit entre  $-80$  et  $225$  °C; Ainsi elle couvre totalement l'intervalle de température de nos tests mais cela requiert une bonne isolation (minimiser les fuites d'air chaud) afin de confiner la chaleur autour du microcontrôleur.

Le thermocouple de type K est sensible à l'intervalle de température  $[-80, 1325]$  °C. Il a été choisi pour son avantage économique puisque la mesure ne nécessite pas une grande précision de la température; de plus avec ce thermocouple on peut aller jusqu'à un chiffre après la virgule (un dixième de degré Celsius).

Le thermomètre digital (Digi-sense 8095-10) est adaptable avec plusieurs types de thermocouples (J, K, T, etc.) donc ses performances dépendent du thermocouple utilisé pour effectuer les mesures.

Le deuxième problème a été abordé en prenant une marge de sécurité puisqu'on a commandé d'emblée trois cartes d'évaluation afin d'éliminer tous les doutes concernant la fiabilité des résultats et d'éviter les pertes de temps qui peuvent survenir à l'attente d'une nouvelle carte d'évaluation en cas d'une détérioration fatale du microcontrôleur.

Mais cela reste insuffisant puisque les mesures à effectuer sont nombreuses et peuvent prendre un certain temps donc il faut être assez méthodique en privilégiant les mesures visant à qualifier les fonctions principales du microcontrôleur et qui intéressent les applications de l'entreprise telles que l'ADC en priorité, la source de courant digitale IDAC etc. et en respectant des poses régulières en effectuant des mesures à plus de  $125$ °C pour permettre le refroidissement de la carte.

## **VI.2 ÉVALUATION DU CONVERTISSEUR ANALOGIQUE NUMERIQUE :**

Comme l'ADC est le périphérique le plus important du MSC1201Y3, il est normal que le nombre de tests nécessaires pour l'évaluer soit important (de toute façon la qualification d'un ADC en haute température est généralement exhaustive). En plus, c'est un outil indispensable pour les applications de Schlumberger puisque toutes les mesures analogiques telles que les mesures de porosité, densité et résistivité d'une structure (casing) d'un puits de pétrole engagent une conversion analogique numérique.

### **VI.2.1 Méthode d'évaluation :**

Pour évaluer l'ADC, on effectuera plusieurs séries de mesures balayant plusieurs configurations du convertisseur en variant le gain, la fréquence, la tension de référence, l'état du buffer etc. Le but est de recueillir à chaque fois 1024 échantillons puis tracer des histogrammes qui décrivent la fréquence de répétition de chaque échantillon afin de calculer l'écart type puis le nombre de bits effectifs du convertisseur à chaque mesure. Ce nombre de bits effectifs ENOB sera notre jauge puisqu'on évaluera les performances de l'ADC en se basant sur le degré de diminution de ENOB dans l'intervalle de température  $[25, 185]$  °C.

Un type de mesures sera aussi effectué visant à observer le comportement de la tension de référence interne en haute température car ses variations influent sur les résultats de la conversion.

### VI.2.2 Mesures à effectuer :

On effectuera huit mesures en essayant de couvrir l'intervalle de conversion [0, VREF] pour 24 configurations différentes (Voir *Tableau VI.1*); le choix des valeurs est conditionné par la valeur du gain puisqu'on testera trois PGA différents 1, 64 et 128 (Voir *Tableau VI.2*). On s'intéressera aussi à la tension de référence car on comparera les résultats obtenus en référence interne et externe à VREF = 2.5 V. On changera aussi l'état du buffer et la fréquence d'acquisition (deux valeurs 1.28 et 101 Hz).

VREF PGA	Référence <b>Interne</b> VREF = 2.5V	Référence <b>Externe</b> VREF = 2.5V
<b>PGA = 1</b>	Buffer off et F <sub>DATA</sub> =1.28 Hz	Buffer off et F <sub>DATA</sub> =1.28 Hz
	Buffer <b>on</b> et F <sub>DATA</sub> =1.28 Hz	Buffer <b>on</b> et F <sub>DATA</sub> =1.28 Hz
	Buffer off et F <sub>DATA</sub> = <b>101</b> Hz	Buffer off et F <sub>DATA</sub> = <b>101</b> Hz
	Buffer <b>on</b> et F <sub>DATA</sub> = <b>101</b> Hz	Buffer <b>on</b> et F <sub>DATA</sub> = <b>101</b> Hz
<b>PGA = 64</b>	Buffer off et F <sub>DATA</sub> =1.28 Hz	Buffer off et F <sub>DATA</sub> =1.28 Hz
	Buffer <b>on</b> et F <sub>DATA</sub> =1.28 Hz	Buffer <b>on</b> et F <sub>DATA</sub> =1.28 Hz
	Buffer off et F <sub>DATA</sub> = <b>101</b> Hz	Buffer off et F <sub>DATA</sub> = <b>101</b> Hz
	Buffer <b>on</b> et F <sub>DATA</sub> = <b>101</b> Hz	Buffer <b>on</b> et F <sub>DATA</sub> = <b>101</b> Hz
<b>PGA = 128</b>	Buffer off et F <sub>DATA</sub> =1.28 Hz	Buffer off et F <sub>DATA</sub> =1.28 Hz
	Buffer <b>on</b> et F <sub>DATA</sub> =1.28 Hz	Buffer <b>on</b> et F <sub>DATA</sub> =1.28 Hz
	Buffer off et F <sub>DATA</sub> = <b>101</b> Hz	Buffer off et F <sub>DATA</sub> = <b>101</b> Hz
	Buffer <b>on</b> et F <sub>DATA</sub> = <b>101</b> Hz	Buffer <b>on</b> et F <sub>DATA</sub> = <b>101</b> Hz

*Tableau VI.1 : Différentes configurations de l'ADC.*

Valeurs à convertir PGA	PGA = 1 (Full scale = 2.5 V)	PGA = 64 (Full scale = 0.039 V)	PGA = 128 (Full scale = 0.0195 V)
1 <sup>ère</sup> valeur	0.00 V	0.00 V	0.00 V
2 <sup>ème</sup> valeur	0.032 V	0.005 V	0.00025 V
3 <sup>ème</sup> valeur	0.60 V	0.01 V	0.05 V
4 <sup>ème</sup> valeur	1.25 V	0.02 V	0.01 V
5 <sup>ème</sup> valeur	1.50 V	0.025 V	0.0125 V
6 <sup>ème</sup> valeur	2.00 V	0.03 V	0.015 V
7 <sup>ème</sup> valeur	2.30 V	0.035 V	0.0175 V
8 <sup>ème</sup> valeur	2.499 V	0.039 V	0.0195 V

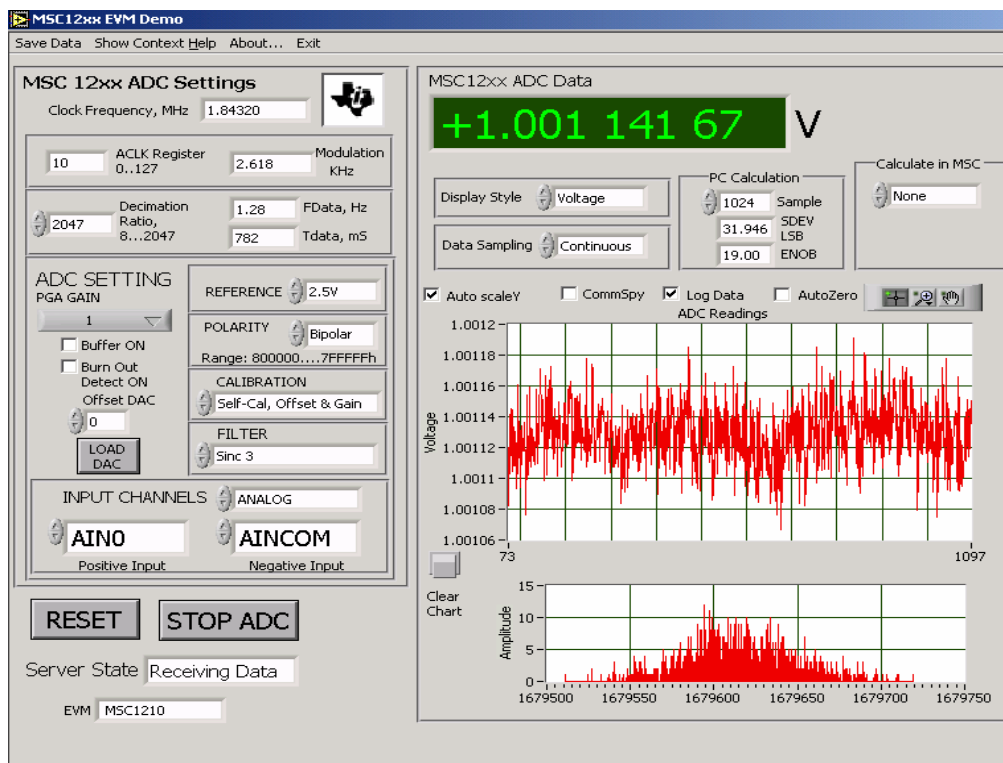
*Tableau VI.2 : Valeurs à convertir pour chaque PGA.*

Si on comptabilise bien, on devrait effectuer près de 200 acquisitions pour une seule température et puisque notre but est de monter en température, il faudra multiplier ce nombre par cinq vu qu'on doit faire les mesures pour les températures 25, 125, 150, 175 et 185 °C ; ainsi on aura près de mille acquisitions de 1024 points à faire et par conséquent plus de mille histogrammes à tracer et d'ENOB à calculer.

### VI.2.3 Travail préliminaire :

Le travail préliminaire, dans notre cas, est de programmer le microcontrôleur. En ce qui nous concerne c'est une étape décisive car la réussite des tests dépend étroitement de la bonne programmation du composant . Vu que les mesures à effectuer sont nombreuses, on a décidé de réaliser une interface graphique qui permette de régler les paramètres désirés et de récupérer les résultats de la conversion dans un fichier de manière totalement transparente pour l'utilisateur. Pour réaliser cette interface il faut écrire deux programmes; le premier, côté microcontrôleur, permet de gérer les différentes configurations et la conversion en respectant le protocole de communication avec l'ordinateur; le deuxième, côté ordinateur, permet de piloter le microcontrôleur par une interface graphique via un protocole de communication qu'on doit établir .

Par chance un programme d'interface graphique sur Labview<sup>(1)</sup> destiné à piloter l'ADC du microcontrôleur MSC1210 a été donné comme exemple sur le CD d'installation fourni avec la carte d'évaluation, il fallait juste l'adapter côté microcontrôleur. C'est ce qu'on a réussi à faire. L'interface graphique est représentée par la *Figure VI.3* ci-dessous :

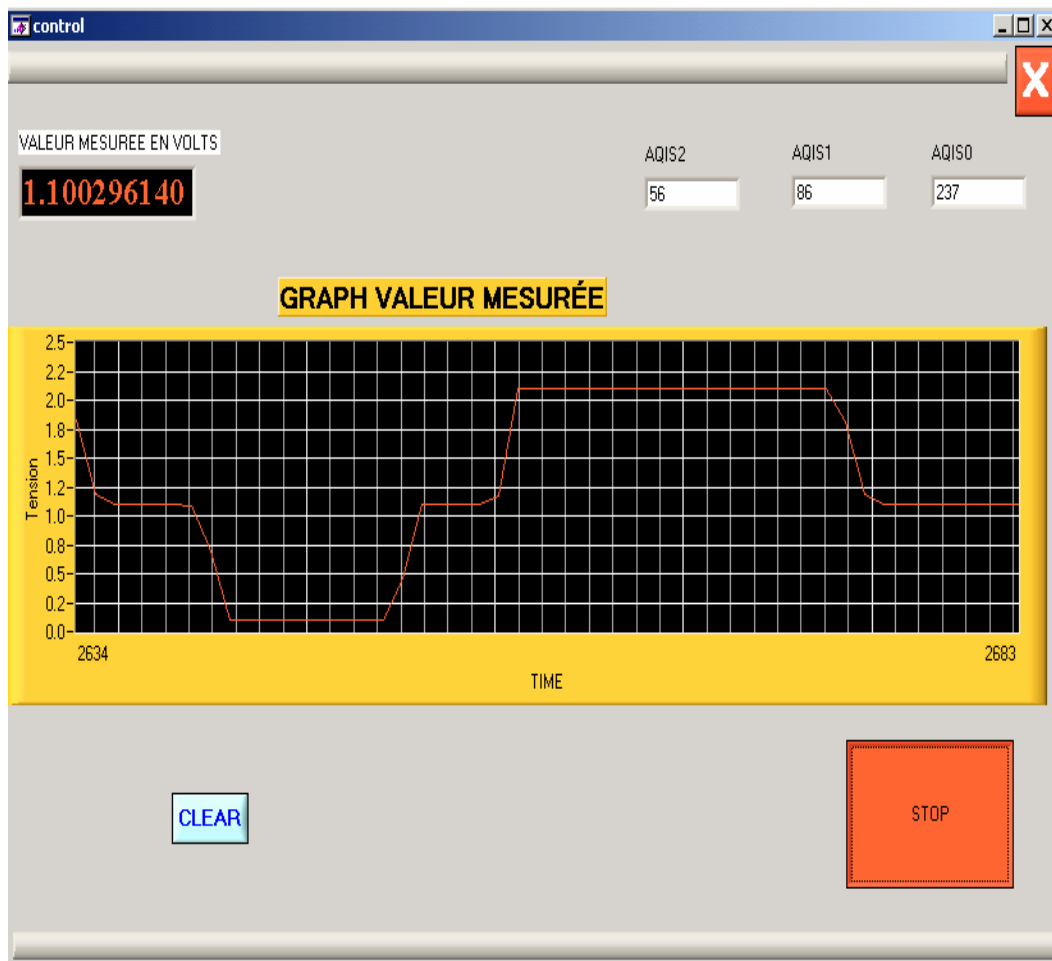


*Figure VI.3: Interface graphique (Version Démo)*

(1) : Labview est un logiciel et un langage de programmation proposé par National Instruments. Sa fonction Principale est de créer des interfaces graphiques et de contrôler différents équipements à partir d'un PC. Pour plus d'informations consulter la Référence [26].

On remarque que tous les réglages des paramètres du convertisseur peuvent être effectués directement sur l'interface graphique. En plus, on a deux graphes le premier (celui du haut *Figure VI.3*) trace les échantillons et le second (celui du bas) trace l'histogramme relatif aux échantillons obtenus. Le programme de l'interface graphique est complexe et long, il faut dire qu'il a été réalisé par des professionnels, c'est pour cette raison qu'on évitera de le présenter. Malheureusement, cette interface posait quelques problèmes dès qu'on essayait de faire une acquisition en référence externe puisqu'elle affichait tout le temps zéro. Ainsi l'obligation de réaliser une autre interface graphique pour les mesures en référence externe s'imposait. Pour ce, on a choisi Labwindows CVI<sup>(1)</sup> comme logiciel pour la réaliser car il permet de programmer en langage C.

Au niveau du protocole, on a considéré le cas le plus simple qui consiste à envoyer le caractère 's' pour START donc début de conversion et le caractère 'a' pour ARRET donc fin de conversion; la récupération des résultats se fait par le buffer de l'USART (relié à une RS232 sur la carte). La nouvelle interface graphique est moins performante mais elle permet de récupérer les résultats en référence externe. La *Figure VI.4* représente la nouvelle interface graphique.



*Figure VI.4: Nouvelle interface graphique.*

(1) : Labwindows CVI est un logiciel proposé par National instruments, il a la même fonction que Labview sauf qu'il est programmé en langage C. (Voir Référence [27]).



Exposons maintenant les infogrammes qui reflètent le programme établi pour réaliser cette interface graphique (L'Annexe B expose le programme)

a) *Côté microcontrôleur:*

Les tâches principales de ce côté sont la conversion analogique numérique puis le renvoi du résultat vers l'interface série (ordinateur) ; tout cela via le protocole de communication entre le microcontrôleur et l'ordinateur qu'il faudra aussi programmer.

L'infogramme de ce côté se présente comme suit :

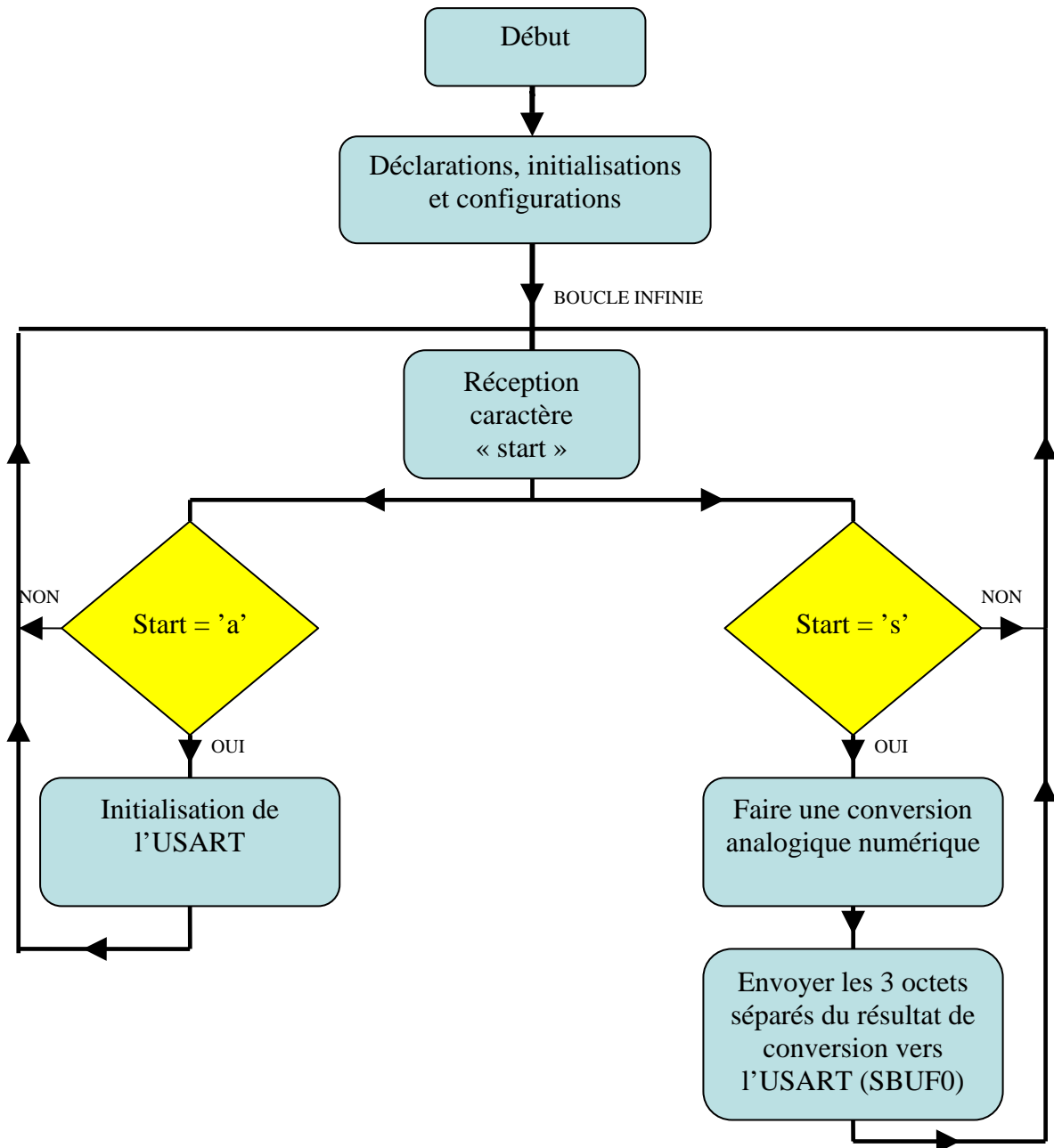


Figure VI.5: Infogramme représentant le programme côté microcontrôleur.

La partie initialisation et configuration permet de régler la fréquence de l'horloge, la fréquence d'acquisition, le gain d'amplification du convertisseur et la synchronisation du port de communication carte d'évaluation – ordinateur.

Le programme comporte une boucle infinie dans laquelle on teste continuellement un caractère reçu (instauration du protocole) ; le résultat permet de gérer la conversion d'un signal d'entrée analogique.

Dans le cas où le convertisseur est en service, le résultat de conversion est envoyé vers le buffer du port série sous forme de trois octets séparés représentant l'octet de poids faible, de poids moyen et de poids fort respectivement.

**b) Côté ordinateur:**

Le programme à écrire doit permettre le contrôle de la carte d'évaluation via le protocole de communication préalablement programmé sur le microcontrôleur, il doit gérer aussi la récupération des résultats de conversion et l'affichage numérique et graphique sur l'interface graphique. La *Figure VI.6* représente l'infogramme relatif au programme de contrôle coté ordinateur.

On remarque que le programme consiste en l'ouverture et la fermeture du port série d'où la gestion de la communication avec le microcontrôleur, aussi il consiste à récupérer les données envoyées par le microcontrôleurs, à les traiter pour avoir un résultat sur 24 bits qui est traduit en volt puis affiché sur un graphe et un afficheur numérique.

L'affichage des octets reçus séparément permet de vérifier s'ils ont été envoyés et reçus dans le bon ordre.

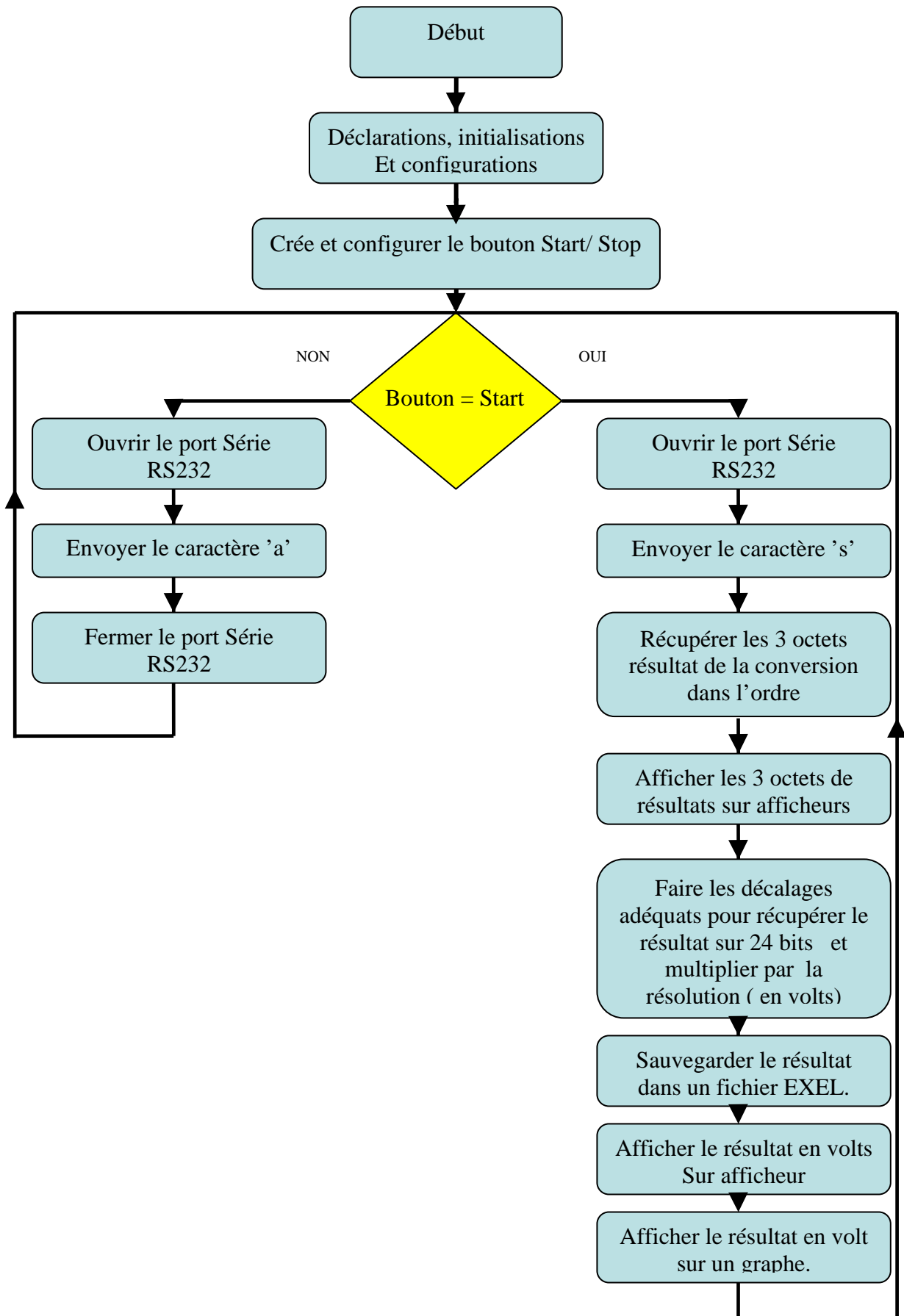


Figure VI.6: Infogramme représentant le programme côté microcontrôleur

Ainsi, on utilisera la première interface graphique (démo) pour les mesures en référence interne et la deuxième pour les mesures en référence externe.

#### VI.2.4 Résultats des mesures effectuées:

Pour chaque valeur mesurée on va créer un fichier Excel contenant les 1024 échantillons, le calcul de l'ENOB et l'histogramme correspondant.

Pour avoir un ordre de grandeur pour chaque configuration, comparons à présent les ENOB (*Tableau VI.3*) et les histogrammes (*Figure VI.7*) d'une même valeur (par exemple 2.0V) à 25 °C pour les configurations PGA1 , VREF interne = 2.5V, buffer off et  $F_{Data}=1.28$  Hz ou buffer on et  $F_{Data}=1.28$  Hz ou buffer off et  $F_{Data}= 101$ Hz ou buffer on et  $F_{Data}= 101$ Hz .Cela a pour but d'avoir une référence de comparaison à 25 °C sur laquelle on se basera pour évaluer les résultats obtenus en haute température (150, 175 et 185 °C).

Configuration	ENOB sur Interface Graph	ENOB calculé	ENOB Théorique
PGA=1, VREF = 2.5V interne			
Buffer off et $F_{Data} = 1.28$ Hz	18.73	18.53	21.6
Buffer on et $F_{Data} = 1.28$ Hz	19.43	19.20	21.5
Buffer off et $F_{Data} = 101$ Hz	12.1	11.87	12.5
Buffer on et $F_{Data} = 101$ Hz	12.09	11.895	12.5

*Tableau VI.3 : ENOB pour des mesures de la valeur 2.0 V à température ambiante (25 °C).*

Le tableau précédent représente les variations du nombre de bits effectifs en fonction de l'état du buffer et de la fréquence. Dans la première colonne du tableau l'ENOB est directement relevé sur l'interface graphique de démonstration alors que dans la deuxième l'ENOB à été calculé à partir de l'écart type des échantillons recueillis par la relation suivante:

$$ENOB = 24 - (20 \log \delta - 1.76) / 6.02 \quad \text{Où } \delta \text{ est l'écart type.}$$

Cette relation est valable quand la loi de distribution des échantillons est la loi normale (gaussienne); c'est pour cela que les histogrammes sont importants, ils permettent de confirmer si l'approximation au premier ordre dans le calcul de l'ENOB est valable. En observant la Figure on remarque que pour la fréquence 1.28 Hz l'approximation est totalement justifiée puisque on voit bien l'allure de la gaussienne, par contre elle l'est moins pour la fréquence 101Hz. En ce qui concerne la valeur théorique de l'ENOB elle a été relevée de la datasheet (*Figure VI.8*); on remarque un certain décalage entre la valeur théorique et la valeur calculée de l'ENOB, ce qui permet d'avoir une idée sur l'influence de bruit dans des conditions normales (température ambiante), cela appuie l'idée d'éviter de prendre la valeur théorique comme référence pour ne considérer que l'influence de l'augmentation de température. On peut tirer une deuxième conclusion : on perd la moitié de la précision quand on monte en fréquence, cela est prévu par la datasheet mais ce convertisseur n'est vraisemblablement pas fait pour des acquisitions rapides.

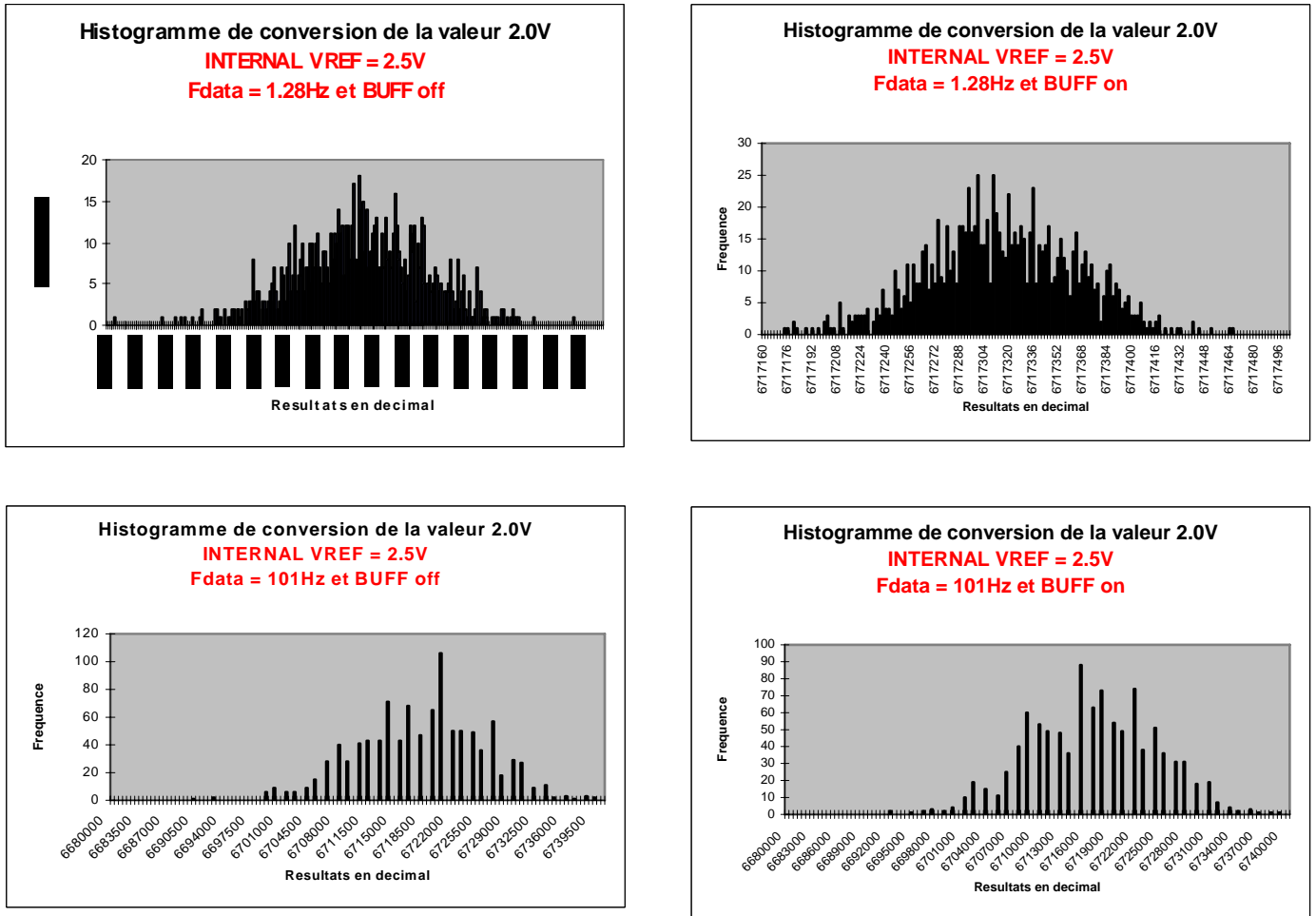


Figure VI.7 : Histogrammes relatifs à différentes mesures de la valeur 2.0 V à température ambiante.

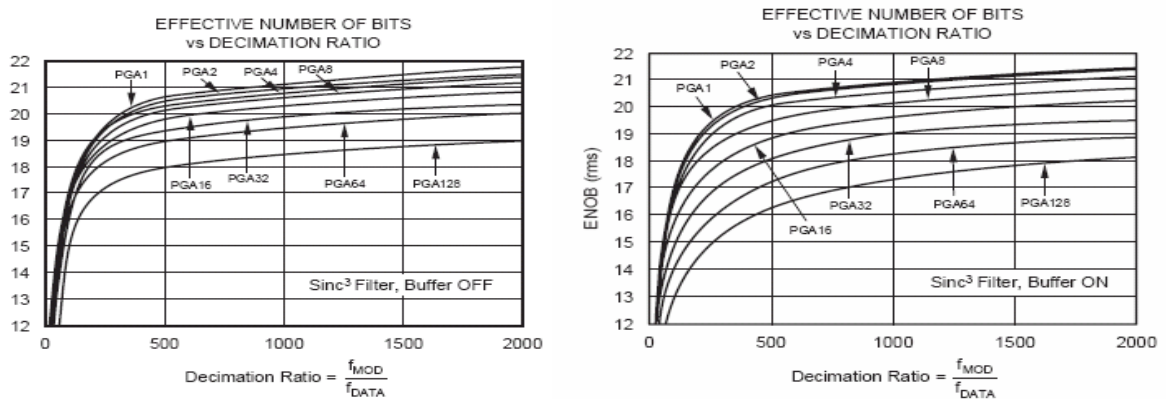


Figure VI.8: Valeur Théorique de l'ENOB (datasheet)

Après avoir effectué toutes les mesures, tracé tous les histogrammes (ce qui nous a pris environ 3 mois) une synthèse de tous les résultats a été établie permettant d'aboutir à une conclusion sur la validité d'utilisation du convertisseur analogique numérique en haute température, c'est ce que nous présentons dans l'Annexe C, sur les tableau de mesures d'ENOB.

### VI.2.5 Analyse des résultats:

Une première observation importante est : la précision diminue ( diminution de l'ENOB) en augmentant le gain d'amplification (PGA) indépendamment de la température; on a déclaré précédemment que l'augmentation du PGA entraîne une augmentation de la résolution, et bien cela se faisant au dépend de la précision de la conversion, comme le montre les résultats.

Le deuxième point qui attire l'attention, c'est les très bons résultats qu'on obtient en utilisant une source de tension stabilisée comme référence de l'ADC puisque on peut atteindre des ENOB allant jusqu'à 21.40 bits (une valeur proche des données du constructeur).

On remarque aussi que le buffer améliore la précision à fréquence réduite et son fonctionnement n'est pas altéré par l'augmentation de température; on gagne généralement un bit effectif en l'activant. Cela dit, à fréquence plus élevée son influence sur la mesure n'est plus perceptible et il vaut mieux le désactiver dans ce cas car il introduit une consommation supplémentaire.

De façon générale la technologie de l'ADC fait que sa précision *n'est pas vraiment altérée* par la **haute température** puisque dans le cas le plus extrême (185°C) **pour la même configuration**, la **perte de précision** se mesure en **demi bit effectif**.

On a remarqué un autre phénomène en effectuant les mesures en référence interne; il s'agit de la diminution de la valeur moyenne des échantillons quand la température augmente comme le montre le **Tableau VI.4** relevant les valeurs moyennes, en code et en volt, des mesures de la valeur 2.0 V (PGA = 1 et VREF=2.5 V interne).

Config. Temp.	Buffer off et F <sub>Data</sub> =1.28 Hz		Buffer <b>on</b> et F <sub>Data</sub> =1.28 Hz		Buffer off et F <sub>Data</sub> = <b>101</b> Hz		Buffer on et F <sub>Data</sub> = <b>101</b> Hz	
	Moyenne (DEC)	Moyenne (V)	Moyenne (DEC)	Moyenne (V)	Moyenne (DEC)	Moyenne (V)	Moyenne (DEC)	Moyenne (V)
25 C	6718108.354	2.00199	6717314.978	2.00176	6717867.752	2.00192	6716691.698	2.00157
125 C	6706645.584	1.99858	6705862.143	1.99835	6707212.915	1.99875	6707514.671	1.99884
150 C	6705723.902	1.99831	6704933.369	1.99807	6706581.109	1.99856	6707656.543	1.99888
175 C	6697452.646	1.99584	6696785.467	1.99564	6697727.702	1.99592	6699361.947	1.99641
185 C	6692089.212	1.99424	6691571.397	1.99409	6694438.96	1.99494	6692934.927	1.99449

Diminution

**Tableau VI.4** : Effet de la température sur la valeur moyenne des mesures en référence interne.

On remarque que la diminution de la valeur moyenne, quand la température augmente, est généralisée à toutes les configurations. Donc l'hypothèse qu'elle soit due à un dysfonctionnement du buffer ou autre est totalement écartée. Elle pourrait être expliquée par la variation de la référence interne en fonction de la température c'est pour cela qu'on a décidé de mesurer la référence interne en haute température.

### VI.2.6 Influence de la température sur la référence interne de l'ADC:

Les mesures effectuées sont assez simples, il faut mesurer avec un voltmètre la référence interne dans les deux modes  $V_{REF} = 2.5 \text{ V}$  et  $V_{REF} = 1.25 \text{ V}$  grâce au point test sur la carte d'évaluation puis observer le signal sur un oscilloscope pour vérifier sa stabilité (pas de fluctuation de la référence interne). L'opération a été répétée pour différentes valeurs de la température  $\{25, 85, 125, 150, 175, 185\} \text{ } ^\circ\text{C}$ .

Les résultats obtenus sont comme suit :

#### Résultats obtenus sur voltmètres

Temperature ( $^\circ\text{C}$ )	VREF Interne 2.5 (V)	VREF Interne 1.25 (V)
25	2.511	1.252
85	2.527	1.26
125	2.534	1.263
150	2.538	1.265
175	2.543	1.268
185	2.546	1.27

Tableau VI.5 : Influence de la température sur la référence interne

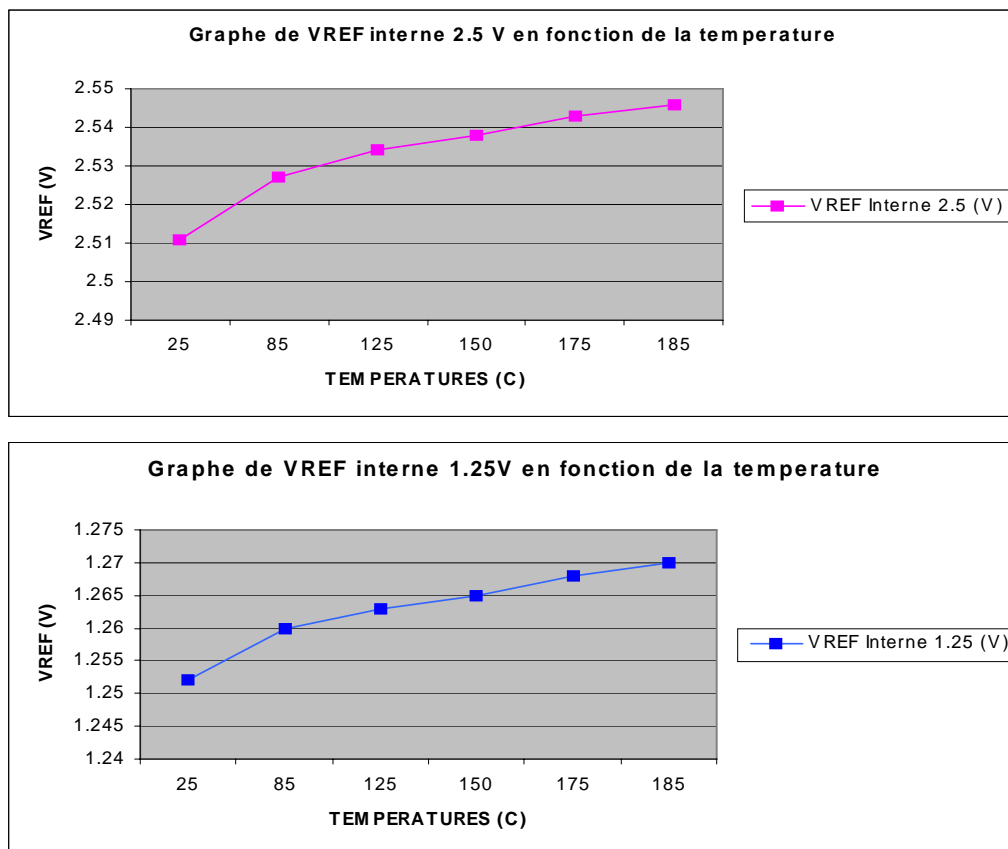


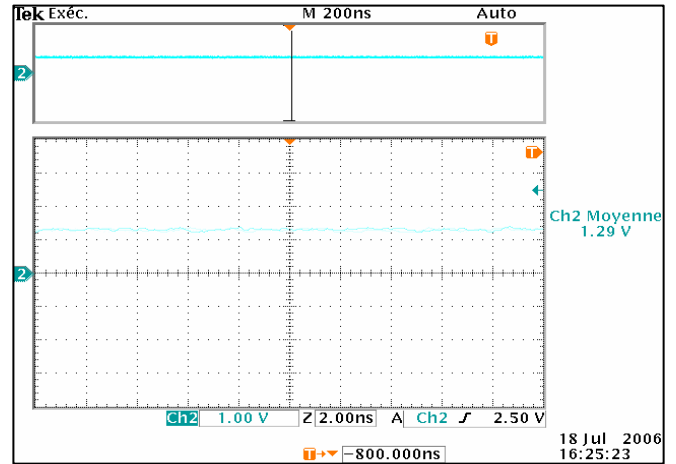
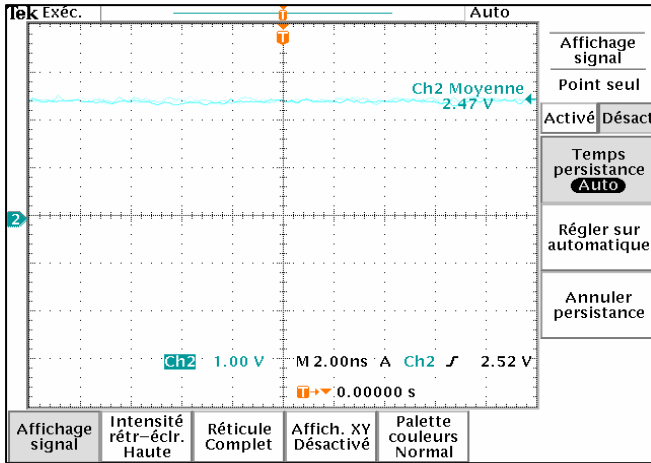
Figure VI.9: Graphiques représentant les variations de la référence interne en fonction de la température.

Vérification de la stabilité de référence interne par oscilloscope

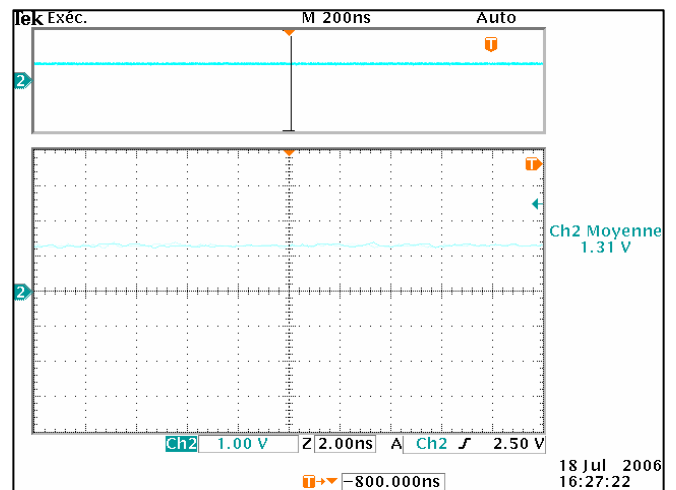
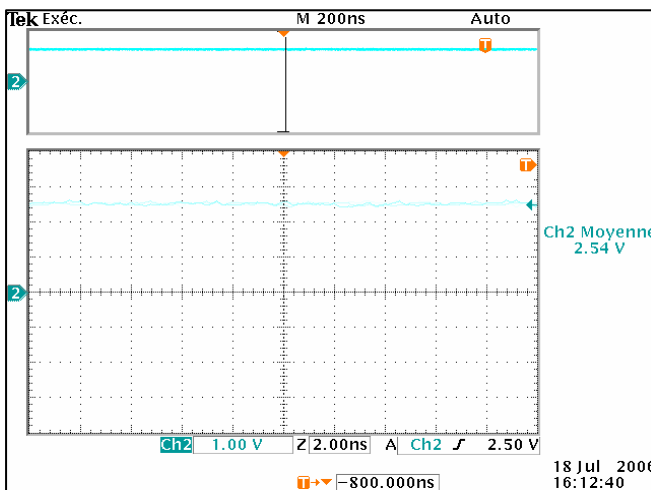
VREF = 2.5 V

VREF = 1.25 V

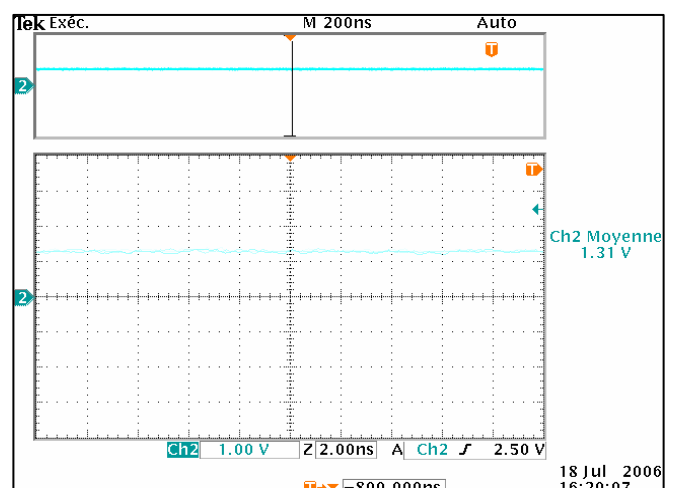
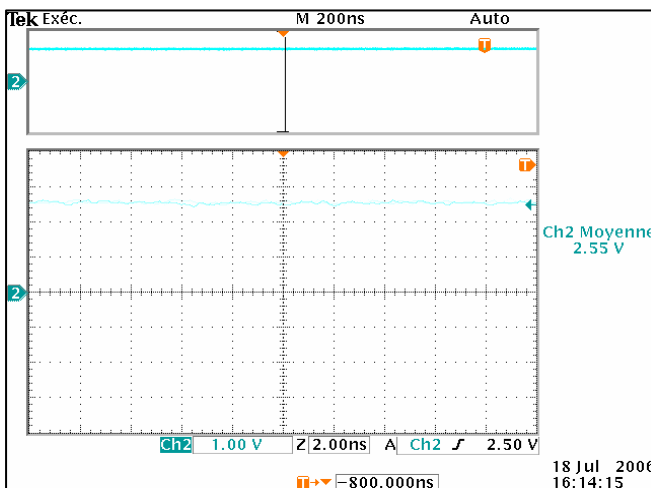
À 25 °C



À 125 °C



À 150 °C





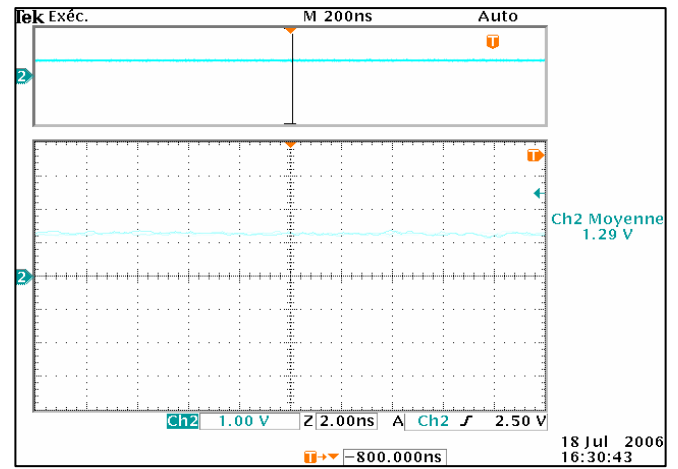
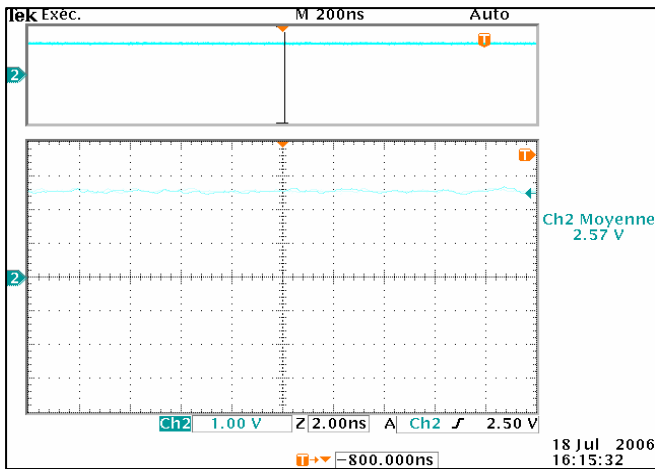
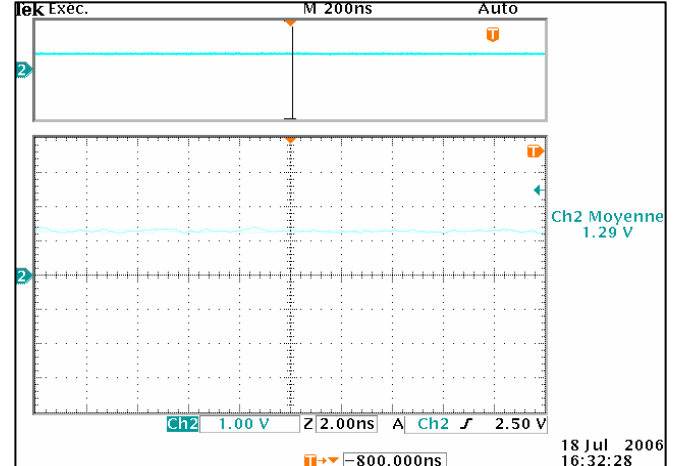
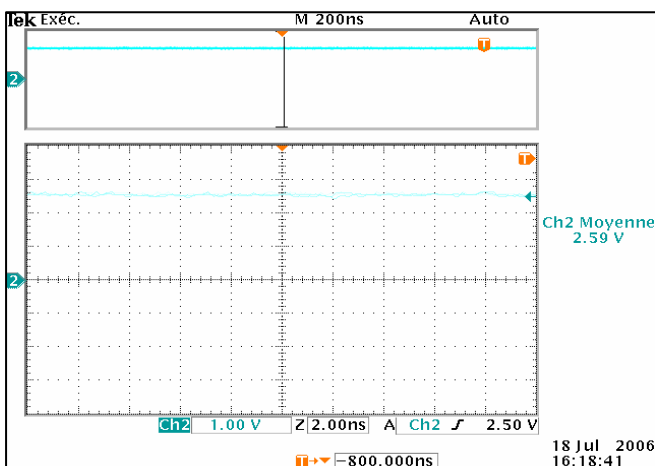
VREF= 2.5 VVREF= 1.25 VÀ 175 °CÀ 185 °C

Figure VI.10 : Visualisation de la référence interne sur l'oscilloscope à différentes températures.

### Remarques et Analyse :

- 1) Comme on le prévoyait la tension référence interne est proportionnelle à la température car plus cette dernière augmente plus la VREF augmente. Qu'elle soit à 2.5 ou 1.25 V, la référence interne se comporte de la même manière vis-à-vis de la température.
- 2) Une hausse de 25°C à 185°C se traduit par une hausse de 1.5 % de la valeur de la tension de référence et la résolution passe de 298 e-9 à 302 e-9 V (VREF = 2.5V). Ceci se répercute par une erreur sur la mesure qui lit une valeur inférieure à la valeur réelle; cela n'a rien à voir avec la précision de la mesure au niveau influence de bruit mais c'est une erreur non constante (augmente avec l'augmentation de la valeur à mesurer) due à la variation de VREF qui vient se soustraire de la valeur réelle.
- 3) D'après les mesures effectuées sur oscilloscope, la fluctuation de la référence interne reste minimale car il faut observer le signal dans un petit intervalle de temps (échelle 2 ns) pour remarquer les variations.

Ainsi la tension de référence augmente avec la température et reste stable à température constante même si celle-ci est élevée.

Pour pouvoir utiliser la référence interne il faut introduire une correction sur la résolution en la calculant à partir d'un tableau de mesures de la référence interne, dans des intervalles de température réguliers et rapprochés, qu'il faudra effectuer.

### **VI.2.7 Conclusion :**

La température n'influe pas à grande échelle sur la précision de la mesure mais elle influe sur la tension de référence interne qui en augmentant crée un décalage entre le code et son équivalent en volt.

Pour une utilisation optimale de l'ADC en haute température il faut choisir une fréquence d'acquisition  $F_{Data}$  inférieure à 5 Hz, activer le buffer, utiliser une tension de référence externe stable et à faible bruit, privilégiant un gain d'amplification égal à 1 si les valeurs à mesurer le permettent.

## **VI.3 EVALUATION DE LA SOURCE DE COURANT DIGITALE IDAC EN HAUTE TEMPERATURE :**

Une source de courant digitale peut être utile dans un système car elle est commandée par programmation. Au niveau des applications Schlumberger, un outil de mesure (logging) met en relation plusieurs cartes électroniques et des éléments mécaniques (moteurs, pistons,...) ; on est quelque fois confronté à commander ou actionner un élément ou composant électronique à partir d'une carte précédente et dans ce cas l'IDAC peut s'avérer une solution efficace.

### **VI.3.1 Méthode d'évaluation :**

Pour enclencher la source de courant digitale il faut affecter une valeur au registre **IDAC** ; ensuite il reste juste à mesurer le courant de sortie analogique IDAC ( présente sur la carte d'évaluation ). Les mesures sont effectuées pour les températures {25, 85, 125, 150, 175, 185} °C. On doit, ensuite, comparer les mesures effectuées en haute température à celles faites à 25°C considérées comme référence. Pour finir, on va tracer les erreurs de linéarité différentielles (DNL) et intégrales (INL) <sup>(1)</sup> correspondant à chaque température puis les comparer entre eux.

### **VI.3.2 Mesures à effectuer :**

On prend des valeurs à intervalles réguliers (50µA dans notre cas) entre 0 et 1mA (minimum et maximum respectivement) des courants qu'on voudrait avoir en sortie; puis on calcule les valeurs correspondantes qu'il faudrait mettre dans le registre **IDAC** considérant une certaine marge due à la résolution fixe ( 3,9µA) de la source digitale (on prendra les valeurs les plus proches des valeurs désirées). Puis on mesure à l'aide d'un ampèremètre le courant de sortie.

### **VI.3.3 Résultats des mesures effectuées :**

Les résultats sont présentés dans l'Annexe C , sur les tableaux de mesure d'IDAC.

---

(1) : Les définitions et le calcul du DNL et INL sont illustrés dans l'Annexe G.

### VI.3.4 Analyse et discussion des résultats:

On remarque que l'intensité de courant de la source de courant digitale diminue avec l'augmentation de la température spécialement pour des valeurs de courants élevés; cette diminution est progressive et proportionnelle à la valeur du registre **IDAC**. Cela est expliqué par la diminution de la résolution (équivalent du LSB en  $\mu\text{A}$ ) qui passe de 3,9 à 3,63  $\mu\text{A}$  avec une augmentation de la température de 25 à 185 °C.

La référence de l'ADC n'a aucune incidence sur la référence interne de la source de courant digitale puisqu'on trouve les mêmes résultats pour les références interne et externe, en plus leurs comportements en fonction des températures sont totalement opposés puisque la référence interne du convertisseur est proportionnelle à l'augmentation de température alors que la référence interne de la source est inversement proportionnelle à l'augmentation de température

En ce qui concerne les erreurs de non linéarité différentielle (DNL) et intégrale (INL) on remarque que la première est minime car elle est de l'ordre de 0,05 bit sur les 8 bits du registre **IDAC**; cela veut dire que l'écart entre deux valeurs successives du registre **IDAC** est égal à la résolution (un LSB).

Alors que la deuxième devient importante ( 3 bits sur les 8 bits de l'**IDAC**) pour les grandes valeurs du registre **IDAC** en haute température; ceci traduit une courbure de la caractéristique « code (registre **IDAC**) - courant de sortie» pour les valeurs élevées du courant. Cette courbure se fait dans le sens de la diminution, vu le comportement de la source de courant digitale en fonction de la température, et peut être qualifiée par l'aplatissement de la caractéristique code – courant de sortie. Ainsi, plus la température est élevée plus cet aplatissement est important.

### VI.3.5 Conclusion :

La source de courant digitale est inversement proportionnelle à la température; la diminution est principalement focalisée pour les valeurs élevées du courant mais n'est pas handicapante pour autant puisqu'elle n'excède pas 10% à 185 °C. Cette source peut donc être utilisée dans certaines applications dont les besoins sont qualitatifs; Autrement dit, elle perd sa propriété de linéarité en haute température et appelle à prendre des précautions lors de l'utilisation dans des applications de haute précision.

## VI.4 EVALUATION DE LA MEMOIRE FLASH EN HAUTE TEMPERATURE :

La mémoire flash est inférieure importante au sein d'un microcontrôleur car elle héberge le programme de l'application désirée; lorsque cette dernière s'arrête, la possibilité de reprogrammer le microcontrôleur doit être envisageable même en haute température, en plus le MSC1201Y3 contient une partie réservée pour les données donc on se doit de vérifier la fiabilité de la mémoire flash en haute température

### VI.4.1 Méthode d'évaluation :

Pour évaluer la mémoire Flash, on programme plusieurs fois le microcontrôleur à différentes températures grâce à TI downloader, on mesure à chaque fois grâce à un chronomètre, le temps total de programmation:

$$\mathbf{Temps_{total} = Temps_{accès\ mémoire} + Temps_{effacement} + Temps_{écriture\ mémoire}}$$

En cas de réussite de celle-ci on relève la valeur de ce temps alors que dans le cas contraire on mentionne la cause de l'échec (message d'erreur).

On calcule, ensuite, le pourcentage de réussite à chaque température et on compare les temps de chargements mémoires.

#### VI.4.2 Tests à effectuer:

On effectue une série de dix chargements d'un programme de 191 lignes assembleur (Programme de l'ADC) dans la mémoire flash pour les températures {25, 85, 125, 150, 175, 185}°C. Deux chargements sont séparés par une durée de 30 secondes pendant laquelle le chauffage ne sera pas arrêté par contre par soucis de protection de la carte, elle est laissée au repos à température ambiante pendant 2 minutes après chaque série de tests.

On règle la fréquence de chargement sur TI downloader à 11 Mhz et la vitesse du port (baud rate) à 9600. Après chaque chargement on inscrit dans un tableau le résultat de celui-ci, sa durée en cas de réussite ou le message d'erreur en cas d'échec.

#### VI.4.3 Résultats des tests effectués :

TEMPERATURE	temps	Numéro de chargement	Résultat	Durée de chargement	Observation
25 °C	0 s	1	Réussi	9,34	
	30 s	2	Réussi	9,47	
	1mn	3	Réussi	9,56	
	1mn 30	4	Réussi	9,81	
	2mn	5	Réussi	9,5	
	2mn 30	6	Réussi	10,01	
	3mn	7	Réussi	9,68	
	3mn 30	8	Réussi	9,71	
	4mn	9	Réussi	9,66	
	4mn 30	10	Réussi	9,5	
85 °C	0 s	1	Réussi	9,83	
	30 s	2	Réussi	10,09	
	1mn	3	Réussi	9,61	
	1mn 30	4	Réussi	9,8	
	2mn	5	Réussi	9,82	
	2mn 30	6	Réussi	9,9	
	3mn	7	Réussi	9,85	
	3mn 30	8	Réussi	9,98	
	4mn	9	Réussi	9,68	
	4mn 30	10	Réussi	9,85	
125 °C	0 s	1	Réussi	10	
	30 s	2	Réussi	9,73	
	1mn	3	Réussi	9,79	
	1mn 30	4	Réussi	9,84	
	2mn	5	Réussi	9,97	
	2mn 30	6	Réussi	10,04	
	3mn	7	Réussi	10,01	
	3mn 30	8	Réussi	9,55	
	4mn	9	Réussi	9,8	
	4mn 30	10	Réussi	9,82	

150 °C	0 s	1	Réussi	10	
	30 s	2	Réussi	9,57	
	1mn	3	Réussi	9,55	
	1mn 30	4	Réussi	9,76	
	2mn	5	Réussi	9,61	
	2mn 30	6	Réussi	10,04	
	3mn	7	Réussi	9,62	
	3mn 30	8	Réussi	9,86	
	4mn	9	Réussi	9,94	
	4mn 30	10	Réussi	9,75	
175 °C	0 s	1	Réussi	10,22	
	30 s	2	Echoué	ERROR	FLASH MEMORY NOT ERASED
	1mn	3	Réussi	12,64	Temps de chargement élevé.
	1mn 30	4	Echoué	ERROR	FLASH MEMORY NOT ERASED
	2mn	5	Echoué	ERROR	FLASH MEMORY NOT ERASED
	2mn 30	6	Réussi	10,18	
	3mn	7	Echoué	ERROR	FLASH MEMORY NOT ERASED
	3mn 30	8	Réussi	9,54	
	4mn	9	Réussi	9,8	
	4mn 30	10	Réussi	9,65	
185 °C	0 s	1	Echoué	ERROR	WRIGHT REGISTER FAILED
	30 s	2	Echoué	ERROR	FLASH MEMORY NOT ERASED
	1mn	3	Echoué	ERROR	FLASH MEMORY NOT ERASED
	1mn 30	4	Echoué	ERROR	FLASH MEMORY NOT ERASED
	2mn	5	Echoué	ERROR	FLASH MEMORY NOT ERASED
	2mn 30	6	Echoué	ERROR	WRIGHT REGISTER FAILED
	3mn	7	Echoué	ERROR	FLASH MEMORY NOT ERASED
	3mn 30	8	Echoué	ERROR	WRIGHT REGISTER FAILED
	4mn	9	Echoué	ERROR	WRIGHT REGISTER FAILED
	4mn 30	10	Echoué	ERROR	FLASH MEMORY NOT ERASED

Tableau VI.6 : Influence de la température sur la mémoire flash.

#### VI.4.4 Analyse et discussion des résultats :

En observant les résultats obtenus, deux plages de température se distinguent :

La première est localisée entre 25 et 150°C et affiche un résultat de 100% de réussite ; cela veut dire que tous les chargements en mémoire flash se sont terminés avec succès dans toutes les phases d'accès mémoire, d'effacements et d'écritures mémoire. En ce qui concerne les temps de chargements, ils se situent dans l'intervalle [9.34 , 10.09] secondes avec un grand nombre autour de 9,70 secondes, on ne remarque aucun ralentissement majeur de la vitesse de chargement. Ainsi on peut dire que la mémoire flash reste fiable jusqu'à 150°C.

La deuxième plage de température se situe au delà de 150°C où les premiers cas d'échec commencent à apparaître à 175°C puisque quatre chargements sur dix se sont terminés par un échec (60 % de réussite) et tous sont dus à des erreurs d'effacement de la mémoire flash.

En plus, les chargements réussis paraissent plus lents, notamment le deuxième, puisqu'il dure trois secondes de plus que le temps moyen de chargement dans ces conditions. A 185°C, tous les chargements ont échoué (0 % de réussite); plus de la moitié des échecs sont dus à des problèmes d'effacement (représente 60% du taux d'échec) alors que les autres sont dus à des erreurs d'écriture de la mémoire flash ( 40% du taux d'échec).

On peut dire alors que la mémoire flash n'est pas du tout fiable au delà de 150°C.

#### **VI.4.5 Conclusion :**

La mémoire flash garde toutes ses propriétés de fiabilité jusqu'à 150°C ; au-delà, la programmation devient difficile. C'est pour cela qu'il est conseillé d'écrire dans la mémoire flash à des températures inférieures ou égales à 150°C.

#### **VI.5 EVALUATION DE L'HORLOGE INTERNE :**

L'horloge définit la vitesse du système (microcontrôleur), donc il est important de connaître son degré de dégradation en haute température.

Bien sûr, on évaluera l'horloge interne (oscillateur RC interne) puisqu'on s'intéresse aux caractéristiques internes du microcontrôleur.

##### **VI.5.1 Méthode d'évaluation :**

Pour évaluer l'horloge interne il faut tout d'abord l'activer grâce au registre de configuration hardware *HCR2* puis mettre le signal d'horloge en sortie d'un port d'entrées/sorties pour pouvoir mesurer sa fréquence avec un oscilloscope. Ceci est possible en affectant la valeur (08)<sub>h</sub> au registre *PASEL* pour avoir le signal SYSCLK en sortie de la broche P3.6 (la pin 6 du port 3).

On note les résultats obtenus à différentes températures puis on trace la caractéristique : fréquence d'horloge interne en fonction de la température ; cette courbe traduit les dégradations de l'horloge en haute température.

##### **VI.5.2 Mesures à effectuer:**

On choisit le mode 'oscillateur interne basse fréquence' et on devrait aboutir sur un signal de 15 Mhz en sortie. On mesure, ensuite, la fréquence d'horloge pour les températures {25, 85, 125, 150, 175, 185}°C puis on trace la courbe  $F_{SYS} = f(T)$  où  $F_{SYS}$  est la fréquence du système et T la température en °C.

##### **VI.5.3 Résultats des mesures effectuées :**

Veillez observer les résultats sur la page suivante.

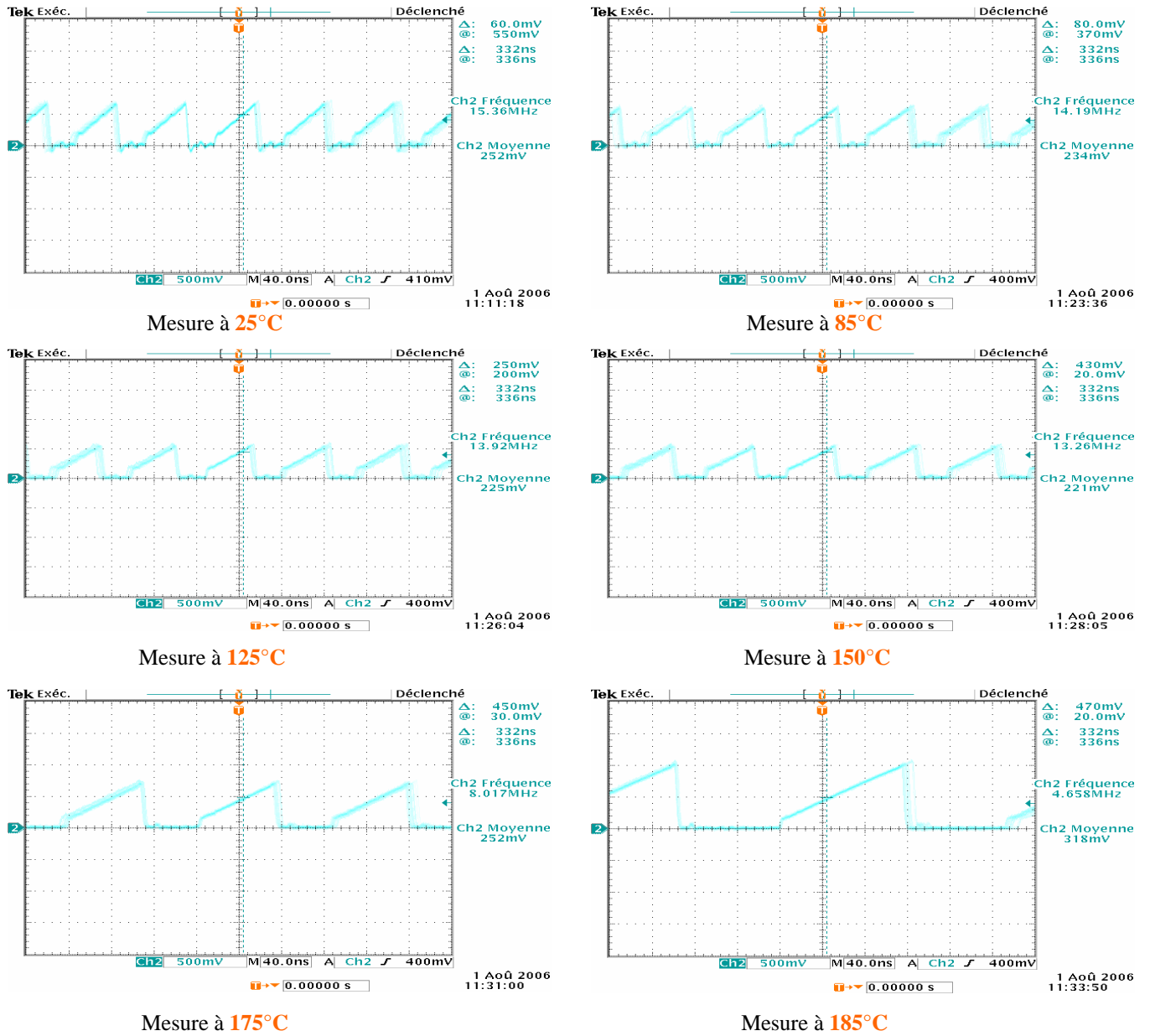


Figure VI.11 : Mesures de l'horloge interne prises à différentes températures.

Température (°C)	F <sub>sys</sub> (MHz)
25	15,36
85	14,19
125	13,92
150	13,26
175	8,017
185	4,658

Tableau VI.7: Influence de la température sur l'horloge interne.

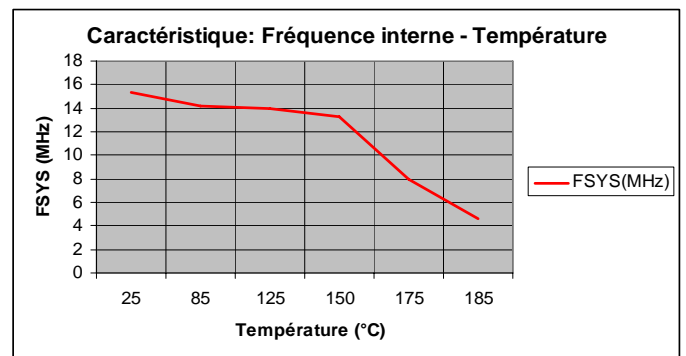


Figure VI.12: Caractéristique  $F_{sys} = f(T)$  pour le microcontrôleur MSC1201Y3.

#### VI.5.4 Analyse et discussion des résultats :

La première remarque qu'on peut faire c'est que le signal d'horloge ne se déforme pas quand la température est élevée ; par contre la fréquence diminue énormément au delà de 150°C.

En effet, la diminution de la fréquence interne, quand la valeur de la température est inférieure à 150°C, n'excède pas les 15 % (fréquence à 25°C prise comme référence), ce qui pourrait être considéré comme acceptable; par contre à 175°C et 185°C la fréquence passe à 8MHz et 4MHz respectivement. Donc on perd plus de 70% de la vitesse de traitement dans un intervalle de 35°C.

Ainsi on distingue deux zones de variation de la fréquence en fonction de la température, la première décrit une diminution faible de la fréquence d'horloge interne alors que la deuxième est caractérisée par une dégradation importante de celle-ci. On peut en conclure que le microcontrôleur MSC1201 peut être utilisé pour des applications qui requièrent une certaine vitesse de traitement jusqu'à 150°C sans préjudice majeur, au-delà l'horloge subit trop de dégradations pour donner des résultats acceptables.

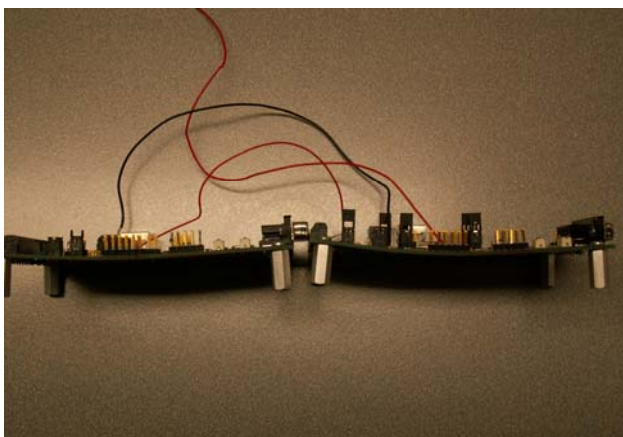
#### VI.5.5 Conclusion :

La fréquence de l'horloge interne diminue quand la température augmente ; cette diminution n'est pas très gênante à des températures inférieures à 150°C mais devient très importante au delà de cette température. C'est pour cela qu'il est conseillé d'utiliser en haute température une horloge externe stable pour éviter tout problème lié à la vitesse de traitement du MSC1201Y3.

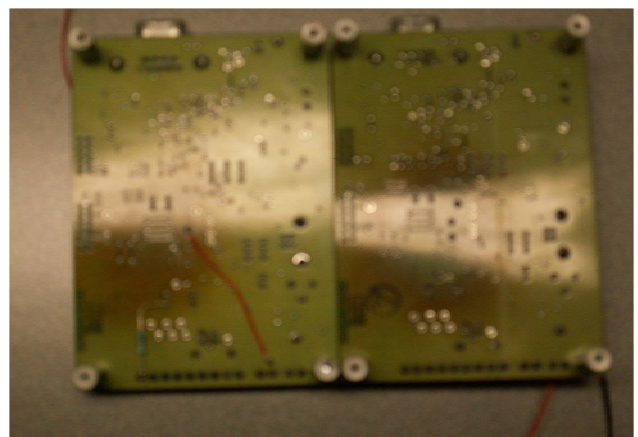
#### VI.6 POBLÈMES RENCONTRÉS DURANT L'ÉVALUATION DU MSC1201Y3 :

Plusieurs désagréments se sont présentés pendant l'évaluation du microcontrôleur MSC1201Y3. Le plus important est la dégradation de deux cartes d'évaluation sous l'effet de la température ; les cartes n'étaient pas seulement brûlées autour du microcontrôleur mais avaient subi une déformation mécanique due à la détente provoquée par la chaleur (Voir *Figure VI.13*).

La déformation a entraîné le décollement de quelques pattes du microcontrôleur (soudures cassées) et par conséquent la détérioration des cartes d'évaluation.



a) Déformations thermomécaniques du circuit imprimé



b) Brûlure du circuit imprimé autour du microcontrôleur

*Figure VI.13 : Dégradation des cartes d'évaluation sous l'effet de la chaleur.*



Un autre problème est survenu durant l'évaluation du microcontrôleur, il s'agit de l'arrêt de fonctionnement du RESET matériel pour des températures supérieures à 160°C. En effet, après avoir soumis une carte d'évaluation à des tests en haute température durant quelques heures, le RESET fonctionnait environ une fois sur deux à 175 et 185°C. Cela nous a poussé à effectuer quelques tests supplémentaires qui ont montré un dysfonctionnement du circuit RESET à ces températures. Ainsi, la précaution de commander trois cartes d'évaluation était plus que bénéfique puisqu'elle a permis de faire un grand nombre de mesures en évitant la perte de temps.

## CONCLUSION GENERALE

Pour avoir un fonctionnement fiable du microcontrôleur MSC1201Y3, il est conseillé de suivre les consignes suivantes :

- En cas d'utilisation du convertisseur analogique numérique, il est préférable d'opter pour une acquisition lente ( $F_{Data} < 5$  Hz), un gain d'amplification égal à 1 et une référence externe stabilisée.
- Pour toute application au delà de 150°C, il vaut mieux utiliser une horloge externe pour éviter tout désagrément concernant la vitesse d'exécution.
- La source de courant digitale, peut être mise en œuvre dans un système reliant plusieurs cartes en prenant un certain nombre de précautions visant à corriger les erreurs de non linéarité en haute température.
- L'écriture dans la mémoire flash à des températures supérieures à 150°C est totalement déconseillée, ainsi il vaut mieux programmer le microcontrôleur en dessous de cette limite.
- Enfin, lorsque le composant est utilisé en haute température il vaut mieux le protéger avec un gel à base de silicone pour alléger l'impact de la température sur ses différentes fonctions.

Dans l'ensemble, ce microcontrôleur est assez robuste à haute température car il donne des résultats acceptables. Il est conseillé de suivre les consignes ci-dessus pour avoir un fonctionnement optimal. Les résultats obtenus et ces consignes peuvent faire office de documentation du composant MSC1201Y3 en haute température.

Ce stage fut très intéressant et enrichissant car il m'a permis de côtoyer le monde de l'industrie électronique et de l'instrumentation au sein d'une entreprise de renommée mondiale. J'ai aussi acquis un certain savoir faire dans le domaine de l'électronique haute température, des relations de travail avec les fournisseurs et les collègues. Il m'a permis d'acquérir une certaine autonomie et responsabilité lors d'une prise de décision.

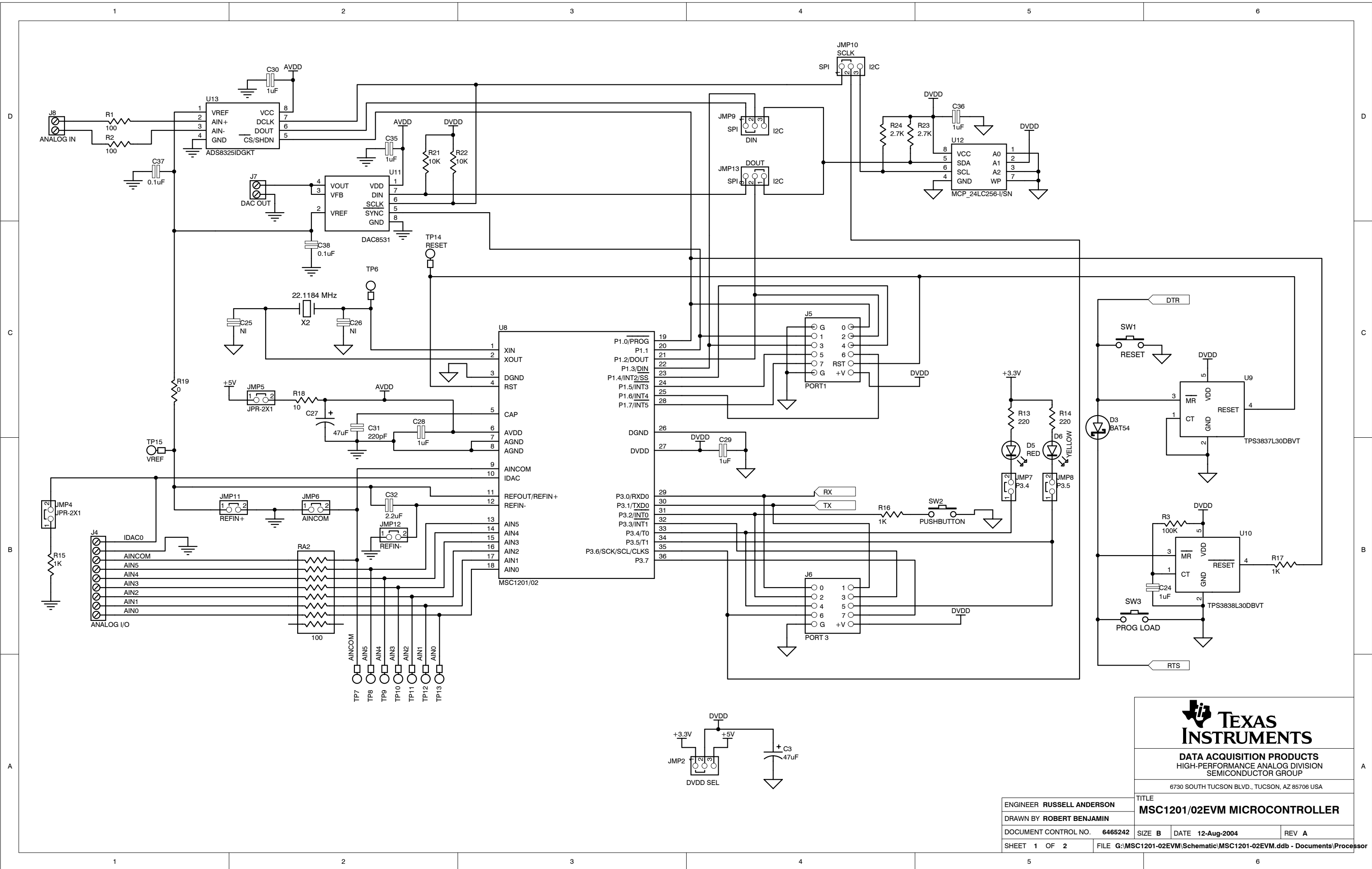
L'électronique haute température est un domaine qui m'a vraiment intéressé, parce qu'il touche à des industries lourdes, et par ce qu'il reste beaucoup à faire dans ce domaine permettant le développement de tout type d'applications en haute température.

## REFERENCES BIBLIOGRAPHIQUES

- [1] Barbara F., “High Temperature Electronics Reliability”, Groupe Techno Electronic – SRPC, Mars 2004.
- [2] Patrick Mc Cluskey F., Grzybowski R., Podlesak T., “High Temperature Electronics” - CRC Press, NY,1997.
- [3] Parmentier B., ”The Arrhenius Relation For Electronics In High Temperature Conditions”, Groupe Techno Electronic – SRPC, Juillet 2001.
- [4] Patrick McCluskey F., “High Temperature Packaging Technology, Passives, and Reliability”, CALCE Electronic Products and Systems Center University of Maryland, College Park, MD 20742, Mai 2003.
- [5] Encyclopédie Electronique en ligne – Diffusion : <http://www.techno-science.net/>
- [6] Electronique International – Latch up : <http://www.electronique.biz/>
- [7] Wikipedia Encyclopedia – Electromigration : <http://en.wikipedia.org/wiki/Electromigration>
- [8] Datasheet du MSC1201Y3 : <http://focus.ti.com/docs/prod/folders/print/msc1201y3.html>
- [9] Physique Appliquée – Convertisseurs Analogiques Numériques : <http://www.syscope.net/elec/>
- [10] Encyclopédie Wikipédia – Intel 8051 : <http://fr.wikipedia.org/wiki/8051>
- [11] Site officiel d’Analog Devices : <http://www.analog.com/>
  - Aduc C70XX : [http://www.analog.com/en/prod/0.,762\\_0\\_ADUC7128%2C00.html](http://www.analog.com/en/prod/0.,762_0_ADUC7128%2C00.html)
  - Aduc C8XX : [http://www.analog.com/en/prod/0.,762\\_0\\_ADUC836%2C00.html](http://www.analog.com/en/prod/0.,762_0_ADUC836%2C00.html)
- [12] Site officiel d’Atmel Corporation : <http://www.atmel.com/>
  - AT91 : [http://www.atmel.com/dyn/products/product\\_card.asp?part\\_id=1988](http://www.atmel.com/dyn/products/product_card.asp?part_id=1988)
- [13] Site officiel de Cygnal Integrated Products : <http://www.silabs.com/tgwWebApp/public/index.htm>  
C8051FXXX: [http://www.silabs.com/public/documents/tpub\\_doc/dsheet/Microcontrollers/Precision\\_Mixed-Signal/en/C8051F12x-13x.pdf](http://www.silabs.com/public/documents/tpub_doc/dsheet/Microcontrollers/Precision_Mixed-Signal/en/C8051F12x-13x.pdf)
- [14] Site officiel de Dallas Maxim : <http://www.maxim-ic.com/>
  - DS80C320/323 : <http://pdfserv.maxim-ic.com/en/ds/DS80C320-DS80C323.pdf>
  - MAXQ3120 : <http://pdfserv.maxim-ic.com/en/ds/MAXQ3120.pdf>
  - MAXQ2000 : <http://pdfserv.maxim-ic.com/en/ds/MAXQ2000-MAXQ2000-RBX.pdf>
- [15] Site officiel de Freescale semiconductor : <http://www.freescale.com/>
- [16] Site officiel de Fujitsu : <http://www.fujitsu.com/us/>
  - MB95XXX : <http://www.fujitsu.com/downloads/MICRO/fma/pdf/e712551.pdf>
  - MB90XXX : <http://www.fujitsu.com/downloads/MICRO/fma/pdf/e713734.pdf>

- [17] Site officiel d'Infineon : <http://www.infineon.com/cgi-bin/ifx/portal/ep/home.do?tabId=0>  
- XC800 : [http://www.infineon.com/upload/Document/XC88xCLM\\_ds\\_v01.pdf](http://www.infineon.com/upload/Document/XC88xCLM_ds_v01.pdf)  
- XC166 : [http://www.infineon.com/upload/Document/AIM/Microcontroller/16bit/xc164\\_16\\_ds\\_v2.2\\_2006\\_03.pdf](http://www.infineon.com/upload/Document/AIM/Microcontroller/16bit/xc164_16_ds_v2.2_2006_03.pdf)
- [18] Site officiel de Lattice : <http://www.latticesemi.com/>
- [19] Site officiel de Microchip : <http://www.microchip.com/>  
- dsPIC30F6014A : <http://ww1.microchip.com/downloads/en/devicedoc/70143b.pdf>
- [20] Site officiel de National Semiconductor : <http://www.national.com/>
- [21] Site officiel de Nec Electronics: <http://www.necel.com/>
- [22] Site officiel de Renesas : <http://www.renesas.com/>
- [23] Site officiel de Toshiba : <http://www.toshiba.co.jp/worldwide/>
- [24] Encyclopédie Wikipédia – ARM : <http://fr.wikipedia.org/wiki/ARM>
- [25] Encyclopédie Wikipédia – RISC : <http://fr.wikipedia.org/wiki/RISC>
- [26] Site officiel de National Instruments - Labview : <http://www.ni.com/upgrade.htm>
- [27] Site officiel de National Instruments - Labwindows CVI : <http://www.ni.com/lwcvif/>
- [28] Wikipedia Encyclopedia – SOI : [http://en.wikipedia.org/wiki/Silicon\\_on\\_insulator](http://en.wikipedia.org/wiki/Silicon_on_insulator)
- [29] Wikipedia Encyclopedia – SIC : [http://en.wikipedia.org/wiki/Silicon\\_carbide](http://en.wikipedia.org/wiki/Silicon_carbide)

# ANNEXE A

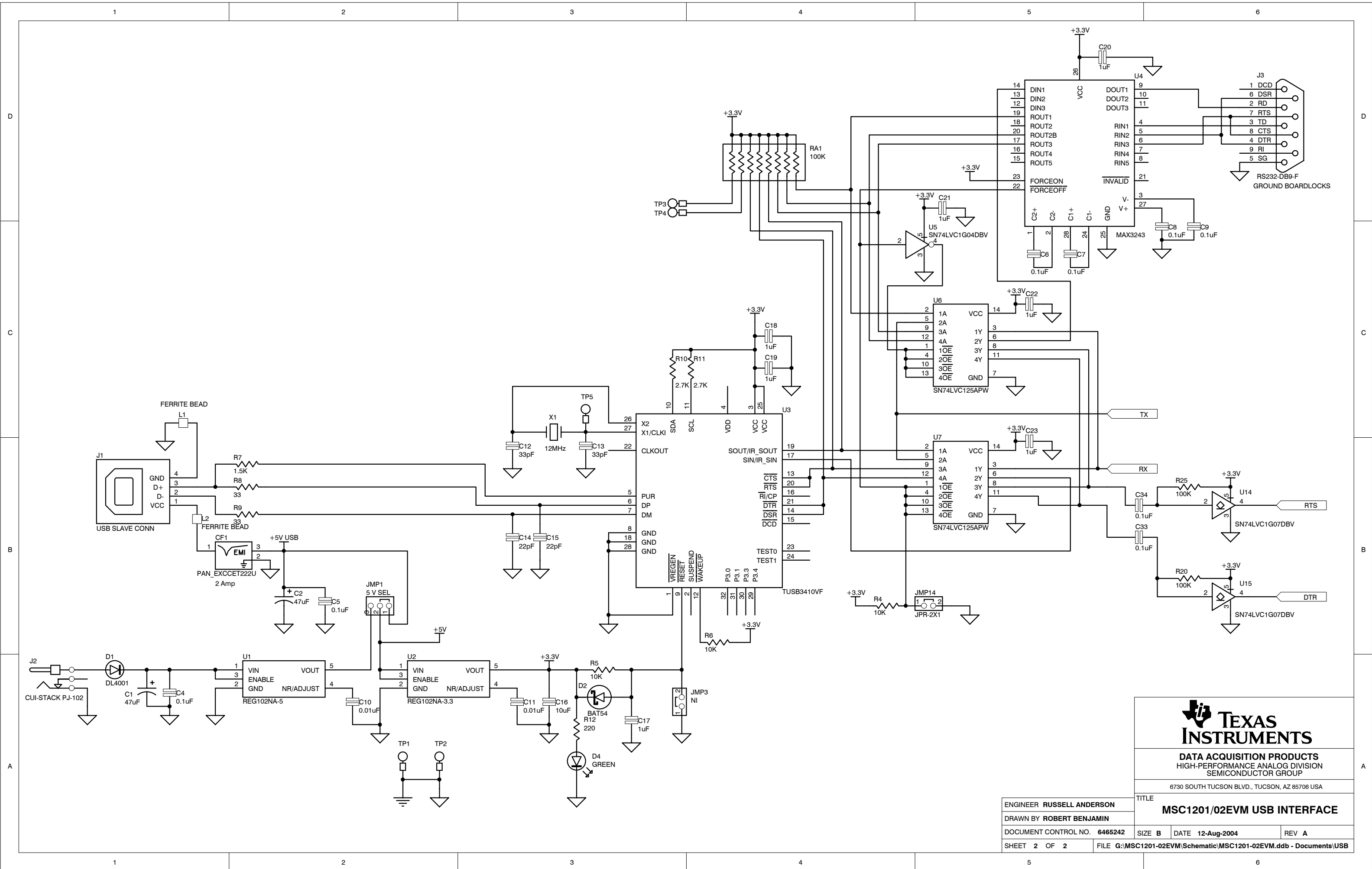



**TEXAS INSTRUMENTS**

**DATA ACQUISITION PRODUCTS**  
HIGH-PERFORMANCE ANALOG DIVISION  
SEMICONDUCTOR GROUP

6730 SOUTH TUCSON BLVD., TUCSON, AZ 85706 USA

ENGINEER <b>RUSSELL ANDERSON</b>		TITLE	
DRAWN BY <b>ROBERT BENJAMIN</b>		<b>MSC1201/02EVM MICROCONTROLLER</b>	
DOCUMENT CONTROL NO. <b>6465242</b>	SIZE <b>B</b>	DATE <b>12-Aug-2004</b>	REV <b>A</b>
SHEET <b>1</b> OF <b>2</b>	FILE G:\MSC1201-02EVM\Schematic\MSC1201-02EVM.ddb - Documents\Processor		




**TEXAS INSTRUMENTS**  
 DATA ACQUISITION PRODUCTS  
 HIGH-PERFORMANCE ANALOG DIVISION  
 SEMICONDUCTOR GROUP  
 6730 SOUTH TUCSON BLVD., TUCSON, AZ 85706 USA

ENGINEER <b>RUSSELL ANDERSON</b>		TITLE	
DRAWN BY <b>ROBERT BENJAMIN</b>		<b>MSC1201/02EVM USB INTERFACE</b>	
DOCUMENT CONTROL NO. <b>6465242</b>	SIZE <b>B</b>	DATE <b>12-Aug-2004</b>	REV <b>A</b>
SHEET <b>2</b> OF <b>2</b>	FILE G:\MSC1201-02EVM\Schematic\MSC1201-02EVM.ddb - Documents\USB		

# ANNEXE B



a) *Coté microcontrôleur :*

Le code est comme suit :

```

/*****
 * This program has as a goal to drive the ADC of the MSC1201 by a computer via a graphical *
 * interface. This program was written by Billal Adel Boussadia. *
 *****/
*****Beginning*****/

#include <REG1200.H>
#include <stdio.h>
#include <stdlib.h>
#define LSB 298 e-9 // LSB value at Vref = 2.5 V
#include "ISD51.H"
extern char rx_byte(void);
short AQIS[3];
char start;
void main (void)
{ int i,j;
  LVDCON = 0x88; // switch off low voltage detect
  CKCON = 0x10; // CLK for USART
  PCON = 0x80; // serial port 0 baud rate doubler enabled
  SCON0 = 0x40; // serial port 0 control mode 1 ,receive disabled
  TMOD = 0x20; // timer 1 mode 2: 8-bit counter with auto reload
  TH1 = 0xEE; // timer 1 MSB set baud rate 19200bps on msc1201
  TR1 = 1; // run timer 1
  TI = 1; // send first char to usart
  PDCON &= ~0x48; // Turn on the ADC
  ACLK = 10; // Fdata = 1.28 Hz
  ADCON2=0xFF;
  ADCON3=0x07;
  ADMUX = 0x08; // AINP = AIN0, AINN = AINCOM
  ADCON0 = 0x10; // External Vref , Buffer Off, PGA=1
  ADCON1 = 0x31; // sinc3 filter
  while(1)
  {
    start=rx_byte(); // read char
    switch (start)
    {
      case 's':
        AQIS[0]=ADRESL; // if 's'==> convert
        AQIS[1]=ADRESM;
        AQIS[2]=ADRESH;
        for(i=0;i<3;i++)
        {
          SBUF0=AQIS[i]; // send to usart buffer
          for(j=0;j<1000;j++); // tempo
        }
        break;
      case 'a':
        TH1 = 0xEE; //Initialise
        TR1 = 1;
        TI = 1;
        break;
    }
  }
}
/*****END*****/

```

b) *Coté ordinateur* (interface graphique):

Le code se présente sous la forme suivante:

```

/*****
 *   This is the second part of the program that permits to drive the MSC1201 ADC via a
 *   graphical interface. This program was written by Billal Adel Boussadia
 *****/
/*****Beginning*****/
#include <.h>
#include "control Panel.h"
#include <utility.h>
#include <ansi_c.h>
#include <formatio.h>
#include <cvirte.h>
#include <userint.h>
static int panelHandle;
#include <rs232.h> // serial port library
unsigned char AQIS[3];
int bytes_read,i=0,j=0,k=0,l=0,m=0,n=0,x,ComPort=1;
#define SER_BUF_LENGTH 255
int controlFileHandle = 0;
unsigned char b1[100],
unsigned long t;
#define LSB 298e-9
/*****Opening serial port subroutine*****/
int SER_OpenCOM( void )
{
    int Ret = 1;
    Ret = OpenComConfig (ComPort, "COM1", 19200, 0, 8, 1, SER_BUF_LENGTH, SER_BUF_LENGTH);
    if ( Ret ) MessagePopup ("Serial Port Open", "Error Opening Serial Port");
    return Ret;
}
/*****Closing serial port subroutine*****/
int SER_CloseCOM( void )
{
    return ( CloseCom(ComPort) );
}
/*****START/STOP Button definition Subroutine*****/
int CVICALLBACK callstart( int panel, int control, int event,
void *callbackData, int eventData1, int eventData2)
{ int BUTTON;

if(event!=EVENT_COMMIT)
return 0;

GetCtrlVal( PANEL, PANEL_TOGGLEBUTTON, &BUTTON );
if ( BUTTON )
    { for(x=0;x<3;x++) AQIS[x]=0; // initialise reception variables
SER_OpenCOM(); // open serial port
SetCtrlAttribute( PANEL,PANEL_TIMER, ATTR_ENABLED, 1 ) ; // enable reception timer
    }
else
    {
SetCtrlAttribute( PANEL, PANEL_TIMER, ATTR_ENABLED, 0 );// Disable reception timer
FlushInQ(ComPort); //erase serial port
b1[0]='a' ;
ComWrt (ComPort, b1, 1); // send character 'a'
CloseFile (controlFileHandle); // close acquisition file
SER_CloseCOM(); // close serial port
    }

return 0;
}

```

```

/*****Quit button definition subroutine*****/
int CVICALLBACK callquit (int panel, int control, int event,
                          void *callbackData, int eventData1, int eventData2)
{
    if (event!=EVENT_COMMIT)
        SER_CloseCOM(); // Close serial port
        QuitUserInterface (0); // quit interface
    return 0;
}

/*****Acquisition subroutine*****/
int CVICALLBACK calltimer (int panel, int control, int event,
                           void *callbackData, int eventData1, int eventData2)
{
    int ret;float datapoint[1]={0.0};
    char DataToFile [2048];
    char *fileName;
    char file_name[1024];

if(event!=EVENT_TIMER_TICK)
    return 0;
fileName = DateStr(); // Open EXEL
filestrcat (fileName, ".xls"); strcpy (file_name, fileName); // storing datas
controlFileHandle = OpenFile (file_name, VAL_WRITE_ONLY, VAL_APPEND, VAL_ASCII); // by days.
b1[0]='s';
ComWrt (ComPort, b1, 1); // send character 's'
SetCtrlAttribute( PANEL, PANEL_TIMER, ATTR_ENABLED, 0 ); // disable reception timer
bytes_read = ComRd (ComPort, AQIS, 3); // data acquisition
if( bytes_read==3) // test if 3 bytes were received
    {
        SetCtrlVal (PANEL, PANEL_NUMERIC_2, AQIS[0]); // print the Low byte
        SetCtrlVal (PANEL, PANEL_NUMERIC_3, AQIS[1]); // print the middle byte
        SetCtrlVal (PANEL, PANEL_NUMERIC_4, AQIS[2]); // print the high byte
        t=(unsigned long)(AQIS[2])<<8;
        t=(t+(unsigned long)(AQIS[1])<<8;
        t=t+(unsigned long)(AQIS[0]);
        SetCtrlVal (PANEL, PANEL_NUMERIC, t*LSB); // print the result in volt
        datapoint[0] = t*LSB;

        while(i>10)
            {
                PlotStripChart (PANEL, PANEL_GRAPH, datapoint,1,0,0,VAL_FLOAT); // plot the result every
                i=0; // 10 acquisitions
                } i++;
        SetCtrlAttribute( PANEL, PANEL_TIMER, ATTR_ENABLED, 1); // enable timer
        sprintf(DataToFile,"%s \t %f \t %lu \n",TimeStr(),t*LSB,t); // print time in reception file
        WriteFile (controlFileHandle, DataToFile, strlen(DataToFile)); // print results in reception file
    }
else
    {
        for(x=0;x<3;x++) AQIS[x]=0; // clear reception variables
        SetCtrlVal (PANEL, PANEL_NUMERIC_2, AQIS[0]); //
        SetCtrlVal (PANEL, PANEL_NUMERIC_3, AQIS[1]); // clear all the numeric panels
        SetCtrlVal (PANEL, PANEL_NUMERIC_4, AQIS[2]); //
        t=(unsigned long)(AQIS[2])<<8;
        t=(t+(unsigned long)(AQIS[1])<<8;
        t=t+(unsigned long)(AQIS[0]);
        SetCtrlVal (PANEL, PANEL_NUMERIC, t*LSB);
        SetCtrlAttribute( PANEL, PANEL_TIMER, ATTR_ENABLED, 1); // enable timer for a new acquisition
    }
return 0;
}









```






```

/*****Clear button definitin subroutine*****/
int CVICALLBACK callclear (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
    if(event!=EVENT_COMMIT)
    return 0;
    ClearStripChart (PANEL, PANEL_GRAPH); // clear graph
    SetCtrlVal (PANEL, PANEL_NUMERIC, 0.0 ); //
    SetCtrlVal (PANEL, PANEL_NUMERIC_2, 0 ); // clear all numeric panels
    SetCtrlVal (PANEL, PANEL_NUMERIC_3, 0 ); //
    SetCtrlVal (PANEL, PANEL_NUMERIC_4, 0 ); //
    return 0;
}
/*****Main Program*****/
int main (int argc, char *argv[])
{
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1;
    if ((panelHandle = LoadPanel (0, "control Panel.uir", PANEL)) < 0)
        return -1;
    DisplayPanel (panelHandle);
    RunUserInterface ();
    DiscardPanel (panelHandle);
    return 0;
}
/*****END*****/

```

# ANNEXE C

Constructeurs	Type	MCU Core	Technology	Resolution (Bits)	Speed (MIPS)	RAM (O)	Memoire Prog (EPROM,FLASH...)	TIMERs N x bits	ADC Bits / Speed(KSPS)	DAC (Bits)	Operating Temp. Min / Max (C°)	Storage Temp. Min / Max (C°)	Consummation (V) / (mA/MHz)	Peripheriques	Boitier type/pins	Autres
 ANALOG DEVICES	ADuC70XX	ARM7 TDMI	-	12	1.3-16 -20 -40	8192	62 K	4 x 8 (Usage general)	12 / 1000	12	- 40 / 125	- 65 / 150	3 / 0.95 - 2.2	- UART - SPI - I2C - 4 parallel Port	LFCSP / 40 LQFP / 80	- PLA - PSM - WDT
	ADuC8XX	8052 (1/12-8k)	-	12-16-24	1-1.3-12-16-20	256-2304	5K-62K	3 x 16	12-16/24 SD / 0.105-400	12	- 40 / 85 - 40 / 125	- 65 / 150	3 / 2 3.6 / 1.465 3.6 / 3.057	- UART - SPI - I2C - 4 parallel Port	LFCSP / 52 MQFP / 56	- PSM - WDT
 ATMEL Corporation	AT91 ( M. R. RM ,SAM ) ,XX...	ARM7 TDMI	High density CMOS	8-16-32	200 (a 180 MHz)	[ 4 K - 256 K ]	[ 32K - 256 K ]	3-6-9-10 x 16	No	No	- 40 / 85	- 60 / 150	[ 8 - 3.6 ] / NS	- 2UART/USART - SPI - parallel port (PIO)	LQFP / 100-144-176 LBGA / 144 LFBGA / 176 - 256 PQFP / 208	- WDT - RTT - POR
 CYGNAL Integrated Products	C8051FXXX	High speed 8051	-	8	20-25-48 -50-100	[ 256 - 8448 ]	[ 2K - 128K ]	2-3-4-5 x 16	8 / 100-500 10 / 100-200 12 / 100 16 / 1 24 / 1	8-12-16	- 55 / 125 - 40 / 85	- 65 / 150	[ 2.7 - 3.6 ] / 1	- 2 UART - SPI - SMBus	TQFP / 48-64-100 LQFP / 32 QFN / 11-20-28	
 DALLAS Semiconductor	DS80C320 DS80C323	8051	CMOS 6 inch	8	8,26	256	-	3 x 16	No	No	- 40 / 85	- 55 / 125	[ 4.25 - 5.5 ] / 1.2 [ 2.7- 5.5 ] / 0.55	- 2 FDHSP	DIP / 40 PLCC / 44 TQFP / 44	- WDT
	MAXQ3120	MAXQ 16bits-RISC	CMOS 6 - 8 inch	16	8	512	32 K	3 x 16	16 SD / 20.8	No	- 40 / 85	- 65 / 150	3.3 / 3.5	- UART	MQFP / 80	- WDT - RTC
	MAXQ2000	MAXQ 16bits-RISC	CMOS 6 - 8 inch	16	20	2 K	64 K	3 x 16	No	No	- 40 / 85	- 65 / 150	2.2 / 0.4	- 1/2 UART - SPI - 1 Wire Bus Master	QFN / 68 TQFN / 56	- WDT - RTC
 Freescale semiconductor	Famille HC08	High performance MC68HC08	-	8	3-3.2-4-4.1 -6-8-8.2	[ 128 - 2048 ]	[ 8 K - 62 K ]	2-3-4-6 x 8	8-10 / NS	No	- 40 / 85	- 55 / 150	[ 2.2 - 5.5 ] / NS	- I2C - SPI - SCI	LQFP / 48 QFP / 44 SDIP / 42	- RTI - WDT
	Famille HCS08	High performance MC68HC08	-	8	20	[ 256 - 4096 ]	[ 4096 - 61 K ]	2-4-6 x 8	10 / NS	No	- 40 / 85	- 55 / 150	[ 1.8 - 3.6 ] - 5 / 1.1	- I2C - SPI - SCI	LQFP / 64 QFN / 48 QFP / 44 SDIP / 42	- RTI - WDT
	Famille HC05	MC68HC05	High density CMOS	8	1.05-2.1	[ 176 - 512 ]	[ 4096 - 15 K ]	1 x 16	10 / NS	No	- 40 / 85 - 40 / 125	- 65 / 150	[ 3 - 5 ] / 0.6	- SPI - SCI	PLCC / 52 SDIP / 56 QFP / 64	
	Famille HC11 et HC711	MC68HC11	High density CMOS	8	2-3-4	[ 192 - 1024 ]	20 K	1 x 16	8 / NS	No	- 40 / 85 - 40 / 125	- 55 / 150	5 / 5	- SPI - SCI	PLCC / 52 CLCC / 52 QFP / 64 TQFP / 52 SDIP / 56 DIP / 48	
 FUJITSU	Famille MB95xxx	F2MC-8FX	CMOS	8	10	[ 1K - 4 K ]	[ 16 K - 60 K ]	2 x 16	10 / NS	No	- 40 / 85	- 55 / 150	[ 3 - 5 ] / 2	- UART - I2C	LQFP / 64	
	Famille MB99xxx	F2MC-8L	CMOS	8	[ 0.032 - 12.5 ]	[ 128 - 2 K ]	[ 4 K - 60 K ]	3 x 8 + 1 x 16 5 x 8	8-12 / NS	No	- 40 / 85	- 55 / 150	[ 3 - 5.5 ] / 2.41	- UART - I2C - USB	LQFP / 64 QFP / 48	
	Famille MB906xx	F2MC-16L	CMOS	16	[ 0.032 - 16 ]	[ 1024 - 8192 ]	[ 16 K - 256 K ]	2-5 x 16	10 / NS	No	- 40 / 85	- 55 / 150	[ 3 - 5.5 ] / 3.125	- UART - SCI	LQFP / 100 QFP / 100	
	Famille MB90xxx	F2MC-16LX	CMOS	16	[ 16 - 25 ]	[ 1024 - 6144K ]	[ 32 K - 384 K ]	2-4 x 16	10 / NS	No	- 40 / 85	- 55 / 150	[ 3 - 5.5 ] / 3.125	- 4 UART - I2C	LQFP / 100 QFP / 100	
 INFINEON	Famille XC800	High performance 8051	CMOS	8	6-12	256	24 K - 32 K	4 x 16	10 / NS	No	- 40 / 85 - 40 / 125	- 65 / 150	3.3 - 5 / NS	- SPI - 2 UART	TQFP / 48 TQFP / 64	- WDT - CAN
	Famille C500/C800	Standard 8051	CMOS	8	1.6-3.33	64-256	8 K	3 x 16	8-10 / NS	No	- 40 / 85 - 40 / 125	- 65 / 150	3.3 / 0.32	- SPI - I2C	P-TSSOP / 38 P-DSO / 28	- WDT - CAN
	Famille XC166	16-bit 5S-Pipelined C166	CMOS	16	40	4K-6K	[ 32 K - 256 K ]	5 x 16	Resolution programmable	No	- 40 / 85 - 40 / 125	- 65 / 150	5 / 12.6	- 2 SPI - I2C - 2 UART	P-TQFP-144 P-TQFP-100 PG-TQFP-64	- CAN - WDT
	Famille C166	16-bit 4S-Pipelined C166	CMOS	16	8.33	10K	16 K - 256 K	5 x 16	10 / NS	No	- 40 / 85 - 40 / 125	- 65 / 150	[ 4.5 - 5.5 ] / 17.5	- SPI - UART	P-TQFP-128 PG-MQFP-80 P-TQFP-64	- WDT - CAN - RTC
 LATTICE	Famille DF68XX	DF68XX 8 bits IP	CMOS	8	15	256	64K	1 x 16	No	No	NS	NS	NS	- UART - SPI	-	
	R8051	8051 8 bits RISC - like core	CMOS	8	24	256	-	1 x 16	No	No	NS	NS	NS	- UART	-	

	DP8051 DP8051 XP	8051 8 bits pipelined-core	CMOS	8	268 Vax MIPS	256	64K	1 x 16	No	No	NS	NS	NS	- UART - SPI - I2C - Codec		
<b>MICROCHIP</b> 	Famille PIC10F2XX	8 bits High- Performance RISC	CMOS	8	1	16 - 23 - 24	0.375 K - 0.75 K	1 x 8	8 / NS	No	- 40 / 85 - 40 / 125	- 65 / 150	[ 2 - 5.5 ] / 0.0425	-	PDIP / 8 SOT-23 / 6	- RTC - WDT
	Famille PIC12FXXX	8 bits High- Performance RISC	CMOS	8	1 - 10	25 - 38 - 41 - 64 - 128	[ 0.75 K - 1.75 K ]	1 x 8 + 1 x 16 2 x 8 + 1 x 16	8 - 10 / NS	No	- 40 / 85 - 40 / 125	- 65 / 150	/ 0.11 - 0.0425 [ 2 - 5.5 ]	-	PDIP / 8 SOIC / 8 DFN - S / 8 MSOP / 8	
	PIC14000	8 bits High- Performance RISC	CMOS	8	10	192	7 K	1 x 8 + 1 x 16	Slope / NS	No	- 55 / 125	- 65 / 150	[ 2.7 - 6 ] / 0.275	I2C	PDIP / 28 SOIC / 28 SSOP / 28	- WDT
	Familles PIC16 C/C/R/ RV XXX	8 bits High- Performance RISC	CMOS	8	10	[ 24 - 368 ]	[ 0.75K - 3584 K ]	1 x 8 1 x 8 + 1 x 16 2 x 8 + 1 x 16	8 - 10 - 12 / NS	8	- 55 / 125	- 65 / 150	[ 2 - 6.25 ] / 0.275	- I2C - USART - compatible SPI - FSP	PDIP / 14 - 18 - 20 - 28 - 40 CDIP / 20 SSOP / 14-20-28 SOIC / 18 - 28 Windowed CERDIP / 18 - 28 - 40 SDIP / 28 MQFP / 44 TOFP / 44 - 64 PLCC / 44 - 68 CLCC / 68 QFN / 14 - 16 - 28 - 44 LQFP / 80	- WDT - POR
	Familles PIC18FXXX	8 bits High- Performance RISC	CMOS	8	10	[ 256 - 3936 ]	[ 0 K - 128 K ]	1 x 8 + 2 x 16 1 x 8 + 3 x 16 2 x 8 + 3 x 16	10 / 30 - 100	No	- 40 / 125	- 65 / 150	[ 2 - 3.6 ] / TBD	- I2C - USART - compatible SPI - FSP	TOFP / 44 - 64 - 80 PLCC / 68 - 48 LQFP / 128 PDIP / 18 - 28 - 40 SOIC / 18 - 28 QFN / 28 - 44 SSOP / 20 - 28	- WDT - POR
	Famille DSPIC30F201X	16 bits High- Performance RISC	CMOS	16	20/30	512 - 1024	12 K	3 x 16	10 - 12 / 100 - 500	No	- 40 / 125	- 65 / 150	[ 2.5 - 5.5 ] / 1.4	- I2C - USART - SPI - ICSP	PDIP / 18 - 28 SOIC / 18 - 28 QFN / 28 - 44	- WDT - POR
	Famille DSPIC30F301X	16 bits High- Performance RISC	CMOS	16	20/30	1024 - 2048	24 K	3 x 16 5 x 16	10 - 12 / 100 - 500	No	- 40 / 125	- 65 / 150	[ 2.5 - 5.5 ] / 2.5	- I2C - 2 USART - SPI - ICSP	SOIC / 18 - 28 QFN / 44 PDIP / 18 - 40 TOFP / 44	- WDT - POR
	Famille DSPIC30F401X	16 bits High- Performance RISC	CMOS	16	20/30	2048	48 K	5 x 16	10 - 12 / 100 - 500	No	- 40 / 125	- 65 / 150	[ 2.5 - 5.5 ] / 2.5	- 2 UART - I2C - SPI - ICSP - CAN - CODEC	TOFP / 44 PDIP / 40 QFN / 44 SOIC / 28	- WDT - POR
	Famille DSPIC30F501X	16 bits High- Performance RISC	CMOS	16	20/30	2048 - 4096	66 K	5 x 16 + 2 x 32	10 - 12 / 100 - 1000	No	- 40 / 125	- 65 / 150	[ 2.5 - 5.5 ] / 4	- 2 UART - I2C - 2 SPI - ICSP - 2 CAN - CODEC	TOFP / 64 - 80	- WDT - POR
	Famille DSPIC30F601X	16 bits High- Performance RISC	CMOS	16	20/30	6144 - 8192	132 K - 144 K	5 x 16 + 2 x 32	10 - 12 / 200 - 1000	No	- 40 / 125	- 65 / 150	[ 2.5 - 5.5 ] / 4	- 2 UART - I2C - 2 SPI - ICSP - 2 CAN - CODEC	TOFP / 64 - 80	- WDT - POR
<b>NATIONAL</b> 	Famille COP8XXX	COP8™	CMOS	8	10	[ 64 - 1 K ]	[ 768 - 32 K ]	1x 16 3 x 16	10 / NS	No	0 / 70 - 40 / 125	- 65 / 140	[ 2.5 - 6 ] / 1.4	- USART - CAN	MDIP / 14 - 16 - 20 - 28 - 40 SOIC / 28 TSSOP / 48 PLCC / 44 - 68 LQFP / 28	
	CR16MAS9	CR16B RISC Core	CMOS	16	25	3K	64 K	2x16	8 / NS	No	- 40 / 85 - 40 / 125	- 65 / 150	[ 4.5 - 5.5 ] / 4.75	- 2 USART - SPI - CAN - Compatible I2C	PQFP / 80	- WDT
<b>NEC</b> 	Famille 78KJ / Kx2	78K0 CPU core - 8 bits CISC	CMOS	8	10	[ 512 - 7K ]	[ 8 K - 128 K ]	1 ou 2 x 8 1 x 16	10 / NS	No	- 40 / 85	- 65 / 150	[ 1.8 - 5.5 ] / NS	- UART - I2C - CSI	SSOP / 30 FPLGA / 36 - 64 QFP / 44 - 48 - 52 - 64 - 80	- WDT
<b>RENESAS</b> 	Famille 740	8-bit micro computer based on the 740 family	CMOS	8	5	[ 192 - 2048 ]	[ 8 K - 60 K ]	4 x 8	10 / NS	8	- 20 / 85	- 40 / 125	[ 1.8 - 5.5 ] / NS	- UART - I2C	QFP / 64 - 80 DIP / 64 CLCC / 64 LQFP / 64 - 80	- WDT
	Famille H8	16-bit micro computer based on the H8 family	CMOS	16	1	512	-	2 x 16	10 / NS	8	- 20 / 75	- 55 / 125	5 / 3.5	- SCI - UART - AGA	QFP / 100 TOFP / 100	- WDT
<b>TEXAS</b> 	Famille MSP430	Ultra - low - power 16-bit RISC mixed -signal processor	CMOS	16	8	[ 128 - 10240 ]	[ 1 K - 120 K ]	2 x 8 + 1 x 16 2 x 16	Slope - 10 - 12 - 14 - 16ad / NS	12	- 40 / 85	- 55 / 150	[ 1.8 - 3.6 ] / 0.33	- UART - SPI - I2C	PDIP / 14 TSSOP / 14 - 28 - 38 QFN / 16 - 24 - 40 - 48 - 64 SOIC / 20 - 28 SSOP / 48 TVSOP / 20 LQFP / 64 - 80 - 100 TOFP / 64	- WDT - DMA

	Famille MSC12XX	8051 CPU Core	CMOS	8	8	256	4 - 8 K	2 x 16	24 / 1	8	- 40 / 125	- 65 / 150	[2.7 - 5.5] / 1	- SPI - I2C - USART	QFN / 36	- IDAC
<b>TOSHIBA</b>	Famille TMP8XXXX	870/C CPU Core	CMOS	8	4	[256 - 1.5 K]	[2 K - 32 K]	2 x 8 + 1 x 16	8 / NS	No	- 40 / 85		[2.7 - 5.5] / NS	- UART - SIO	SOP / 20 - 28 SDIP / 28 - 44 LQFP / 44 - 64 QFP / 64 DIP / 20	- WDT
<b>TOSHIBA</b>	Famille TMP88CX40	870/X CPU Core	CMOS	8	5	[0.5 K - 2 K]	16 K	1 x 8 + 1 x 16	10 / NS	No	- 40 / 85		[4.5 - 5.5] / 0.9	- UART - SIO	SDIP / 28 - 42 - 64 QFP / 44 - 64 - 80 SOP / 28	- WDT
	Famille TMP91XX27	High speed 16 bits 900/L1	CMOS	16	6.75	4K	256 K	1 x 8 + 1 x 16	10 / NS	No	- 40 / 85		[1.8 - 3.6] / 0.426	- UART - I2C	LQFP / 64	- WDT

*Tableau 1 : Etude comparative selon performances des differents microcontrolleurs 8/ 16 bits*



## TEMPERATURE : 25 °C

Configuration Frequence / Buffer	PGA = 1				PGA = 64				PGA = 128						
	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)			
ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer off	0	18,7	20,775		0	15,99	17,11		0	14,96	16,17				
	0,032	18,76	19,675		0,005	16,02	16,905		0,00025	14,845	16,03				
	0,6	18,98	20,12		0,01	16,02	16,96		0,005	14,89	15,855				
	1,25	18,86	21,095		0,02	15,825	17,01		0,01	14,9	16,02				
	1,5	18,635	20,85		0,025	15,86	17,105		0,0125	14,95	16,015				
	2	18,53	20,33		0,03	15,87	17,05		0,015	14,9	16,025				
	2,3	18,4	20,26		0,035	16,03	16,94		0,0175	14,73	15,73				
	2,499	18,24	20,35		0,039	15,85	17,05		0,0195	15,02	16,16				
	MOY	18,638125	20,431875		21,6	MOY	15,933125		17,01625	20	MOY		14,899375	16,00625	19
	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	20,77		21,14		0		16,45	17,47			0	15,88	16,55
0,032		20,725	21,26	0,005	16,82		17,34	0,00025	15,9	16,47					
0,6		20,17	21,32	0,01	16,71		17,52	0,005	15,75	16,41					
1,25		19,87	21,06	0,02	16,79		17,29	0,01	15,785	16,535					
1,5		19,37	21,405	0,025	16,79		17,33	0,0125	15,7	16,49					
2		19,2	21,38	0,03	16,69		16,775	0,015	15,74	16,4					
2,3		18,965	20,98	0,035	16,7		16,93	0,0175	15,73	16,37					
2,499		18,895	20,89	0,039	16,675		17,205	0,0195	15,65	15,94					
MOY		19,745625	21,179375	21,5	MOY		16,703125	17,2325	18,8	MOY		15,766875	16,395625	18,2	
ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off		0	12,21	12,21			0	11,72	11,43			0	10,4	10,13	
	0,032	12,09	12,19	0,005		11,74	11,46	0,00025	10,04		10,1				
	0,6	12,275	12,28	0,01		11,72	11,38	0,005	10,77		10,52				
	1,25	12,09	12,13	0,02		11,65	11,34	0,01	10,65		10,51				
	1,5	11,99	11,99	0,025		11,7	11,385	0,0125	10,69		10,48				
	2	11,87	11,925	0,03		11,475	11,34	0,015	10,72		10,47				
	2,3	11,71	11,77	0,035		11,535	11,215	0,0175	10,675		10,47				
	2,499	11,51	11,45	0,039		11,325	11,15	0,0195	10,72		10,51				
	MOY	11,968125	11,993125	12,5		MOY	11,608125	11,3375	12		MOY	10,583125	10,39875	11,5	
	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125	12,125			0	11,86	11,89			0	12,25	11,9	
0,032		12,165	12,17	0,005	12,02		12,14	0,00025	10,52	10,6					
0,6		12,21	12,21	0,01	12,04		12,14	0,005	11,12	11,29					
1,25		12,09	11,92	0,02	11,83		11,94	0,01	11,18	11,21					
1,5		11,96	11,925	0,025	11,845		11,865	0,0125	11,06	11,23					
2		11,895	11,945	0,03	11,82		12	0,015	11,05	11,24					
2,3		11,69	11,77	0,035	11,68		11,79	0,0175	11,06	11,27					
2,499		11,48	11,56	0,039	11,5		11,7	0,0195	11,09	11,23					
MOY		11,951875	11,953125	12,5	MOY		11,824375	11,933125	12,5	MOY		11,16625	11,24625	12	

*Tableau 2: Calculs d'ENOB pour différentes configurations du convertisseur analogique numérique à 25°C*

## TEMPERATURE : 125 °C

Configuration Frequence / Buffer	PGA = 1				PGA = 64				PGA = 128			
	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)
ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer off	0	19,38	21,6		0	16,67	17,11		0	15,68	16,17	
	0,032	19,96	19,95		0,00025	15,34	16					
	0,6	19,815	20,285		0,005	15,8	16,38					
	1,25	19,38	21,09		0,01	16,94	17,37					
	1,5	18,415	21,255		0,02	16,75	17,73					
	2	19,24	20,47		0,025	16,56	16,89					
	2,3	18,77	20,71		0,03	16,72	17,22					
	2,499	18,64	20,35		0,035	16,78	16,88					
	MOY	19,2	20,71375		21,6	MOY	16,739375		17,2475	20	MOY	
ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,19	21,14		0	17,37	17,47		0	16,24	16,55	
	0,032	21,25	21,32		0,00025	16,265	16,47					
	0,6	20,23	21,56		0,005	16,2	16,76					
	1,25	19,59	21,3		0,01	17,36	16,94					
	1,5	19,425	21,44		0,02	17,16	17,55					
	2	18,905	21,235		0,025	17,175	17,33					
	2,3	18,99	20,925		0,03	17,27	17,48					
	2,499	18,7	20,74		0,035	17,18	17,29					
	MOY	19,785	21,2075		21,5	MOY	17,250625		17,40625	18,8	MOY	
ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0	12,23	12,21		0	11,97	11,43		0	10,41	10,13	
	0,032	12,165	12,19		0,00025	12,06	12,03					
	0,6	12,28	12,28		0,005	11,95	11,995					
	1,25	12,23	12,13		0,01	11,95	11,995					
	1,5	11,995	11,99		0,02	11,8	11,92					
	2	11,92	11,94		0,025	11,72	11,95					
	2,3	11,71	11,77		0,03	11,77	11,836					
	2,499	11,57	11,51		0,035	11,63	11,64					
	MOY	12,0125	12,0025		12,5	MOY	11,795		11,776375	12	MOY	
ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,21	12,125		0	11,99	11,89		0	11,65	11,9	
	0,032	12,18	12,105		0,00025	12,09	12,09					
	0,6	12,21	12,26		0,005	12,17	12,2					
	1,25	12,055	12,08		0,01	11,88	11,965					
	1,5	11,99	11,95		0,02	11,88	11,97					
	2	11,91	11,92		0,025	11,88	11,97					
	2,3	11,87	11,705		0,03	11,935	11,87					
	2,499	11,51	11,565		0,035	11,71	11,755					
	MOY	11,991875	11,96375		12,5	MOY	11,8875		11,9	12,5	MOY	

*Tableau 3: Calculs d'ENOB pour différentes configurations du convertisseur analogique numérique à 125°C*

## TEMPERATURE : 150 °C

Configuration Frequence / Buffer	PGA = 1				PGA = 64				PGA = 128						
	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)			
ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer off	0	19,34	19,99		0	16,65	17,18		0	15,73	16,51				
	0,032	19,44	19,9		0,00025	16,67	17,52		15,81	16,27					
	0,6	19,59	20,31		0,005	16,74	16,66		15,73	16,45					
	1,25	19,15	20,77		0,01	16,66	17,43		15,7	16,48					
	1,5	18,795	20,88		0,025	16,61	16,79		15,6	16,02					
	2	18,5	20,165		0,03	16,645	17,31		15,725	16,6					
	2,3	17,57	20,48		0,035	16,59	17,06		15,73	16,25					
	2,499	18,22	20,53		0,039	16,58	17,065		15,625	16,09					
	MOY	18,825625	20,378125		21,6	MOY	16,643125		17,126875	20	MOY		15,70625	16,33375	19
	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,36		21,67		0		17,3	17,76			0	16,255	16,72
0,032		20,99	21,64	0,00025	17,14		17,79	16,31	16,485						
0,6		20,255	21,58	0,005	17,225		17,32	16,37	16,47						
1,25		19,495	21,35	0,01	17,085		17,29	16,26	16,22						
1,5		18,8	21,37	0,025	17		17,11	16,295	16,41						
2		19,1	20,995	0,03	17,155		16,86	16,285	16,27						
2,3		18,865	20,925	0,035	16,86		17,745	16,24	16,35						
2,499		18,325	20,9	0,039	16,92		16,95	15,375	16,42						
MOY		19,64875	21,30375	21,5	MOY		17,085625	17,353125	18,8	MOY		16,17375	16,418125	18,2	
ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off		0	12,21	12,14			0	11,965	11,42			0	10,69	10,13	
	0,032	12,115	12,17	0,00025		12,08	12,11	10,18	10,1						
	0,6	12,27	12,17	0,005		12,14	12	11,1	11,135						
	1,25	11,98	12,03	0,01		11,93	11,91	11,155	11,14						
	1,5	11,995	11,73	0,025		11,965	11,87	11,23	11,14						
	2	11,94	11,93	0,03		11,77	11,91	11,15	11,08						
	2,3	11,74	11,78	0,035		11,625	11,65	11,135	11,04						
	2,499	11,47	11,5	0,039		11,46	11,485	11,17	11,12						
	MOY	11,965	11,93125	12,5		MOY	11,866875	11,794375	12		MOY	10,97625	10,860625	11,5	
	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125	12,125			0	12,085	11,89			0	11,605	11,9	
0,032		12,18	12,16	0,00025	12,07		12,12	10,725	10,64						
0,6		12,23	12,175	0,005	12,17		11,81	11,22	11,21						
1,25		12,02	11,96	0,01	11,96		11,94	11,27	11,28						
1,5		12,05	11,91	0,025	11,9		11,88	11,17	11,23						
2		11,895	11,83	0,03	11,85		11,485	11,11	11,19						
2,3		11,9	11,8	0,035	11,74		11,755	11,21	11,2						
2,499		11,51	11,56	0,039	11,37		11,43	11,17	11,17						
MOY		11,98875	11,94	12,5	MOY		11,893125	11,78875	12,5	MOY		11,185	11,2275	12	

**Tableau 4:** Calculs d'ENOB pour différentes configurations du convertisseur analogique numérique à 150°C

## TEMPERATURE : 175 °C

Configuration Frequence / Buffer	PGA = 1				PGA = 64				PGA = 128			
	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)
ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer off	0	19,45	20,82		0	16,66	17,15		0	15,63	15,71	
	0,032	19,64	19,84		0,005	16,76	17,085		0,00025	15,715	16,58	
	0,6	19,57	20,19		0,01	16,635	16,91		0,005	15,72	15,95	
	1,25	18,94	20,83		0,02	16,7	17,25		0,01	15,79	16,01	
	1,5	18,55	20,68		0,025	16,5	16,92		0,0125	15,67	16,37	
	2	18,22	20,595		0,03	16,7	17,335		0,015	15,71	15,405	
	2,3	18,47	20,49		0,035	16,68	17,275		0,0175	15,67	15,16	
	2,499	18,18	20,43		0,039	16,43	17,375		0,0195	15,67	16,385	
	MOY	18,8775	20,484375	21,6	MOY	16,633125	17,1625	20	MOY	15,696875	15,94625	19
ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,22	21,54		0	17,19	17,1		0	15,99	16,02	
	0,032	21,46	21,46		0,005	17,335	17,64		0,00025	16,12	16,23	
	0,6	20,13	21,17		0,01	17,27	17,38		0,005	16,085	15,67	
	1,25	19,325	21,06		0,02	17,11	17,29		0,01	16,2	16,35	
	1,5	18,98	21,18		0,025	17,14	17,06		0,0125	16,33	16,18	
	2	18,62	20,825		0,03	17,14	17,11		0,015	16,225	16,25	
	2,3	18,13	20,43		0,035	16,93	16,56		0,0175	16,12	16,54	
	2,499	17,85	20,37		0,039	17,025	17,03		0,0195	16,27	15,91	
	MOY	19,464375	21,004375	21,5	MOY	17,1425	17,14625	18,8	MOY	16,1675	16,14375	18,2
ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0	12,21	12,03		0	12	12,07		0	10,625	11,42	
	0,032	12,91	12,14		0,005	12,08	12,06		0,00025	10,22	10,1	
	0,6	12,22	12,21		0,01	12,01	11,38		0,005	11,2	11,175	
	1,25	12,1	12,08		0,02	11,915	11,84		0,01	11,2	11,11	
	1,5	11,94	11,99		0,025	11,945	11,87		0,0125	11,15	11,14	
	2	11,95	11,925		0,03	11,81	11,84		0,015	11,18	11,14	
	2,3	11,69	11,81		0,035	11,725	11,74		0,0175	11,09	11,11	
	2,499	11,46	11,47		0,039	11,38	11,375		0,0195	11,1	11,08	
	MOY	12,06	11,956875	12,5	MOY	11,858125	11,771875	12	MOY	10,970625	11,034375	11,5
ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,21	11,89		0	11,95	12,06		0	11,4	11,9	
	0,032	12,37	12,155		0,005	12,11	12,035		0,00025	10,75	10,64	
	0,6	12,19	12,165		0,01	12,11	12,14		0,005	11,19	11,12	
	1,25	12,1	11,97		0,02	11,955	11,93		0,01	11,22	11,14	
	1,5	12,02	12		0,025	11,78	11,81		0,0125	11,18	11,23	
	2	11,84	11,83		0,03	11,875	11,82		0,015	11,16	11,2	
	2,3	12,47	11,66		0,035	11,74	11,79		0,0175	11,18	11,185	
	2,499	11,48	11,49		0,039	11,515	11,475		0,0195	11,11	11,17	
	MOY	12,085	11,895	12,5	MOY	11,879375	11,8825	12,5	MOY	11,14875	11,198125	12

**Tableau 5:** Calculs d'ENOB pour différentes configurations du convertisseur analogique numérique à 175°C

## TEMPERATURE : 185 °C

Configuration Frequence / Buffer	PGA = 1				PGA = 64				PGA = 128																																																																																																																																																																																																																																								
	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)	Valeurs (V)	REF INTERNE 2.5V ENOB Mesuré(bits)	REF EXTERNE 2.5V ENOB Mesuré(bits)	ENOB Theorique(bits)																																																																																																																																																																																																																																					
ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer off	0	19,95	20,38		0	16,53	17,06		0	15,61	16,08																																																																																																																																																																																																																																						
	0,032	19,95	20,15		0,6	19,13	20,31		1,25	18,71	20,98		1,5	18,635	21,23	2	18,105	20,53	2,3	18,14	20,7	2,499	18,29	20,52	MOY	18,86375	20,6	21,6	MOY	16,524375	16,975625	20	MOY	15,62875	15,986875	19	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,12	21,605		0	17,08	17,45		0	16,225	17,02		0,032	21,13	21,27	0,6	20,11	21,02	1,25	18,14	20,79	1,5	18,5	21,13	2	18,47	20,715	2,3	17,7	20,32	2,499	18,01	20,53	MOY	19,1475	20,9225	21,5	MOY	16,8675	17,09875	18,8	MOY	16,030625	16,51125	18,2	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0	12,21	12,21		0	12,02	11,43		0	10,64	10,13		0,032	12,11	12,21	0,6	12,27	12,22	1,25	12,14	11,92	1,5	11,95	11,945	2	11,96	11,87	2,3	11,74	11,77	2,499	11,445	11,48	MOY	11,978125	11,953125	12,5	MOY	11,814375	11,416875	12	MOY	10,943125	10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125	12,125		0	11,95	11,89		0	11,61	11,9		0,032	12,16	12,19	0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																																			
	0,6	19,13	20,31		1,25	18,71	20,98		1,5	18,635	21,23		2	18,105	20,53	2,3	18,14	20,7	2,499	18,29	20,52	MOY	18,86375	20,6	21,6	MOY	16,524375	16,975625	20	MOY	15,62875	15,986875	19	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,12		21,605		0		17,08	17,45			0	16,225	17,02			0,032	21,13	21,27	0,6	20,11	21,02	1,25	18,14	20,79	1,5	18,5	21,13	2	18,47	20,715	2,3	17,7	20,32	2,499	18,01	20,53	MOY	19,1475	20,9225	21,5	MOY	16,8675	17,09875	18,8	MOY	16,030625	16,51125		18,2	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0		12,21	12,21			0	12,02	11,43			0	10,64	10,13		0,032	12,11	12,21	0,6	12,27	12,22	1,25	12,14	11,92	1,5	11,95	11,945	2	11,96	11,87	2,3	11,74	11,77	2,499	11,445	11,48	MOY	11,978125	11,953125	12,5	MOY	11,814375	11,416875		12	MOY	10,943125		10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on		0	12,125	12,125			0	11,95	11,89		0	11,61	11,9		0,032	12,16	12,19	0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																										
	1,25	18,71	20,98		1,5	18,635	21,23		2	18,105	20,53		2,3	18,14	20,7	2,499	18,29	20,52	MOY	18,86375	20,6	21,6	MOY	16,524375	16,975625	20	MOY	15,62875	15,986875	19	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,12		21,605			0		17,08		17,45				0	16,225	17,02				0,032	21,13	21,27	0,6	20,11	21,02	1,25	18,14	20,79	1,5	18,5	21,13	2	18,47	20,715	2,3	17,7	20,32	2,499	18,01	20,53	MOY	19,1475	20,9225	21,5	MOY	16,8675	17,09875	18,8	MOY	16,030625		16,51125		18,2		ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0			12,21	12,21				0	12,02	11,43			0	10,64	10,13		0,032	12,11	12,21	0,6	12,27	12,22	1,25	12,14	11,92	1,5	11,95	11,945	2	11,96	11,87	2,3	11,74	11,77	2,499	11,445	11,48	MOY	11,978125		11,953125	12,5	MOY		11,814375	11,416875			12	MOY	10,943125			10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on		0	12,125	12,125			0	11,95	11,89		0	11,61	11,9		0,032	12,16	12,19	0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																	
	1,5	18,635	21,23		2	18,105	20,53		2,3	18,14	20,7		2,499	18,29	20,52	MOY	18,86375	20,6	21,6	MOY	16,524375	16,975625	20	MOY	15,62875	15,986875	19	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,12		21,605			0			17,08		17,45						0	16,225	17,02					0,032	21,13	21,27	0,6	20,11	21,02	1,25	18,14	20,79	1,5	18,5	21,13	2	18,47	20,715	2,3	17,7	20,32	2,499	18,01	20,53	MOY	19,1475	20,9225	21,5	MOY	16,8675	17,09875	18,8	MOY		16,030625		16,51125			18,2			ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0				12,21	12,21				0	12,02	11,43			0	10,64	10,13		0,032	12,11	12,21	0,6	12,27	12,22	1,25	12,14	11,92	1,5	11,95	11,945	2	11,96	11,87	2,3	11,74	11,77		2,499	11,445	11,48		MOY	11,978125			11,953125	12,5	MOY			11,814375	11,416875			12	MOY	10,943125			10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on		0	12,125	12,125			0	11,95	11,89		0	11,61	11,9		0,032	12,16	12,19	0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																								
	2	18,105	20,53		2,3	18,14	20,7		2,499	18,29	20,52		MOY	18,86375	20,6	21,6	MOY	16,524375	16,975625	20	MOY	15,62875	15,986875	19	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,12		21,605			0			17,08			17,45								0	16,225	17,02						0,032	21,13	21,27	0,6	20,11	21,02	1,25	18,14	20,79	1,5	18,5	21,13	2	18,47	20,715	2,3	17,7	20,32	2,499	18,01	20,53	MOY	19,1475	20,9225	21,5	MOY	16,8675	17,09875	18,8		MOY		16,030625			16,51125				18,2				ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0				12,21	12,21				0	12,02	11,43			0	10,64	10,13		0,032	12,11	12,21	0,6	12,27	12,22	1,25	12,14	11,92	1,5	11,95	11,945	2		11,96	11,87	2,3		11,74	11,77			2,499	11,445	11,48			MOY	11,978125			11,953125	12,5	MOY			11,814375	11,416875			12	MOY	10,943125			10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on		0	12,125	12,125			0	11,95	11,89		0	11,61	11,9		0,032	12,16	12,19	0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																															
	2,3	18,14	20,7		2,499	18,29	20,52		MOY	18,86375	20,6		21,6	MOY	16,524375	16,975625	20	MOY	15,62875	15,986875	19	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,12		21,605			0			17,08			17,45											0	16,225	17,02							0,032	21,13	21,27	0,6	20,11	21,02	1,25	18,14	20,79	1,5	18,5	21,13	2	18,47	20,715	2,3	17,7	20,32	2,499	18,01	20,53	MOY	19,1475	20,9225	21,5	MOY	16,8675	17,09875		18,8		MOY			16,030625				16,51125					18,2				ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0				12,21	12,21				0	12,02	11,43			0	10,64	10,13		0,032	12,11	12,21	0,6	12,27	12,22	1,25	12,14		11,92	1,5	11,95		11,945	2			11,96	11,87	2,3			11,74	11,77			2,499	11,445	11,48			MOY	11,978125			11,953125	12,5	MOY			11,814375	11,416875			12	MOY	10,943125			10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on		0	12,125	12,125			0	11,95	11,89		0	11,61	11,9		0,032	12,16	12,19	0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																						
	2,499	18,29	20,52		MOY	18,86375	20,6		21,6	MOY	16,524375		16,975625	20	MOY	15,62875	15,986875	19	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,12		21,605			0			17,08			17,45														0	16,225	17,02								0,032	21,13	21,27	0,6	20,11	21,02	1,25	18,14	20,79	1,5	18,5	21,13	2	18,47	20,715	2,3	17,7	20,32	2,499	18,01	20,53	MOY	19,1475	20,9225	21,5	MOY	16,8675		17,09875		18,8			MOY				16,030625					16,51125					18,2				ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0				12,21	12,21				0	12,02	11,43			0	10,64	10,13		0,032	12,11	12,21		0,6	12,27	12,22		1,25	12,14			11,92	1,5	11,95			11,945	2			11,96	11,87	2,3			11,74	11,77			2,499	11,445	11,48			MOY	11,978125			11,953125	12,5	MOY			11,814375	11,416875			12	MOY	10,943125			10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on		0	12,125	12,125			0	11,95	11,89		0	11,61	11,9		0,032	12,16	12,19	0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12													
	MOY	18,86375	20,6		21,6	MOY	16,524375		16,975625	20	MOY		15,62875	15,986875	19	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,12		21,605			0			17,08			17,45																	0	16,225	17,02									0,032	21,13	21,27	0,6	20,11	21,02	1,25	18,14	20,79	1,5	18,5	21,13	2	18,47	20,715	2,3	17,7	20,32	2,499	18,01	20,53	MOY	19,1475	20,9225	21,5	MOY		16,8675		17,09875			18,8				MOY					16,030625					16,51125					18,2				ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0				12,21	12,21				0	12,02	11,43			0	10,64		10,13		0,032		12,11	12,21			0,6	12,27	12,22			1,25	12,14			11,92	1,5	11,95			11,945	2			11,96	11,87	2,3			11,74	11,77			2,499	11,445	11,48			MOY	11,978125			11,953125	12,5	MOY			11,814375	11,416875			12	MOY	10,943125			10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on		0	12,125	12,125			0	11,95	11,89		0	11,61	11,9		0,032	12,16	12,19	0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12				
	MOY	16,524375	16,975625		20	MOY	15,62875		15,986875	19	ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on		0	21,12	21,605			0		17,08			17,45						0																	16,225	17,02										0,032	21,13	21,27	0,6	20,11	21,02	1,25	18,14	20,79	1,5	18,5	21,13	2	18,47	20,715	2,3	17,7	20,32	2,499	18,01	20,53	MOY	19,1475	20,9225	21,5	MOY		16,8675		17,09875			18,8				MOY					16,030625					16,51125					18,2					ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off				0	12,21				12,21		0			12,02	11,43				0		10,64	10,13				0,032	12,11			12,21	0,6			12,27	12,22	1,25			12,14	11,92			1,5	11,95	11,945			2	11,96			11,87	2,3	11,74			11,77	2,499			11,445	11,48	MOY			11,978125	11,953125			12,5	MOY	11,814375			11,416875	12			MOY	10,943125	10,45875			11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0		12,125	12,125			0	11,95	11,89		0	11,61	11,9		0,032	12,16	12,19	0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5
MOY	15,62875	15,986875	19																																																																																																																																																																																																																																														
ACLK = 10 et DECIMATION = 2047  FDATA = 1.28 Hz et Buffer on	0	21,12	21,605		0	17,08	17,45		0	16,225		17,02																																																																																																																																																																																																																																					
	0,032	21,13	21,27		0,6	20,11	21,02		1,25	18,14		20,79		1,5	18,5			21,13		2			18,47						20,715					2,3			17,7		20,32		2,499			18,01	20,53	MOY	19,1475		20,9225	21,5							MOY	16,8675	17,09875	18,8	MOY	16,030625	16,51125	18,2	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0	12,21	12,21		0	12,02	11,43		0	10,64	10,13		0,032	12,11	12,21	0,6	12,27	12,22	1,25	12,14	11,92	1,5		11,95	11,945	2		11,96		11,87	2,3		11,74		11,77			2,499					11,445									11,48	MOY				11,978125		11,953125			12,5	MOY	11,814375			11,416875	12	MOY	10,943125	10,45875	11,5		ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125	12,125		0		11,95	11,89		0	11,61		11,9				0,032	12,16	12,19			0,6	12,24			12,22	1,25	12,08			12,06	1,5			11,96	11,99	2			11,85	11,86			2,3	11,83	11,73			2,499	11,72			11,57	MOY	11,995625			11,968125		12,5		MOY	11,854375			11,915	12,5	MOY		11,15125	11,344375	12																														
	0,6	20,11	21,02		1,25	18,14	20,79		1,5	18,5		21,13		2	18,47			20,715		2,3			17,7						20,32		2,499			18,01		20,53	MOY		19,1475		20,9225		21,5	MOY	16,8675	17,09875	18,8		MOY	16,030625	16,51125						18,2	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0	12,21	12,21		0	12,02		11,43		0		10,64	10,13			0,032	12,11	12,21		0,6	12,27	12,22	1,25	12,14	11,92	1,5	11,95	11,945	2	11,96	11,87	2,3	11,74		11,77	2,499	11,445	11,48		MOY		11,978125	11,953125		12,5		MOY			11,814375									11,416875	12				MOY		10,943125			10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on			0	12,125	12,125		0	11,95			11,89		0		11,61	11,9		0,032		12,16	12,19	0,6	12,24			12,22	1,25	12,08	12,06	1,5		11,96	11,99			2	11,85	11,86			2,3	11,83			11,73	2,499	11,72			11,57	MOY			11,995625	11,968125	12,5			MOY	11,854375			11,915	12,5	MOY			11,15125		11,344375		12																																								
	1,25	18,14	20,79		1,5	18,5	21,13		2	18,47		20,715		2,3	17,7			20,32		2,499			18,01					20,53	MOY		19,1475		20,9225	21,5		MOY	16,8675		17,09875		18,8	MOY	16,030625	16,51125	18,2	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0		12,21	12,21		0					12,02		11,43		0		10,64	10,13				0,032		12,11	12,21			0,6	12,27	12,22		1,25	12,14	11,92	1,5	11,95	11,945	2	11,96	11,87	2,3	11,74	11,77	2,499	11,445	11,48	MOY	11,978125	11,953125	12,5		MOY	11,814375	11,416875	12		MOY		10,943125	10,45875		11,5		ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on							0	12,125				12,125					0	11,95				11,89		0		11,61	11,9					0,032		12,16	12,19		0,6		12,24	12,22	1,25	12,08		12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83		11,73	2,499	11,72	11,57	MOY		11,995625	11,968125			12,5	MOY	11,854375			11,915	12,5			MOY	11,15125	11,344375			12																																																					
	1,5	18,5	21,13		2	18,47	20,715		2,3	17,7		20,32		2,499	18,01			20,53		MOY			19,1475		20,9225			21,5	MOY	16,8675	17,09875		18,8	MOY		16,030625	16,51125		18,2	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0	12,21	12,21		0		12,02		11,43			0	10,64				10,13				0,032		12,11	12,21				0,6		12,27	12,22			1,25	12,14	11,92		1,5	11,95	11,945	2	11,96	11,87	2,3	11,74	11,77	2,499	11,445	11,48	MOY	11,978125	11,953125	12,5	MOY	11,814375	11,416875	12	MOY	10,943125	10,45875	11,5		ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125	12,125					0				11,95		11,89					0					11,61	11,9						0,032		12,16	12,19					0,6		12,24	12,22		1,25		12,08	12,06	1,5	11,96		11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5		MOY	11,15125	11,344375	12																																																																	
	2	18,47	20,715		2,3	17,7	20,32		2,499	18,01		20,53		MOY	19,1475			20,9225		21,5		MOY	16,8675		17,09875		18,8	MOY	16,030625	16,51125	18,2		ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0		12,21	12,21		0		12,02	11,43			0		10,64		10,13				0,032	12,11			12,21				0,6		12,27	12,22				1,25		12,14	11,92			1,5	11,95	11,945		2	11,96	11,87	2,3	11,74	11,77	2,499	11,445	11,48	MOY	11,978125	11,953125	12,5	MOY	11,814375	11,416875	12	MOY	10,943125	10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125	12,125			0	11,95			11,89						0	11,61	11,9					0,032					12,16	12,19						0,6		12,24	12,22					1,25		12,08	12,06		1,5		11,96	11,99	2	11,85		11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																																					
	2,3	17,7	20,32		2,499	18,01	20,53		MOY	19,1475		20,9225		21,5	MOY			16,8675	17,09875	18,8		MOY	16,030625	16,51125	18,2		ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0	12,21	12,21				0	12,02	11,43			0		10,64	10,13					0,032		12,11				12,21	0,6	12,27		12,22				1,25		12,14	11,92				1,5		11,95	11,945			2	11,96	11,87		2,3	11,74	11,77	2,499	11,445	11,48	MOY	11,978125	11,953125	12,5	MOY	11,814375	11,416875	12	MOY	10,943125	10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125		12,125		0			11,95	11,89			0					11,61	11,9		0,032				12,16	12,19			0,6		12,24	12,22						1,25		12,08	12,06					1,5		11,96	11,99		2		11,85	11,86	2,3	11,83		11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																																								
	2,499	18,01	20,53		MOY	19,1475	20,9225		21,5	MOY		16,8675		17,09875	18,8	MOY		16,030625	16,51125	18,2	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0	12,21	12,21				0	12,02	11,43				0	10,64	10,13					0,032	12,11					12,21		0,6				12,27	12,22	1,25	12,14	11,92				1,5		11,95	11,945				2		11,96	11,87			2,3	11,74	11,77		2,499	11,445	11,48	MOY	11,978125	11,953125	12,5	MOY	11,814375	11,416875	12	MOY	10,943125	10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125		12,125			0		11,95			11,89				0			11,61		11,9			0,032		12,16		12,19	0,6			12,24	12,22	1,25	12,08			12,06			1,5		11,96	11,99					2		11,85	11,86		2,3		11,83	11,73	2,499	11,72		11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																																											
	MOY	19,1475	20,9225		21,5	MOY	16,8675		17,09875	18,8	MOY	16,030625		16,51125	18,2	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0	12,21	12,21			0	12,02	11,43				0	10,64	10,13					0,032	12,11					12,21	0,6					12,27	12,22	1,25				12,14	11,92	1,5	11,95	11,945				2		11,96	11,87				2,3		11,74	11,77			2,499	11,445	11,48		MOY	11,978125	11,953125	12,5	MOY	11,814375	11,416875	12	MOY	10,943125	10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125		12,125			0			11,95		11,89							0			11,61	11,9				0,032		12,16		12,19	0,6	12,24		12,22	1,25	12,08	12,06		1,5	11,96			11,99		2	11,85	11,86				2,3		11,83	11,73		2,499		11,72	11,57	MOY	11,995625		11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																																														
	MOY	16,8675	17,09875		18,8	MOY	16,030625		16,51125	18,2	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0		12,21	12,21			0	12,02			11,43		0				10,64	10,13						0,032	12,11					12,21	0,6					12,27	12,22	1,25				12,14	11,92	1,5	11,95	11,945				2		11,96	11,87				2,3		11,74	11,77			2,499	11,445	11,48		MOY	11,978125	11,953125	12,5	MOY	11,814375	11,416875	12	MOY	10,943125	10,45875	11,5		ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0		12,125			12,125					0							11,95			11,89					0		11,61		11,9		0,032		12,16	12,19	0,6	12,24		12,22	1,25			12,08		12,06	1,5	11,96				11,99		2	11,85		11,86		2,3	11,83	11,73	2,499		11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																																										
MOY	16,030625	16,51125	18,2																																																																																																																																																																																																																																														
ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer off	0	12,21	12,21		0	12,02	11,43		0	10,64		10,13																																																																																																																																																																																																																																					
	0,032	12,11	12,21		0,6	12,27	12,22		1,25	12,14		11,92		1,5	11,95			11,945	2			11,96		11,87				2,3	11,74						11,77	2,499					11,445	11,48					MOY	11,978125	11,953125				12,5	MOY	11,814375	11,416875	12	MOY			10,943125	10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0		12,125	12,125		0	11,95	11,89		0	11,61	11,9		0,032	12,16	12,19	0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99			2		11,85			11,86					2,3							11,83			11,73					2,499		11,72		11,57		MOY		11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5			MOY	11,15125	11,344375	12																																																																																																								
	0,6	12,27	12,22		1,25	12,14	11,92		1,5	11,95		11,945		2	11,96			11,87	2,3			11,74		11,77				2,499	11,445						11,48	MOY					11,978125	11,953125				12,5	MOY	11,814375	11,416875		12		MOY	10,943125	10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0		12,125	12,125		0		11,95	11,89		0		11,61	11,9			0,032	12,16	12,19		0,6	12,24	12,22	1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86			2,3		11,83			11,73					2,499							11,72	11,57		MOY					11,995625		11,968125		12,5		MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																																																																															
	1,25	12,14	11,92		1,5	11,95	11,945		2	11,96		11,87		2,3	11,74			11,77	2,499			11,445		11,48				MOY	11,978125						11,953125	12,5				MOY	11,814375	11,416875		12		MOY	10,943125	10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0		12,125	12,125		0		11,95	11,89		0		11,61		11,9			0,032		12,16	12,19			0,6	12,24	12,22		1,25	12,08	12,06	1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73			2,499		11,72			11,57					MOY	11,995625					11,968125	12,5	MOY		11,854375					11,915	12,5	MOY		11,15125		11,344375	12																																																																																																																					
	1,5	11,95	11,945		2	11,96	11,87		2,3	11,74		11,77		2,499	11,445			11,48	MOY			11,978125		11,953125				12,5	MOY				11,814375		11,416875	12		MOY		10,943125	10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0		12,125	12,125		0		11,95	11,89		0		11,61		11,9			0,032		12,16		12,19			0,6		12,24	12,22			1,25	12,08	12,06		1,5	11,96	11,99	2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57			MOY		11,995625			11,968125		12,5			MOY	11,854375	11,915				12,5	MOY	11,15125	11,344375	12																																																																																																																																	
	2	11,96	11,87		2,3	11,74	11,77		2,499	11,445		11,48		MOY	11,978125			11,953125	12,5			MOY		11,814375			11,416875	12	MOY		10,943125		10,45875		11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125		12,125		0		11,95	11,89		0		11,61		11,9			0,032		12,16		12,19			0,6		12,24		12,22			1,25		12,08	12,06			1,5	11,96	11,99		2	11,85	11,86	2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125			12,5		MOY		11,854375	11,915		12,5		MOY	11,15125	11,344375	12																																																																																																																																									
	2,3	11,74	11,77		2,499	11,445	11,48		MOY	11,978125		11,953125		12,5	MOY			11,814375	11,416875		12	MOY		10,943125	10,45875		11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0		12,125	12,125			0		11,95	11,89		0		11,61		11,9			0,032		12,16		12,19			0,6		12,24		12,22			1,25		12,08		12,06			1,5		11,96	11,99			2	11,85	11,86		2,3	11,83	11,73	2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375			11,915	12,5	MOY		11,15125	11,344375	12																																																																																																																																															
	2,499	11,445	11,48		MOY	11,978125	11,953125		12,5	MOY		11,814375		11,416875	12	MOY		10,943125	10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0		12,125	12,125		0		11,95		11,89			0	11,61		11,9			0,032		12,16		12,19			0,6		12,24		12,22			1,25		12,08		12,06			1,5		11,96		11,99			2		11,85	11,86			2,3	11,83	11,73		2,499	11,72	11,57	MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125		11,344375	12																																																																																																																																																				
	MOY	11,978125	11,953125		12,5	MOY	11,814375		11,416875	12	MOY	10,943125		10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125	12,125			0	11,95	11,89			0		11,61	11,9				0,032	12,16		12,19			0,6		12,24		12,22			1,25		12,08		12,06			1,5		11,96		11,99			2		11,85		11,86			2,3		11,83	11,73			2,499	11,72	11,57		MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																																																																																																																								
	MOY	11,814375	11,416875		12	MOY	10,943125		10,45875	11,5	ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0		12,125	12,125			0	11,95			11,89		0			11,61		11,9					0,032	12,16		12,19			0,6		12,24		12,22			1,25		12,08		12,06			1,5		11,96		11,99			2		11,85		11,86			2,3		11,83	11,73			2,499	11,72	11,57		MOY	11,995625	11,968125	12,5	MOY	11,854375	11,915	12,5	MOY	11,15125	11,344375	12																																																																																																																																																								
MOY	10,943125	10,45875	11,5																																																																																																																																																																																																																																														
ACLK = 10 et DECIMATION = 26  FDATA = 101 Hz et Buffer on	0	12,125	12,125		0	11,95	11,89		0	11,61		11,9																																																																																																																																																																																																																																					
	0,032	12,16	12,19		0,6	12,24	12,22		1,25	12,08		12,06		1,5	11,96			11,99	2			11,85		11,86			2,3		11,83					11,73	2,499		11,72			11,57		MOY		11,995625			11,968125		12,5		MOY			11,854375		11,915	12,5	MOY			11,15125	11,344375	12																																																																																																																																																																																		
	0,6	12,24	12,22		1,25	12,08	12,06		1,5	11,96		11,99		2	11,85			11,86	2,3			11,83		11,73			2,499		11,72					11,57	MOY		11,995625			11,968125		12,5		MOY			11,854375		11,915	12,5	MOY			11,15125	11,344375	12																																																																																																																																																																																									
	1,25	12,08	12,06		1,5	11,96	11,99		2	11,85		11,86		2,3	11,83			11,73	2,499			11,72		11,57			MOY		11,995625					11,968125	12,5		MOY			11,854375		11,915	12,5	MOY			11,15125	11,344375	12																																																																																																																																																																																																
	1,5	11,96	11,99		2	11,85	11,86		2,3	11,83		11,73		2,499	11,72			11,57	MOY			11,995625		11,968125			12,5		MOY					11,854375	11,915	12,5	MOY			11,15125	11,344375	12																																																																																																																																																																																																							
	2	11,85	11,86		2,3	11,83	11,73		2,499	11,72		11,57		MOY	11,995625			11,968125	12,5			MOY		11,854375			11,915	12,5	MOY				11,15125	11,344375	12																																																																																																																																																																																																														
	2,3	11,83	11,73		2,499	11,72	11,57		MOY	11,995625		11,968125		12,5	MOY			11,854375	11,915		12,5	MOY		11,15125		11,344375	12																																																																																																																																																																																																																						
	2,499	11,72	11,57		MOY	11,995625	11,968125		12,5	MOY		11,854375		11,915	12,5	MOY		11,15125	11,344375	12																																																																																																																																																																																																																													
	MOY	11,995625	11,968125		12,5	MOY	11,854375		11,915	12,5	MOY	11,15125		11,344375	12																																																																																																																																																																																																																																		
	MOY	11,854375	11,915		12,5	MOY	11,15125		11,344375	12																																																																																																																																																																																																																																							
MOY	11,15125	11,344375	12																																																																																																																																																																																																																																														

*Tableau 6: calculs d'ENOB pour différentes configurations du convertisseur analogique numérique à 185°C*

## SYNTHESE DES MESURES SELON DIFFERENTES TEMPERATURES

	Configuration												
	Temperatures												
	FDATA = 1.28 Hz et Buffer off			FDATA = 1.28 Hz et Buffer on			FDATA = 101 Hz et Buffer off			FDATA = 101 Hz et Buffer on			
	REF INTERNE 2.5V	REF EXTERNE 2.5V		REF INTERNE 2.5V	REF EXTERNE 2.5V		REF INTERNE 2.5V	REF EXTERNE 2.5V		REF INTERNE 2.5V	REF EXTERNE 2.5V		
	ENOB Mesuré(bits)	ENOB Mesuré(bits)	ENOB Theorique(bits)	ENOB Mesuré(bits)	ENOB Mesuré(bits)	ENOB Theorique(bits)	ENOB Mesuré(bits)	ENOB Mesuré(bits)	ENOB Theorique(bits)	ENOB Mesuré(bits)	ENOB Mesuré(bits)	ENOB Theorique(bits)	
PGA=1	25	18,638125	20,431875	21,6	19,745625	21,179375	21,5	11,968125	11,993125	12,5	11,951875	11,953125	12,5
	125	19,2	20,71375	21,6	19,785	21,2075	21,5	12,0125	12,0025	12,5	11,991875	11,96375	12,5
	150	18,825625	20,378125	21,6	19,64875	21,30375	21,5	11,965	11,93125	12,5	11,98875	11,94	12,5
	175	18,8775	20,484375	21,6	19,464375	21,004375	21,5	12,06	11,956875	12,5	12,085	11,895	12,5
	185	18,86375	20,6	21,6	19,1475	20,9225	21,5	11,978125	11,953125	12,5	11,995625	11,968125	12,5
PGA=64	25	15,933125	17,01625	20	16,703125	17,2325	18,8	11,608125	11,3375	12	11,824375	11,933125	12,5
	125	16,739375	17,2475	20	17,250625	17,40625	18,8	11,795	11,776375	12	11,8875	11,9	12,5
	150	16,643125	17,126875	20	17,085625	17,353125	18,8	11,866875	11,794375	12	11,893125	11,3375	12,5
	175	16,633125	17,1625	20	17,1425	17,14625	18,8	11,858125	11,771875	12	11,879375	11,8825	12,5
	185	16,524375	16,975625	20	16,8675	17,09875	18,8	11,814375	11,416875	12	11,854375	11,915	12,5
PGA=128	25	14,899375	16,000625	19	15,766875	16,395625	18,2	10,583125	10,39875	11,5	11,16625	11,24625	12
	125	15,63875	16,23625	19	16,301875	16,43125	18,2	10,955625	10,53375	11,5	11,181875	11,21125	12
	150	15,70625	16,33375	19	16,17375	16,418125	18,2	10,97625	10,860625	11,5	11,185	11,2275	12
	175	15,696875	15,94625	19	16,1675	16,14375	18,2	10,970625	11,034375	11,5	11,14875	11,198125	12
	185	15,62875	15,986875	19	16,030625	16,51125	18,2	10,943125	10,45875	11,5	11,15125	11,344375	12

*Tableau 7: Synthèse de calculs d'ENOB pour différentes mesures du convertisseur analogique numérique en fonction de la température.*

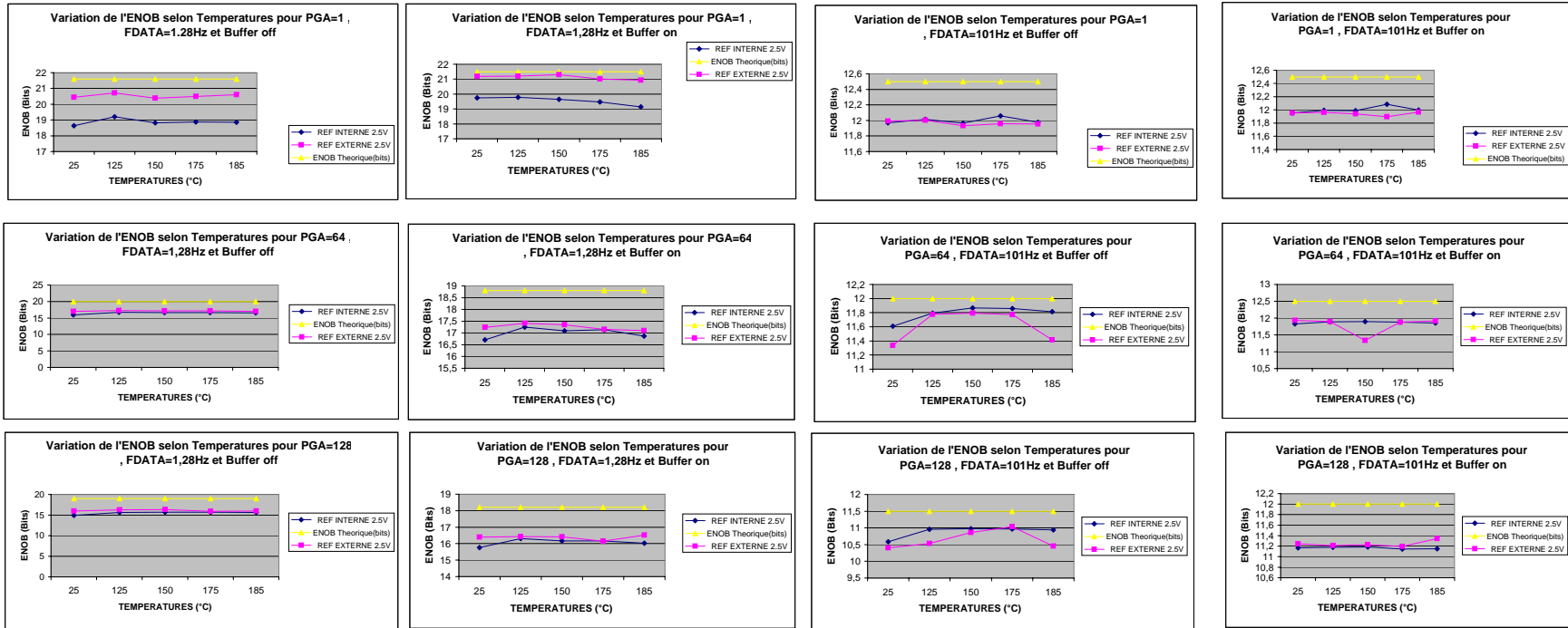


Figure 1 : Influence de la température sur le convertisseur analogique numérique à différentes configurations.

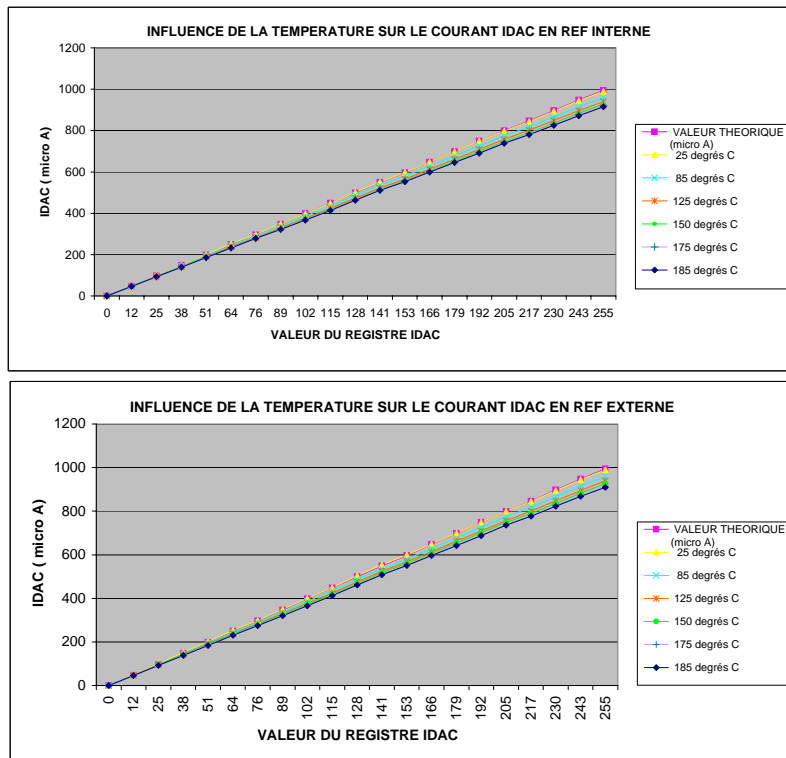
## SYNTHESE DES MESURES DU COURANT DE SORTIE IDAC

VALEURS MESURÉES (micro A)

Valeur désirée(micro A)	REGISTRE IDAC (HEX)	REGISTRE IDAC (DEC)	VALEUR THEORIQUE (micro A)	REF INTERNE 2.5V					
				25 degrés C	85 degrés C	125 degrés C	150 degrés C	175 degrés C	185 degrés C
MIN	0	0	0	0,6	0,5	0,2	0	0	0,6
50	C	12	46,9	46,7	47,6	45,3	44,8	46	46,4
100	19	25	97,5	98,9	97,4	93,7	92,5	93,1	93,3
150	26	38	148,2	148,5	146,3	140,8	138,8	139,4	139,2
200	33	51	198,9	199,6	195,5	188,8	186	186	185,5
250	40	64	249,6	252	244,6	237,7	234,2	232,4	231,5
300	4C	76	296,4	296,8	289,6	281,6	278	276,7	279,3
350	59	89	347,1	347,7	338,4	329,5	325	323	321,9
400	66	102	397,8	397,8	386,6	376,3	371,1	369	367,6
450	73	115	448,5	448,2	435,1	424,1	418,1	415,1	413,8
500	80	128	499,2	501,4	487,4	474	467,3	464,7	463,1
550	8D	141	549,9	550,5	536,1	521,6	514,5	512,6	511
600	99	153	596,7	597,2	581	565,7	558,2	555,2	553,4
650	A6	166	647,4	646,7	629	612,6	604,2	601,1	598,8
700	B3	179	698,1	697,6	677,5	660,5	651,4	647,3	645,1
750	C0	192	748,8	748,7	725,5	709,3	699,5	693,8	690,6
800	CD	205	799,5	797,3	774,1	756,9	746,9	742,4	738,5
850	D9	217	846,3	844,5	818,9	801,3	790,2	784,8	780,5
900	E6	230	897	893,6	866,3	848,1	836,5	830,6	826
950	F3	243	947,7	944,2	914,1	896	883,6	876,6	871,2
MAX	FF	255	994,5	989,2	959,2	939,9	927,4	920,5	914,7

Valeur désirée(micro A)	REGISTRE IDAC (HEX)	REGISTRE IDAC (DEC)	VALEUR THEORIQUE (micro A)	REF EXTERNE 2.5V					
				25 degrés C	85 degrés C	125 degrés C	150 degrés C	175 degrés C	185 degrés C
MIN	0	0	0	0	0	0	0	0	0
50	C	12	46,9	48,4	47,2	46,5	46,1	45,9	45,8
100	19	25	97,5	99,9	97	95	93,9	93	92,4
150	26	38	148,2	150,6	146	142,7	140,7	139	138,2
200	33	51	198,9	201,4	195,2	190,6	187,8	185,4	184,3
250	40	64	249,6	252,4	244,4	238,5	234,8	231,8	230,3
300	4C	76	296,4	298,7	289,4	283,1	279,2	275,7	274,2
350	59	89	347,1	349,1	338,2	330,6	326	322	320,3
400	66	102	397,8	398,8	386,5	377,6	372,3	367	365,8
450	73	115	448,5	449	435,1	425	419,1	413,6	412,6
500	80	128	499,2	503,8	487,6	475,9	469,1	463,1	460,8
550	8D	141	549,9	553,2	535,9	523,9	517,1	510,3	508,2
600	99	153	596,7	598,6	580,9	567,7	560,1	552,8	550,6
650	A6	166	647,4	649,2	628,8	614,6	606,4	598,4	595,6
700	B3	179	698,1	698,9	677,2	661,8	653,2	644,3	641,9
750	C0	192	748,8	749,2	725,7	709,2	699,9	690,6	687,3
800	CD	205	799,5	798,5	774	757,1	747,5	737,5	735,4
850	D9	217	846,3	844,5	818,5	801	790,6	779,8	777,2
900	E6	230	897	893,9	866,5	847,6	836,4	825,1	822,1
950	F3	243	947,7	943,8	914,8	895,2	883,2	870,3	867,6
MAX	FF	255	994,5	989,4	959,4	939,4	927,2	913,6	909,1

**Tableau 8 : Influence de la Temperature sur le courant de sortie IDAC**



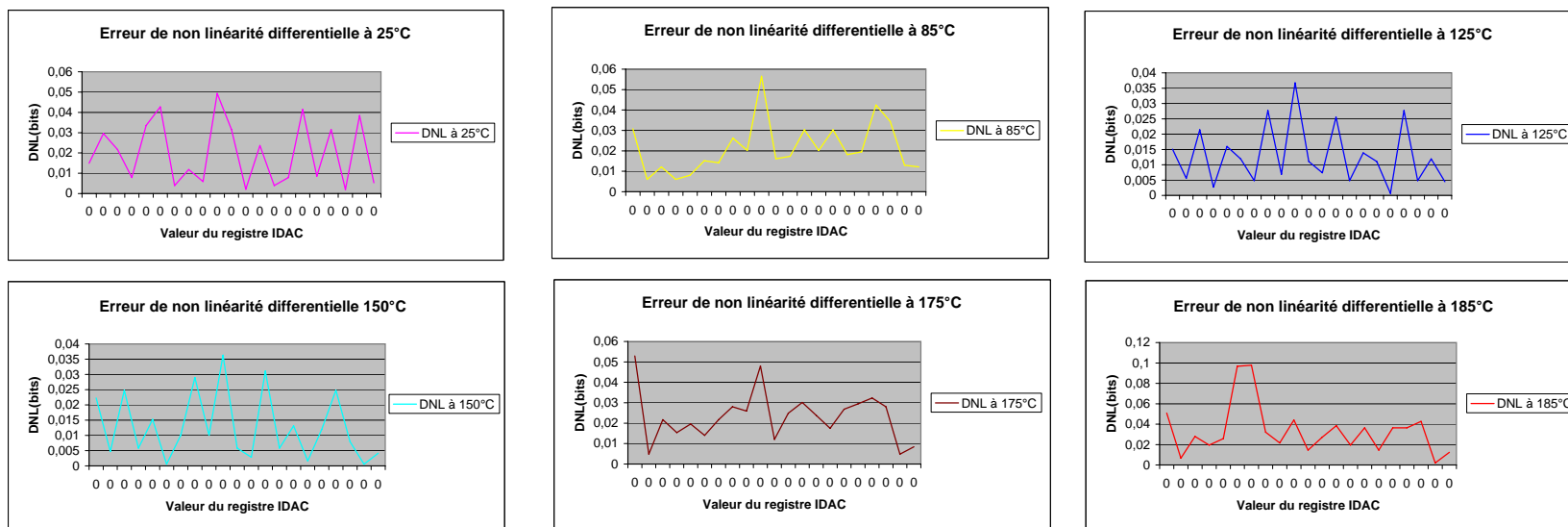
**Figure 2 : Graphes representants le comportement de la source digitale en haute temperature .**



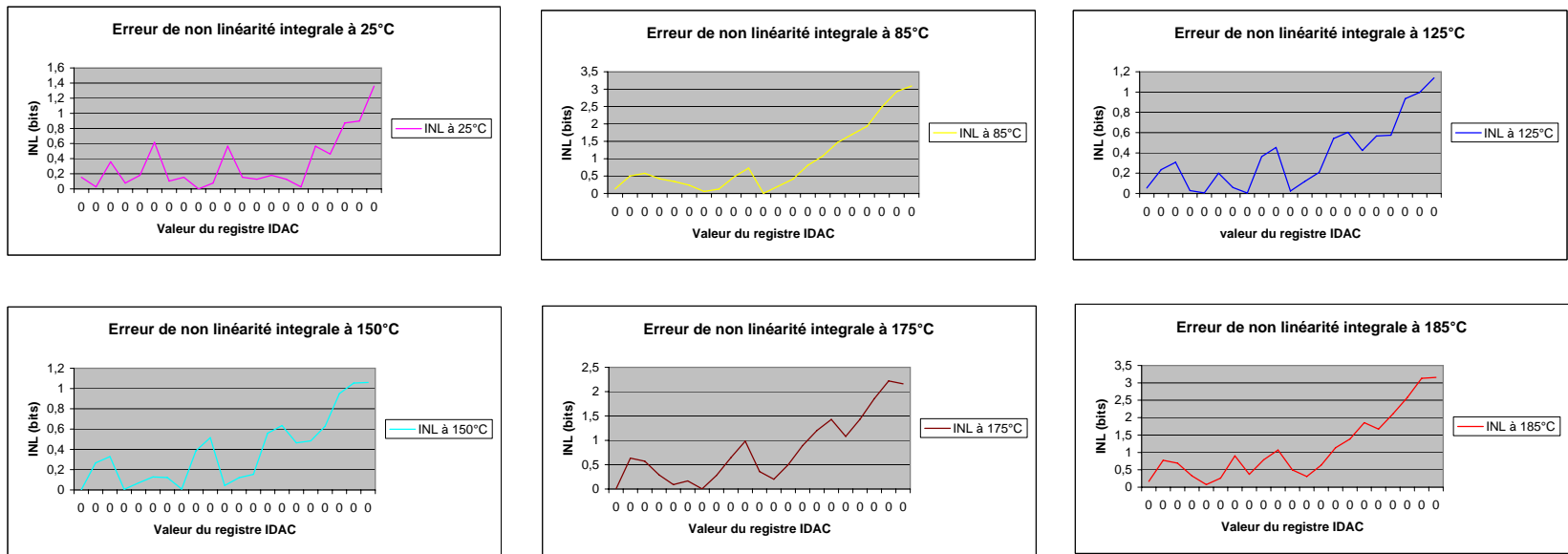
**DNL ET INL DE LA SOURCE DE COURANT DIGITALE A DIFFERENTES TEMPERATURES**

REGISTRE IDAC (HEX)	REGISTRE IDAC (DEC)	25°C		85°C		125°C		150°C		175°C		185°C						
		DNL	INL	Valeur LSB	DNL	INL	Valeur LSB	DNL	INL	Valeur LSB	DNL	INL	Valeur LSB	DNL	INL			
0	#NOM?	0,01495727	0,15384615		0,03080578	0,13131284		0,01509971	0,05401861		0,02225737	0	0,05292029	0	0,0508421	0,16519789		
C	#NOM?	0,0295858	0,02564103	3,96666667	0,00605836	0,50098215	3,775	0,0055772	0,23521518	3,73333333	0,00470488	0,26708847	3,83333333	0,00483186	0,63504348	3,86666667	0,00669477	0,77530312
19	#NOM?	0,02169625	0,35897436	3,896	0,01212342	0,5797408	3,748	0,0214321	0,30771881	3,7	0,02478332	0,32825167	3,724	0,02173493	0,5722293	3,732	0,02787398	0,68827114
26	#NOM?	0,00788955	0,07692308	3,85	0,00606283	0,42213633	3,70526316	0,00273335	0,02910148	3,65263158	0,00582662	0,00606874	3,66842105	0,01539628	0,28967524	3,6631579	0,0194023	0,32590935
33	#NOM?	0,03353057	0,17948718	3,83333333	0,00808303	0,34331956	3,70196078	0,0159654	0,00643211	3,64705882	0,01523637	0,06967733	3,64705882	0,01962204	0,08952363	3,6372549	0,02575606	0,07367948
40	#NOM?	0,04273504	0,61538462	3,821875	0,01515372	0,23824023	3,7140625	0,01190959	0,20111804	3,659375	0,00056087	0,12839553	3,63125	0,01400802	0,16556295	3,6171875	0,0967304	0,26114933
4C	#NOM?	0,00394477	0,1025641	3,81052632	0,01414362	0,05639563	3,70526316	0,00481099	0,05820295	3,65789474	0,01003922	0,12166506	3,64078947	0,02173493	0,00253327	3,675	0,0977654	0,89961552
59	#NOM?	0,01183432	0,15384615	3,80224719	0,0262648	0,12747141	3,70224719	0,02766502	0,00433994	3,65168539	0,02899592	0,0088448	3,62921348	0,02807358	0,28002079	3,61685393	0,03210983	0,37133464
66	#NOM?	0,00591716	0	3,79019608	0,02020421	0,46891385	3,68921569	0,00688863	0,36398519	3,63823529	0,01003922	0,38579171	3,61764706	0,0259607	0,64497731	3,60392157	0,02152022	0,78876239
73	#NOM?	0,04930967	0,07692308	3,78347826	0,0565633	0,73156859	3,68782609	0,03674179	0,45353739	3,63565217	0,03629937	0,51630157	3,60956522	0,04799023	0,98246635	3,59826087	0,04413535	1,06852524
80	#NOM?	0,03155819	0,56410256	3,8078125	0,01616382	0,00375425	3,703125	0,01104391	0,02410582	3,65078125	0,00582662	0,04440977	3,63046875	0,01207121	0,35859338	3,61796875	0,01448445	0,49476568
8D	#NOM?	0,00213675	0,15384615	3,80212766	0,01734226	0,20637536	3,69929078	0,00740804	0,11946499	3,64893617	0,0028427	0,12015585	3,63546099	0,02490425	0,20166767	3,62411348	0,02716801	0,30646785
99	#NOM?	0,02366864	0,12820513	3,79738562	0,0303052	0,41448253	3,69738562	0,02558738	0,20836148	3,64836601	0,03110222	0,15426821	3,62875817	0,03018646	0,50051871	3,61699346	0,03846359	0,63248397
A6	#NOM?	0,00394477	0,17948718	3,78915663	0,02020421	0,8084501	3,69036145	0,00481099	0,54099742	3,63975904	0,00582662	0,55859701	3,62108434	0,02384781	0,89294272	3,60722892	0,0194023	1,13251066
B3	#NOM?	0,00788955	0,12820513	3,7849162	0,0303052	1,07110484	3,68994413	0,01388776	0,60354031	3,63910615	0,01313008	0,63434309	3,61620112	0,01750916	1,20296427	3,60391062	0,03634567	1,38474052
C0	#NOM?	0,04142012	0,02564103	3,77864583	0,01818401	1,46507242	3,69427083	0,01104391	0,42299946	3,64322917	0,00161402	0,46365212	3,61354167	0,02686139	1,43058336	3,596875	0,01448445	1,85723424
CD	#NOM?	0,00854701	0,56410256	3,77609756	0,01953081	1,70146459	3,69219512	0,00065571	0,56657027	3,64341463	0,01197	0,4846344	3,62146342	0,02948217	1,08138525	3,60243902	0,03634567	1,66893641
D9	#NOM?	0,03155819	0,46153846	3,77373272	0,04242638	1,93583432	3,69262673	0,02766502	0,57443884	3,64147465	0,02478332	0,62827434	3,61658986	0,03229935	1,43517126	3,59677419	0,03634567	2,10508445
E6	#NOM?	0,00197239	0,87179487	3,76652174	0,03434559	2,4873773	3,6873913	0,00481099	0,93408408	3,63695652	0,00793292	0,95045746	3,61130435	0,02807358	1,85506275	3,59130435	0,04269944	2,57757816
F3	#NOM?	0,03846154	0,8974359	3,7617284	0,01296517	2,93387002	3,6872428	0,01190959	0,99662698	3,63621399	0,00056087	1,05358543	3,60740741	0,00485219	2,22001928	3,58518519	0,00192944	3,13267082
FF	#NOM?	0,00532931	1,35897436	3,76156863	0,0121155	3,08945205	3,68588235	0,00446879	1,13954207	3,63686275	0,0041581	1,0603159	3,60980392	0,00847762	2,161793	3,58705882	0,01237578	3,15582415
MOYENNE LSB				3,80770082			3,70242773			3,65204833			3,64066812			3,63200776		

**Tableau 9 : Calcul des DNL et INL de l'IDAC**



**Figure 3 : Variation du DNL en fonction de la température**



*Figure 4 : Variation de l'INL en fonction de la température*

# ANNEXE D

*Department of Electrical & Systems Engineering*

*Departmental Papers (ESE)*

---

*University of Pennsylvania*

*Year 1996*

---

An Overview of Sigma-Delta Converters:  
How a 1-bit ADC achieves more than  
16-bit resolution

Pervez M. Aziz \*      Henrik V. Sorensen †  
Jan Van der Spiegel ‡

\* AT&T Bell Laboratories,

† Ariel Corporation,

‡ University of Pennsylvania, jan@seas.upenn.edu

Copyright 1996 IEEE. Reprinted from *IEEE Signal Processing Magazine*, Volume 13, Issue 1, September 1996, pages 61-84.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Pennsylvania's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

This paper is posted at ScholarlyCommons@Penn.

<http://repository.upenn.edu/ese-papers/136>

# An Overview of Sigma-Delta Converters



©The Image Bank/Gary S. Chapman

**How a 1-bit ADC achieves more than 16-bit resolution**

---

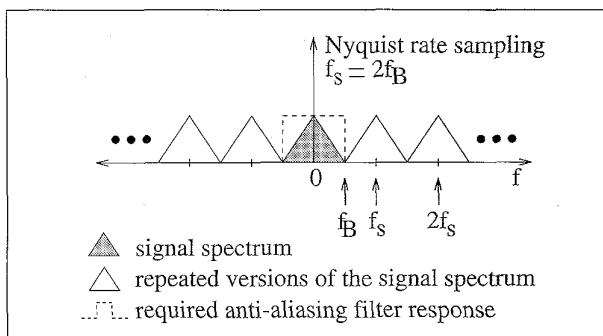
PERVEZ M. AZIZ,  
HENRIK V. SORENSEN, and  
JAN VAN DER SPIEGEL

Although real world signals are analog, it is often desirable to convert them into the digital domain using an analog to digital converter (ADC). Motivating designers to apply this conversion is the efficient transmission and storage of digital signals. Digital representation of an audio signal, for example, allows CD players to achieve virtually error free storage using optical disks [1]. Intricate processing of the signal may also necessitate analog to digital (A/D) conversion, since such processing is only feasible in the digital domain using either conventional digital computers or special purpose digital signal processors (DSPs). Signal processing in the digital domain is also extremely useful in such areas as biomedical applications, providing the needed accuracy for tasks such as ultrasound imaging.

One technique, sigma-delta modulation, has become quite popular for achieving high resolution. One significant advantage of the method is that analog signals are converted using only a 1-bit ADC and analog signal processing circuits having a precision that is usually much less than the resolution of the overall converter.

Although sigma-delta concepts have existed since the middle of the century, it is only in the last two decades that this method has become more attractive [2]. One reason is that recent advances in VLSI technology, focused towards realizing high speed densely packed digital circuits, have made feasible the adequate digital processing of the bit stream. Using sigma-delta A/D methods, high resolution can be obtained for only low to medium signal bandwidths.

This article briefly describes conventional A/D conversion, as well as its performance modeling. We then look at the technique of *oversampling*, which can be used to improve the resolution of classical A/D methods. We discuss how sigma-delta converters use the technique of *noise shaping* in addition to oversampling to allow high resolution conversion of relatively low bandwidth signals. Next, we examine the use of sigma-delta converters to convert narrowband band-pass signals with high resolution. Several parallel sigma-delta converters, which offer the potential of extending high resolution conversion to signals with higher bandwidths, are also described.



1. Nyquist rate sampling showing the original band limited signal spectrum, periodically repeated versions of the signal spectrum due to sampling, and the anti-aliasing filter response needed to band limit the signal.

## PCM A/D Conversion

### Nyquist Rate Conversion

#### Sampling

Analog to digital conversion of a signal is traditionally described in terms of two separate operations: uniform sampling in time, and quantization in amplitude. In the sampling process, a continuous time signal is sampled at uniformly spaced time intervals,  $T_s$ . The samples,  $x[n]$ , of the continuous time signal,  $x(t)$  can be represented as  $x[n] = x(nT_s)$ . The effect, in the frequency domain, of the sampling process is to create periodically repeated versions of the signal spectrum at multiples of the sampling frequency  $f_s = 1/T_s$  [3, pp. 80-87]. This relationship is written in Eq. 1, where  $X_s(f)$  represents the spectrum of the sampled signal, and  $X(f)$  is the spectrum of the original continuous time signal.

$$X_s(f) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X(f - kf_s) \quad (1)$$

The sampling process is shown graphically in Fig. 1 for the case where  $f_s = 2f_B$ , and  $f_B$  is the bandwidth of the signal. In general, the signal can be reconstructed back to continuous time if the repeated versions of the signal spectrum do not overlap. Thus, the signal must be band limited to half the sampling rate, i.e., a signal with bandwidth  $f_B$  must be sampled at a rate greater than twice the bandwidth,  $f_s \geq 2f_B$ .

Interference between the repeated versions of the signal spectrum is known as aliasing and it prevents reconstruction of the signal.

Even if a signal is nominally band limited to  $f/2$ , an anti-aliasing filter is often used to ensure that the signal is indeed band limited. For example, speech has a nominal bandwidth of 4 kHz and so in principle can be sampled at 8 kHz. However, there is some residual signal energy above 4 kHz, which results in aliasing if a 8 kHz sampling rate is used. The anti-aliasing filter is a continuous time analog filter preceding the sampler.

The case where  $f_s = 2f_B$  is known as Nyquist rate sampling, and clearly the anti-aliasing filter here must have a very sharp cutoff at frequency  $f_B = f_s/2$  as shown in Fig 1. Later, we will discuss how the sharp cutoff requirement on this filter can be relaxed.

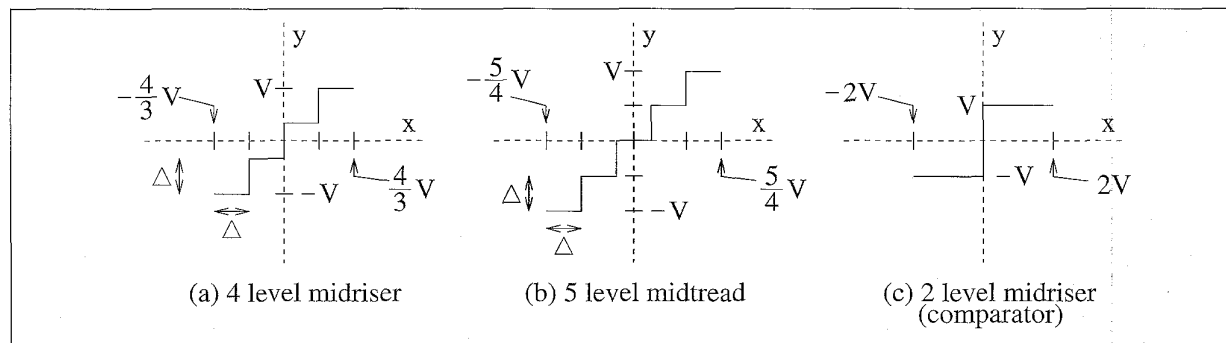
The discretization or quantization in time as a result of the sampling is an invertible operation, since no signal information is lost and the original continuous time signal can be perfectly reconstructed [3, pp. 80-91]. Note that although Fig. 1 shows the sampling process for the case where the signal is a baseband signal, i.e., the spectrum has a bandwidth centered at DC frequency, Eq. 1 still describes the sampled spectrum even if the signal spectrum is centered at some higher frequency  $f_c$ . In this case, for a signal bandwidth  $f_B$ , the signal spectrum occupies the region  $[f_c - f_B/2, f_c + f_B/2]$ , and it will still be possible to avoid aliasing and reconstruct the signal provided that  $f_s \geq 2f_B$ .

#### Quantization

Once sampled, the signal samples must also be quantized in amplitude to a finite set of output values. Typical transfer characteristics of quantizers or A/D converters with an input signal sample,  $x[n]$ , and an output,  $y[n]$ , are shown in Fig. 2.

Quantization is a non-invertible process, since an infinite number of input amplitude values are mapped to a finite number of output amplitude values. The quantized output amplitudes are usually represented by a digital code word composed of a finite number of bits. For example, for the 1 bit A/D converter of Fig. 2c, the output levels  $V$  and  $-V$  can be mapped to digital codes "1" and "0." The digital code words are often said to be in pulse code modulation (PCM) format.

Another way of looking at this would be to plot the digital



2. Transfer characteristics of typical A/D converters (ADCs) or quantizers.

code words instead of the quantized amplitude values for  $y$  in Fig 2. The quantized output amplitude values can also be considered the output of an ideal digital to analog converter (DAC) whose inputs are the corresponding digital code words.

An ADC or quantizer with  $Q$  output levels is said to have  $N$  bits of resolution where  $N = \log_2(Q)$ . As should be clear from Fig 2, for an ADC with  $Q$  quantization levels, only input values separated by at least  $\Delta = 2V/(Q-1)$  can be distinguished or resolved to different output levels.  $N$  digital bits are needed to encode the  $Q$  codewords corresponding with each output level. The difference between the binary digital codes for two adjacent output levels is one least significant bit (LSB) of the overall  $N$  bit codeword. Consequently, a difference in input amplitudes corresponds to a one LSB difference in the digital output code words.

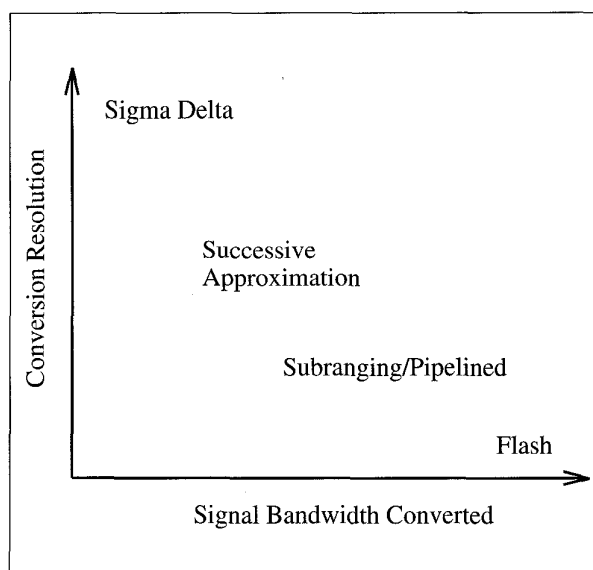
We now point out some general properties of the quantization transfer curves shown in Figs. 2a-2b for a four level (2 bit) "midriser" and a five level (roughly 2 bit) "midtread" ADC. Unlike the midtread ADC, the midriser ADC does not contain a zero output level for a zero input value, effectively creating a DC offset that may be undesirable in some applications. Note that all the transfer characteristics shown in Fig. 2 are symmetric. The midriser needs to have an even number of output levels to produce a completely symmetric transfer curve, whereas the midtread needs an odd number of output levels.

The midriser ADC's symmetric characteristic, with an even number of levels, is an advantage because the number of output levels,  $Q$ , can be made a power of two and encoded with exactly  $N = \log_2(Q)$  bits. However, the number of output levels,  $Q$ , for a symmetric midtread ADC must be odd, and so cannot be made a power of two and encoded as efficiently. The number of bits needed will be  $N = \log_2(Q-1) + 1$ , where  $Q-1$  is chosen a power of two. If the number of levels for the midtread ADC is forced to be a power of two by using only  $Q-1$  levels, it will no longer have a symmetric transfer characteristic and will distort large amplitude symmetric input signals (e.g., a sinusoid). This distortion, of course, may be negligible when the number of output levels is very large.

These issues may play a role in choosing whether a midriser or midtread quantizer transfer characteristic should be used. For the special case of a 2 level quantizer, a midtread characteristic will not be able to represent both positive and negative output levels, and so will severely distort a signal containing samples of both polarities. For this 2 level case, a midriser characteristic, shown in Fig. 2c, will almost always be used.

#### Trading Resolution for Bandwidth

Most conventional A/D converters, such as the successive approximation, subranging, and flash converter types quantize signals sampled at, or slightly above, the Nyquist rate. Consequently, these converters are often referred to as Nyquist rate PCM converters. These, and other converters, provide tradeoffs among signal bandwidth, output resolution, and the complexity of the analog and digital hardware.



3. Bandwidth resolution tradeoffs.

Qualitative bandwidth and resolution tradeoffs of some of these A/D techniques, as well as sigma-delta conversion, is shown in Fig 3. As is evident from Fig 3, sigma-delta A/D converters attain the highest resolution for relatively low signal bandwidths. Consequently, sigma-delta techniques are often used in speech applications where the signal bandwidth is only 4 kHz and where up to 14 bits of resolution may be needed. Similarly, sigma-delta ADCs are popular for digital audio applications, where the signal bandwidth is 20-24 kHz and where high fidelity audio requires 16-18 bits of resolution. Flash converters, on the other hand, may be used for broadcast video applications where the signal band is about 5 MHz, but the resolution required is only about 8 bits.

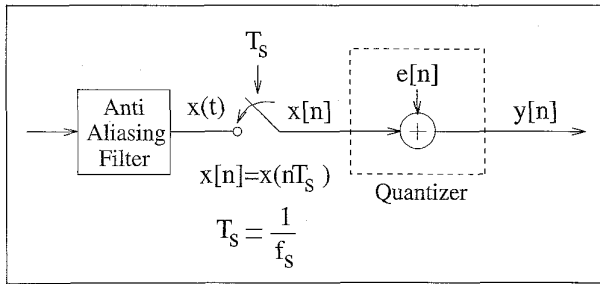
#### Performance Modeling

Having looked at the sampling and quantization processes, we now examine the A/D converter and characterize its performance. The diagrams in Fig. 2 show the transfer characteristic for typical quantizers with input  $x$  and output  $y$ .

Let the maximum and minimum quantized output values always be  $V$  and  $-V$ . The least significant bit (LSB) of an ADC with  $Q$  quantization levels is equivalent to  $2V/(Q-1)$ . For both the midriser and midtread type of ADCs of Fig 2, the magnitude of the quantization error ( $e = y - x$ ) between the output and input does not exceed half a LSB, i.e.,  $|e| \leq \Delta/2$ , provided that  $|x| \leq V + \Delta/2$ . Under these circumstances, the quantizer or ADC is said to be not overloaded. For  $|x| > V + \Delta/2$  (hence,  $|e| > \Delta/2$ ), the ADC is said to be overloaded.

The quantizer embedded in any ADC is a non-linear system, which makes its analysis difficult. To make the analysis tractable, the quantizer is often linearized and modeled by a noise source,  $e[n]$ , added to the signal  $x[n]$ , to produce the quantized output signal  $y[n]$ :

$$y[n] = x[n] + e[n] \quad (2)$$



4. Block diagram and model of a conventional A/D converter (ADC) system.

A block diagram of an A/D system showing the sampling process and the quantizer model is shown in Fig 4. To further simplify the analysis of the noise from the quantizer, the following assumptions about the noise process and its statistics are traditionally made [3, p. 120]:

- The error sequence,  $e[n]$ , is a sample sequence of a stationary random process.
- $e[n]$  is uncorrelated with the sequence  $x[n]$ .
- The probability density function of the error process is uniform over the range of quantization error, i.e., over  $\pm \Delta/2$ .
- The random variables of the error process are uncorrelated, i.e., the error is a white noise process.

Under certain conditions, such as when the quantizer is not overloaded,  $N$  is large, and the successive signal values are not excessively correlated, these assumptions are reasonable [4]. Consider an  $N$  bit ADC with  $Q = 2^N$  quantization levels, i.e., with  $\Delta = 2V/Q - 1 = 2V/(2^N - 1)$ . For a zero mean  $e[n]$ , its variance  $\sigma_e^2$  or power is

$$\sigma_e^2 = \frac{\Delta^2}{12} = \left( \frac{2V}{2^N - 1} \right)^2 / 12 \cong \left( \frac{2V}{2^N} \right)^2 / 12 \quad (3)$$

If the signal is treated as a zero mean random process and its power is  $\sigma_x^2$ , then the signal to quantization noise ratio is:

$$SNR = 10 \log \left( \frac{\sigma_x^2}{\sigma_e^2} \right) = 10 \log \left( \frac{\sigma_x^2}{V^2} \right) + 4.77 + 6.02 \text{ (dB)} \quad (4)$$

Note that for each extra bit of resolution in the ADC, i.e., for every increment in  $N$ , there is about a 6 dB improvement in the SNR. Thus, there is a direct relationship between the resolution of an ADC and its SNR, and it is common to equate differences in SNR in dB to bits by dividing the dB value by 6. For example, if an ADC has a SNR that is 3 dB better than that of another ADC, the better ADC will be said to have 1/2 bit higher resolution. Also, note that for a given  $N$ , the SNR in dB is linearly related to the signal power,  $\sigma_x^2$ , in dB.

Let us now examine the dynamic range of the ADC, which is a measure of the range of input amplitudes for which the ADC produces a positive SNR. For sinusoidal inputs, the dynamic range of the A/D converter is defined as the ratio of the signal power of a full scale sinusoid to the signal power

of a small sinusoidal input that results in a SNR of 1 (or 0 dB) [5]. The signal power of a full scale sinusoid is  $V^2/2$ . A sinusoid with signal power  $\sigma_x^2 = \sigma_e^2 = \Delta^2/12$  will result in an SNR of 1, or 0 dB.

The dynamic range, by definition, is

$$\left( \frac{V^2}{2} / \frac{\Delta^2}{12} \right) \cong \left( \frac{V^2}{2} / \frac{(2V/2^N)^2}{12} \right)$$

This expression reduces to a dynamic range value given by:

$$R = 6.02N + 1.76 \text{ (dB)} \quad (5)$$

Note that the ratio of  $V^2/2$  to  $\Delta^2/12$  is just the peak SNR of the ADC for a sinusoidal input. Consequently, the dynamic range of the Nyquist rate ADC is the same as its peak SNR. Sigma-delta converters do not necessarily have their peak SNR equal to their dynamic range. However, by using the dynamic range of a sigma-delta converter in Eq. 5 and calculating the corresponding  $N$ , we will be able to determine the resolution of a Nyquist rate PCM converter that would be required to produce the same dynamic range.

#### Limitations of Nyquist Rate ADCs

For Nyquist rate converters, each signal sample is quantized at the full precision or resolution of the converter. The resolution of such converters implemented on VLSI chips is limited by the technology in which these chips are fabricated. For example, some successive approximation A/D techniques rely on matching of two capacitors to perform a repeated division of a reference voltage by 2.

If such a converter is to convert signal values to  $N$  bits of resolution, the required matching on the capacitor components needs to be at least one part in  $2^N$ . Matching of components to greater than 10 bits (one part in  $2^{10}$ ) or equivalently to more than 0.1% is difficult in VLSI.

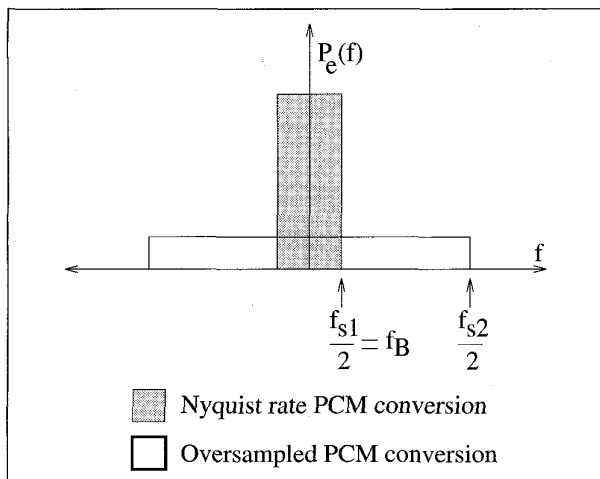
High resolution Nyquist rate converters are extremely difficult to attain in current integrated circuit technology without the use of techniques such as laser trimming of components or calibration. Furthermore, if the signal is sampled too close to the Nyquist rate, the anti-aliasing filter must have a very sharp cutoff, which is a non-trivial design requirement for analog filters.

#### Oversampled PCM Conversion

##### System Description

Oversampled PCM conversion is a technique that improves the resolution obtained from straightforward Nyquist rate PCM conversion. This improvement is achieved by oversampling the signal, i.e., samples are acquired from the analog waveform at a rate significantly faster than the Nyquist rate. Each of these samples is quantized by an  $N$  bit ADC. Since quantization is described by Eq. 2, the total amount of noise power injected into the sampled signal,  $x[n]$ , is  $\sigma_e^2$  and is given by Eq. 3.





5. Quantization noise power spectral density for Nyquist rate PCM and oversampled PCM conversion.

Obviously, this is exactly the same noise power produced by a Nyquist rate converter, but its frequency distribution is different because of the higher sampling rate. The performance modeling criteria designated the noise process as white, which means that the noise power is uniformly distributed between  $-f_s/2$  to  $f_s/2$ , where  $f_s$  is the sampling frequency.

Fig 5 shows the power spectral density,  $P_e(f)$ , of the quantization noise for Nyquist rate sampling with rate  $f_{s1}$  and oversampling rate  $f_{s2}$ . For Nyquist rate sampling where the signal band,  $f_B = f_{s1}/2$ , all the quantization noise power, represented by the area of the tall shaded rectangle, occurs across the signal bandwidth.

In the oversampled case, the same noise power, represented by the area of the unshaded rectangle has been spread over a bandwidth equal to the sampling frequency,  $f_{s2}$ , which is much greater than the signal bandwidth,  $f_B$ . Only a relatively small fraction of the total noise power falls in the band  $[-f_B, f_B]$ , and the noise power outside the signal band can be greatly attenuated with a digital low-pass filter following the ADC.

After the low-pass filtering is performed, the signal can be downsampled to the Nyquist rate without affecting the signal to noise ratio. The collective operation of low-pass filtering and downsampling is known as decimation. The low-pass filter and downsampler are collectively called a decimator. A

block diagram of an oversampled PCM system showing the sampling, the ADC model, and the decimator is presented in Fig 6.

**Performance Modeling for Oversampled PCM Converters**  
By taking the Z transform of Eq. 2, we obtain the Z domain relationship between the input and output of an oversampled PCM converter:

$$Y(z) = X(z) + E(z) \quad (6)$$

where  $Y$ ,  $X$  and  $E$  are the Z transforms of the output, input signal and the quantization error process, respectively. Based on our two-input linear system model for the quantizer, Eq. 6 states that in the Z domain, the output is the input plus the quantization error or noise. We can observe that  $X$  and  $E$  both experience a unity transfer function.

A more general way of writing Eq. 6, where  $X$  and  $E$  do not necessarily experience a unity transfer function, is

$$Y(z) = X(z)H_x(z) + E(z)H_e(z) \quad (7)$$

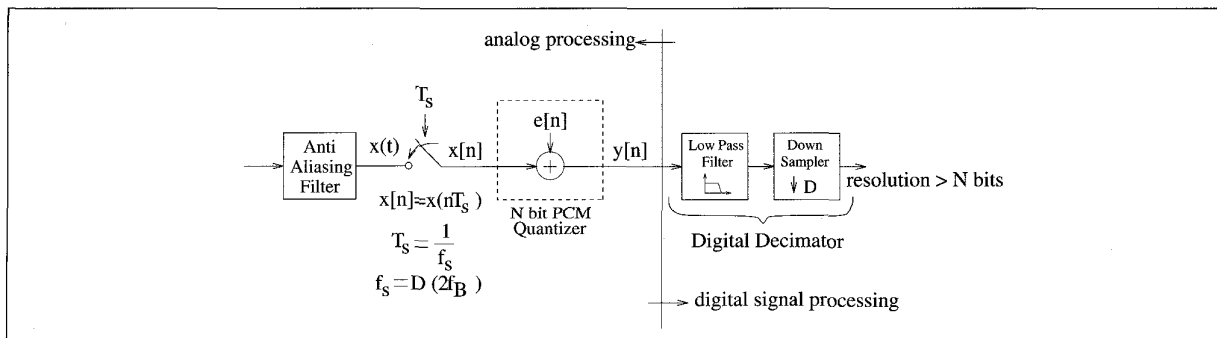
The output is now the input signal modulated by a signal transfer function, denoted by  $H_x(z)$ , plus the quantization noise modulated by a noise transfer function, denoted by  $H_e(z)$ . To evaluate the performance of such a converter, we need to find the total signal and noise power at the output of the converter. To do this, we need to evaluate the power spectral densities,  $P_{xy}(f)$  and  $P_{ey}(f)$ , of the signal and noise at the output of the converter, based on the power spectral densities,  $P_x(f)$  and  $P_e(f)$  of the signal and noise at the input of the converter. We can make use of the fact that if a stationary random process with power spectral density  $P(f)$  is the input to a linear filter with transfer function  $H(f)$ , the power spectral density of the output random process is  $P(f)|H(f)|^2$ . Consequently,

$$P_{xy}(f) = P_x(f)|H_x(f)|^2$$

$$P_{ey}(f) = P_e(f)|H_e(f)|^2$$

For the oversampled PCM converter,  $|H_x(f)| = |H_e(f)| = 1$ , and our white noise assumption for  $e[n]$  states that  $P_e(f) = \sigma_e^2/f_s$ , which implies  $P_{ey}(f)$  is also  $\sigma_e^2/f_s$ .

Assuming an ideal low-pass filter with cutoff frequency



6. Oversampled PCM conversion system.

$f_B$  following the oversampled quantizer, the *in-band* noise power,  $\sigma_{ey}^2$  at the output of the A/D is

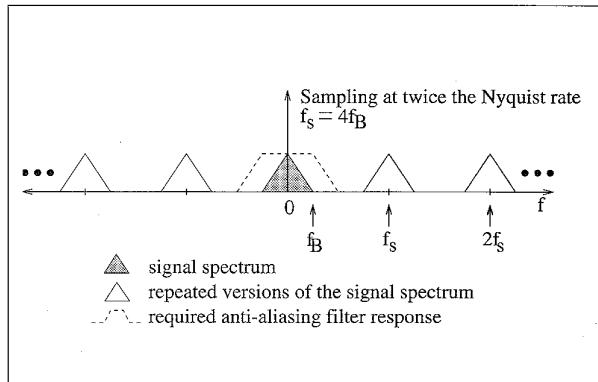
$$\begin{aligned}\sigma_{ey}^2 &= \int_{-f_B}^{f_B} P_{ey}(f)df = 2 \int_0^{f_B} P_{ey}(f)df \\ &= \int_0^{f_B} \frac{2\sigma_e^2}{f_s} df = \sigma_e^2 \left( \frac{2f_B}{f_s} \right)\end{aligned}$$

Note that some of the noise power is now located outside of the signal band as a result of the oversampling, and so the in-band power  $\sigma_{ey}^2$  is less than what it would have been without any oversampling ( $\sigma_e^2$ ). Since the signal power is assumed to occur over the signal band only, it is not modified in any way and the signal power at the output ( $\sigma_{xy}^2$ ) is the same as the input signal power  $\sigma_x^2$ . The maximum achievable SNR in dB is then:

$$\begin{aligned}\text{SNR} &= 10 \log \left( \frac{\sigma_x^2}{\sigma_{ey}^2} \right) \\ &= 10 \log(\sigma_x^2) - 10 \log(\sigma_e^2) + 10 \log \left( \frac{f_s}{2f_B} \right) \quad (\text{dB})\end{aligned} \quad (8)$$

For the case of Nyquist rate sampling where  $f_s = 2f_B$ , this formula reduces to Eq. 3, which is the SNR for the Nyquist rate PCM quantizer. Letting the oversampling ratio  $f_s/2f_B = 2^r$ , we obtain,

$$\text{SNR} = 10 \log(\sigma_x^2) - 10 \log(\sigma_e^2) + 3.01r \quad (\text{dB}) \quad (9)$$



7. Sampling at twice the Nyquist rate, showing the original band limited signal spectrum, repeated versions of the signal spectrum due to sampling, and an anti-aliasing filter that is sufficient to band limit the signal.

For every doubling of the oversampling ratio, i.e., for every increment in  $r$ , the SNR improves by about 3 dB, or the resolution improves by one-half bit.

Note that in this scheme, we are trading speed for resolution. The higher resolutions are obtained at the expense of requiring the internal PCM quantizer to quantize samples at the oversampled rate. Analog circuit complexity has also been traded for digital circuit complexity. The analog circuit

complexity is simplified, since we have said the resolution of the internal  $N$  bit quantizer, an analog circuit, is lower than that of the overall conversion resolution.

Another benefit, which is a direct consequence of the oversampling, is that the analog anti-aliasing filter does not need as sharp a cutoff. This can be seen from Fig. 7, where a signal is sampled at four times the nominal signal bandwidth (twice the Nyquist rate). In this case, the anti-aliasing filter can have a transition band between  $f_B$  and  $f_s/2$  as long as it provides very good attenuation beyond  $f_s/2$ . However, a price is paid in the digital domain, since the digital filter must attenuate the remaining quantization noise power (beyond  $f_B$ ) as much as possible. In the process of filtering out-of-band quantization noise, any other noise that existed in the transition band of the anti-aliasing filter prior to sampling will be attenuated further. The closer the low-pass filter approximates an ideal low-pass filter, the more resources it will need.

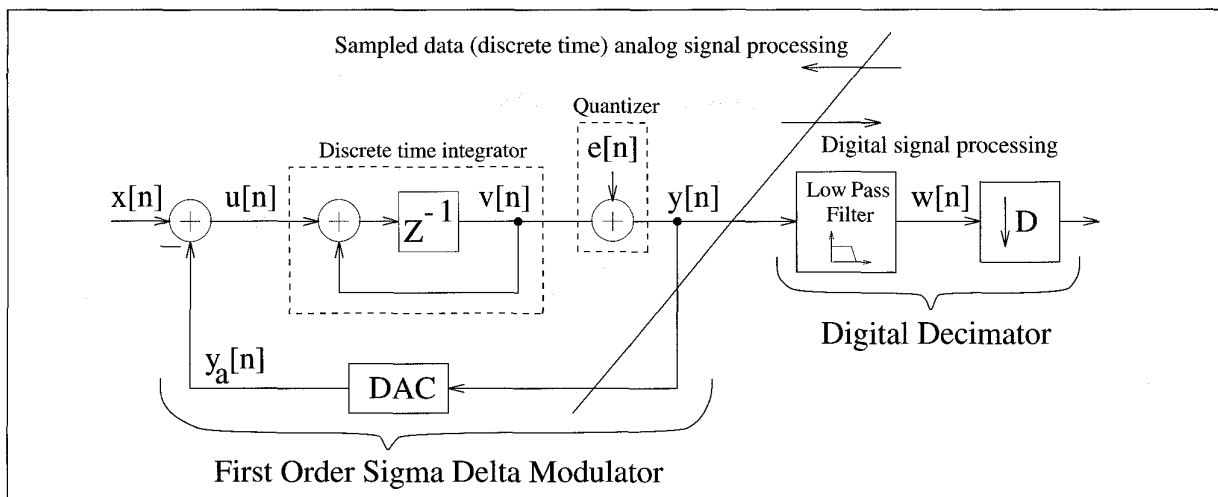
Finally, note that for high resolution conversion, one needs,  $f_s \gg f_B$ , and the signal bandwidth must be small so that  $f_s$  does not exceed the maximum circuit speed attainable in the given technology. Let us use Eq. 8 for a simple calculation. Suppose we apply a full scale sinusoid with amplitude  $V = 1$  corresponding with signal power  $V^2/2 = 0.5$ , as the input to an oversampled PCM ADC with a 20 kHz audio range signal band. Let the final desired resolution be 16 bits (for CD quality audio), which corresponds to a 98 dB SNR according to Eq. 4. Now, if we use an 8 bit A/D in an oversampled PCM scheme, i.e., if we use  $N = 8$  in the expression for  $\sigma_e^2$  in Eq. 8, we can calculate the  $f_s$  required for SNR = 98 dB and  $f_B = 20$  kHz. The needed  $f_s$  is 2.64 GHz! Eight bit ADCs implemented in current CMOS technology certainly can not operate at such a high speed.

Suppose a 12 bit internal ADC is used instead. In this case, the required  $f_s$  is about 10 MHz, an operating speed that is not extremely high, per se, but which is still not trivial for a 12 bit ADC to attain. We will later see how sigma-delta modulation A/D conversion allows the use of internal ADCs with as low as 1 bit (i.e.,  $N = 1$ ) of resolution to achieve an overall resolution of 16 bits for a 20 kHz audio bandwidth.

## Sigma-Delta Modulation A/D Conversion

A general way of writing the  $Z$  domain output of an A/D converter was given in Eq. 7, as  $Y(z) = X(z)H_x(z) + E(z)H_e(z)$ , where  $H_x$  is the signal transfer function (STF) and  $H_e$  is the noise transfer function (NTF). For oversampled PCM conversion, we saw that  $H_x(z) = H_e(z) = 1$ . This need not be the case and, in fact, oversampled A/D converters can be designed to incorporate noise shaping, where  $H_e$  is designed to be different from  $H_x$  such that  $H_x$  usually leaves the signal undisturbed but  $H_e$  shapes the noise to allow a high resolution output [2, 6].

Although the term delta-sigma ( $\Delta\Sigma$ ) was used by some of the earliest researchers in the field [7], the term sigma-delta ( $\Sigma\Delta$ ) has also become almost synonymous with noise shaping ADCs. We will use the term sigma-delta to describe noise shaping ADCs. As noted, oversampling reduces the quanti-



8. First order sigma-delta modulator A/D system.

zation noise power in the signal band by spreading a fixed quantization noise power over a bandwidth much larger than the signal band. Noise shaping or modulation further attenuates this noise in the signal band and amplifies it outside the signal band. Consequently, this process of noise shaping by the sigma-delta modulator can be viewed as pushing quantization noise power from the signal band to other frequencies. The modulator output can then be low-pass filtered to attenuate the out-of-band quantization noise and finally can be downsampled to the Nyquist rate.

The price of attaining high resolution is again a penalty in speed, as the hardware has to operate at the oversampled rate, and an increased complexity of the digital hardware. For high resolution conversion, the sampling frequency  $f_s$  must still be much greater than the signal bandwidth  $f_b$ , but, of course, not as great as needed by oversampled PCM conversion.

## First Order Sigma-Delta Modulation

### Operation and Performance Modeling

A block diagram of a first order sigma-delta modulator A/D system is shown in Fig 8. The system consists of an analog sigma-delta modulator, followed by a digital decimator. The modulator consists of an integrator, an internal A/D converter or quantizer, and a D/A converter (DAC) used in the feedback path.

The signal that is quantized is not the input  $x[n]$  but a filtered version of the difference between the input and an analog representation,  $y_a[n]$ , of the quantized output,  $y[n]$ . The filter, often called the feedforward loop filter, is a discrete time integrator whose transfer function is  $z^{-1}/(1-z^{-1})$ .

The integrator and the rest of the sigma-delta analog circuit are typically implemented in sampled data switched capacitor technology. Consequently, the sampling operation is not shown explicitly in Fig. 8 or any other modulator architectures to be described in the rest of this article. Continuous-time versions of the modulator have also been con-

sidered [8], but this aspect of the modulators will not be discussed here. The linearized model replaces the quantizer with a noise source,  $e[n]$ , as shown in Fig 8.

If the DAC is ideal, it is replaced by a unity gain transfer function. The modulator output  $Y(z)$  is then given by:

$$Y(z) = X(z)z^{-1} + E(z)(1-z^{-1}) \quad (10)$$

so that  $H_x(z) = z^{-1}$  and  $H_e(z) = (1-z^{-1})$ . The output is just a delayed version of the signal plus quantization noise that has been shaped by a first order Z domain differentiator or high-pass filter. The corresponding time domain version of the modulator output is

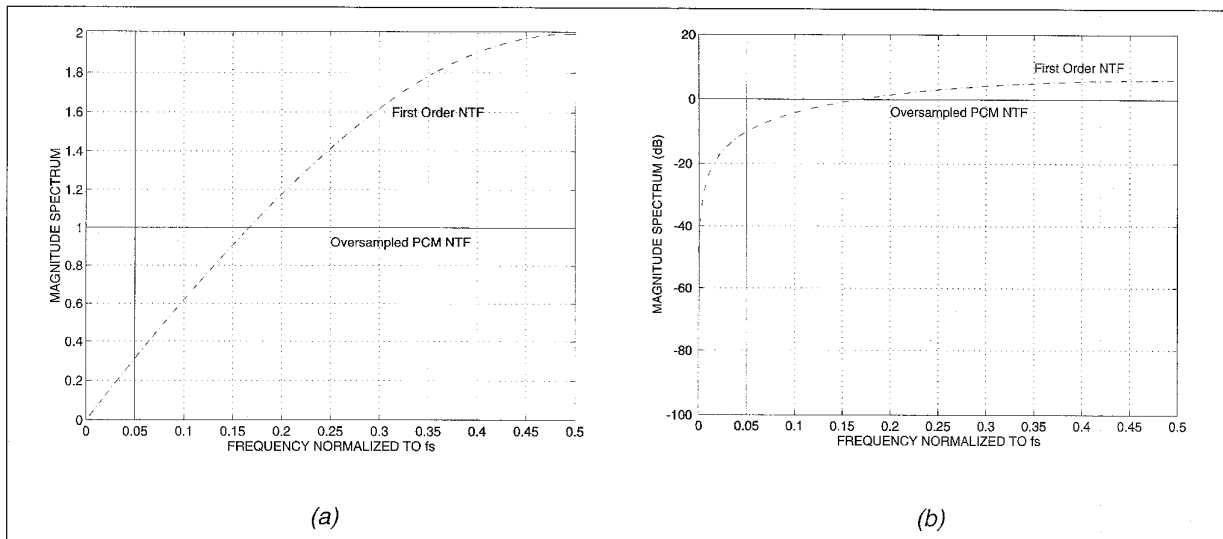
$$y[n] = x[n-1] + e[n] - e[n-1] \quad (11)$$

where the  $e[n]-e[n-1]$  term is the first order difference of  $e[n]$ .

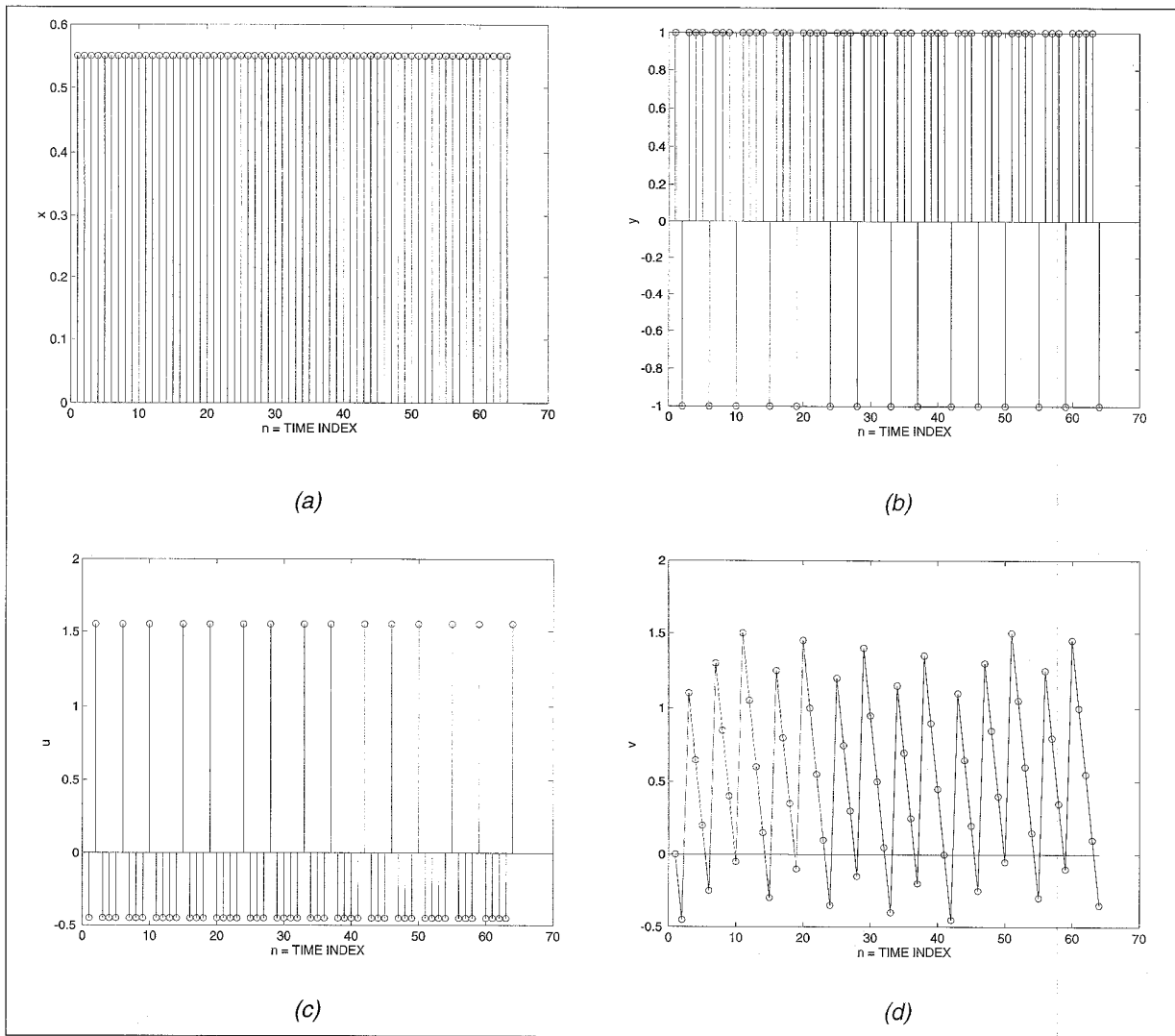
The magnitude spectrum of a first order sigma-delta noise transfer function (NTF) is plotted in Fig 9a, while Fig 9b shows the same plot in dB. The frequency axis has been normalized with respect to the sampling frequency,  $f_s$ .

Since  $H_e(z)$  contains a zero at  $z = 1$ , i.e., at DC frequency on the unit circle of the Z plane, note the zero gain or infinite attenuation provided by the NTF at DC frequency. Note the large attenuation at lower frequencies and relative amplification at higher frequencies. For comparison, the oversampled PCM NTF, which has unity gain, is shown in Fig 9(a). The vertical bar demarcates the extent of the signal band,  $f_b$ , where  $f_b = 0.05 f_s$ . Quantization noise to the left of the bar that contributes to the finite resolution of the modulator is greatly attenuated while noise to the right of the bar is not attenuated as much or is actually amplified. However, noise to the right of the bar can mostly be removed with a digital low-pass filter.

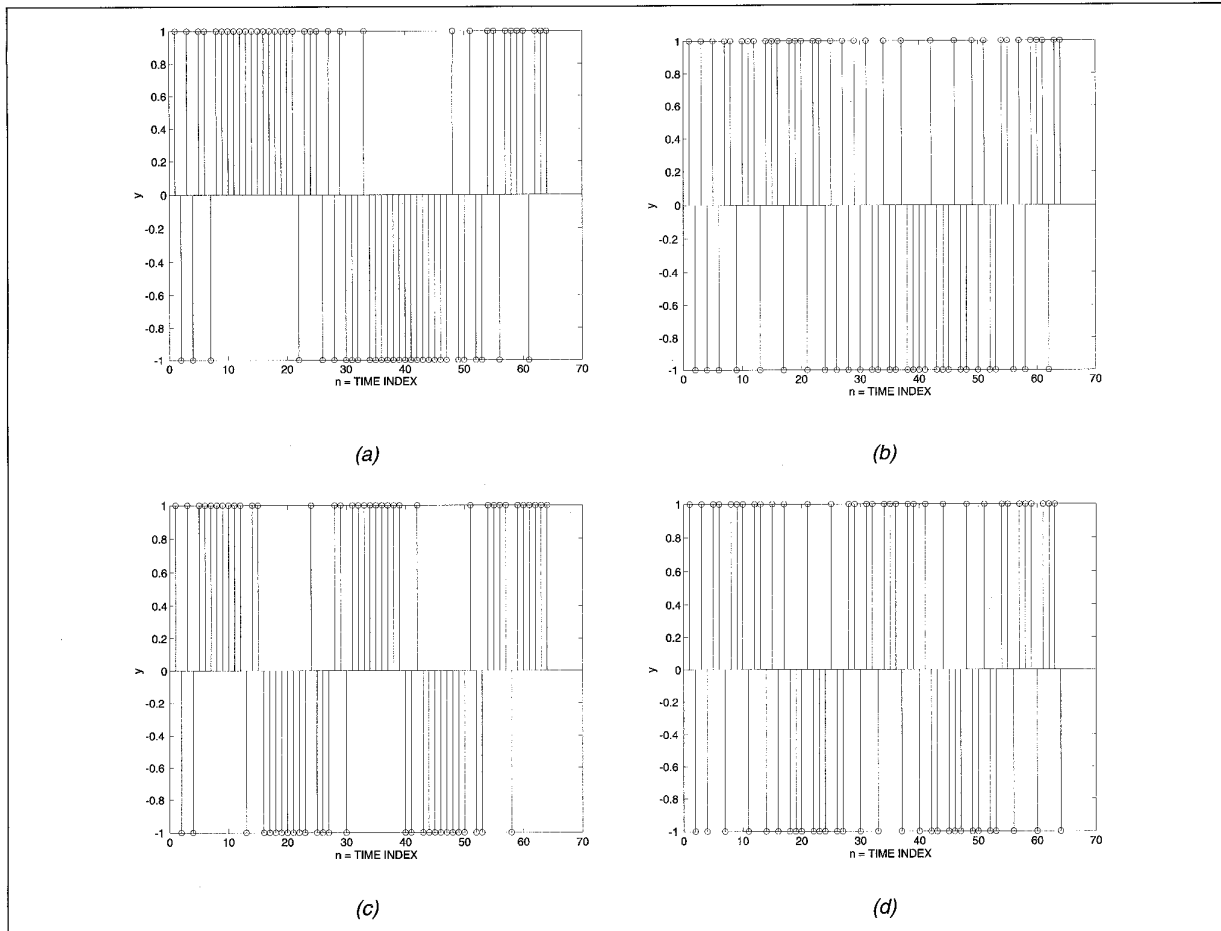
Let us now examine the DAC before going on to examine the performance of the converter. The DAC is required to be nearly as linear as the overall conversion resolution. Any D/A non-linearity can be modeled as an error source that adds directly to the input. This error source benefits from the



9. NTF for a first order sigma-delta modulator (a) magnitude spectrum on linear scale. For comparison, the oversampled PCM NTF, which has unity gain, is shown; (b) magnitude spectrum in dB.



10. 1st order sigma-delta responding to a DC input: (a) DC input  $x[n] = 0.55 = 11/20$ . (b) modulator output  $y[n]$ ; (c) "error" signal  $u[n]$ ; (d) integrator output  $v[n]$



11. 1st order sigma-delta responding to various sinusoidal inputs with sampling frequency of about 1 MHz: (a) amplitude = 0.95, frequency = 20 kHz; (b) amplitude = 0.5, frequency = 20 kHz; (c) amplitude = 0.95, frequency = 40 kHz; (d) amplitude = 0.5, frequency = 40 kHz.

oversampling but unlike  $e[n]$ , which models the A/D quantization error, is not subject to the noise shaping.

Since a 1 bit DAC is perfectly linear, it is common to use a 1 bit DAC and a corresponding 1 bit quantizer, which is simply a comparator. Consequently, provided the sampling frequency is high enough, the sigma-delta A/D allows the use of a 1 bit quantizer to achieve high overall resolution. Using  $H_x = z^{-1}$ ,  $H_e = (1 - z^{-1})$ , and the procedure used for the oversampling PCM A/D, the in-band noise power (i.e., the noise in the frequency range  $[-f_B, f_B]$ ) at the output of a first order sigma-delta modulator is

$$\sigma_{ey}^2 = \sigma_e^2 \frac{\pi^2}{3} \left( \frac{2f_B}{f_s} \right)^3 \quad (12)$$

The SNR in dB is then:

$$\begin{aligned} SNR = & 10 \log(\sigma_x^2) - 10 \log(\sigma_e^2) - 10 \log\left(\frac{\pi^2}{3}\right) \\ & + 30 \log\left(\frac{f_s}{2f_B}\right) \quad (dB) \end{aligned} \quad (13)$$

Letting the oversampling ratio,  $f_s/2f_B = 2^r$ , we obtain:

$$\begin{aligned} SNR = & 10 \log(\sigma_x^2) - 10 \log(\sigma_e^2) \\ & - 10 \log\left(\frac{\pi^2}{3}\right) + 9.03r \quad (dB) \end{aligned} \quad (14)$$

For every doubling of the oversampling ratio i.e., for every increment in  $r$ , the SNR improves by 9 dB, or equivalently, the resolution improves by 1.5 bits.

Let us revisit the example considered at the end of the discussion on oversampled PCM. It was desired to convert a 20 kHz audio band to CD quality resolution of 16 bits. Using Eq. 13, we can compute that the required  $f_s$  with a 1 bit internal ADC is 96.78 MHz. A 1 bit ADC or comparator can operate at this speed in current CMOS technology. However, it is not possible for the sampled data analog switched capacitor integrators to operate at such high speeds. Second order sigma-delta modulation can allow the use of a 1 bit quantizer, allowing us to meet the 16 bit 20 kHz target with a much more reasonable  $f_s$ .

## Qualitative Time Domain Behavior

The sigma-delta modulator can be thought of as a PCM converter with feedback, which attempts to force the output signal  $y[n]$  to be equal to the input signal,  $x[n]$ . Consider the case where a 1 bit A/D converter or comparator is used. The transfer characteristics of this 1 bit ADC with output levels  $V$  and  $-V$  is shown in Fig 2c.

Assume that  $V = 1$ , such that the comparator's digital output is 1 or -1, so that  $y[n]$  and  $y_a[n]$  can be used interchangeably. In the time domain, referring to Fig. 8, we have,

$$v[n] = u[n-1] + v[n-1] \quad (15)$$

$$y[n] = \begin{cases} 1 & v[n] \geq 0 \\ -1 & v[n] < 0 \end{cases} \quad (16)$$

$$u[n] = x[n] - y[n] \quad (17)$$

The "error" between the modulator output and input is  $u[n]$ . Note that this is not the quantization error, which is given by  $e[n] = y[n] - v[n]$ .

Since  $y[n]$  can take on values of 1 or -1 only, it can never equal the input unless the input happens to be one of these two values exactly. Consequently, except for the mentioned cases, there will always be an error,  $u[n] \neq 0$ . Consider a DC input for  $x[n]$ . When  $y[n] = 1$ ,  $y[n]$  is greater than the input  $x[n]$  and the error  $u[n]$  is negative, and so negative values are accumulated by the integrator to produce  $v[n]$ . After a number of clock cycles, enough negative values will have accumulated to cause the quantizer to produce  $y[n] = -1$ , thereby changing the sign of the error  $u[n]$  to be positive. The error between the output and input has been reduced, in some sense, because the positive errors will now cancel the prior negative errors when averaged over a period of time.

Now with  $y[n] = -1$ , the errors will be positive, and positive values of the error will be accumulated again until the quantizer output changes, this time back to  $y[n] = 1$ . Over a period of time, the proportion (or density) of 1's and -1's will be related to the DC input value—the larger the input, the more 1's will be present in the output, and vice versa, for smaller inputs. For this reason, the output of a sigma-delta modulator using a 1 bit quantizer is often said to be in pulse density modulated (PDM) format.

Let us now illustrate the time domain behavior using a few examples. Fig 10a shows a DC input  $x[n] = 0.55 = 11/20$ , while Fig 10b shows the corresponding modulator output,  $y[n]$ . Roughly, three fourths of the output values are 1's, and the others are -1's. Fig 10c shows the error signal,  $u[n]$ , and Fig 10d shows the accumulated error signal or integrator output  $v[n]$ , whose sign change forces the quantizer output to change. For a DC input of  $x[n] = 1$ , all the modulator output values will be 1's. For a zero DC input, half the modulator output values will be 1's, half -1's. For a DC input of -1, all the values will be -1's.

By averaging the modulator output over a period of time, we can approximate the input. This averaging operation represents the low-pass filter block in Fig. 8, since averaging is a crude low-pass filtering operation. Using a better low-pass filter will result in the modulator output being a better approximation to the input for a given oversampling ratio.

Finally, let us look at some time domain examples of the modulator output for sinusoidal inputs. Figures 11a-d show the modulator outputs for various sinusoidal inputs. As for the DC input case, the sinusoid amplitude information is encoded in the relative number of 1's vs -1's. The modulator output pulse pattern has periodic components, and the fundamental period encodes the sinusoid frequency. This is particularly clear in Figs. 11a-c.

## Implementation Imperfections

The results presented thus far have not considered imperfections in the analog hardware. Let us now discuss the consequences of imperfections in some of the main circuit parameters.

The integrator in the modulator may have a gain of  $g$  instead of unity, and may be leaky. For an input  $u[n]$ , an integrator with gain  $g$  and leakage factor  $\alpha$  has an output  $v[n] = g u[n-1] + \alpha v[n-1]$  instead of  $v[n] = u[n-1] + v[n-1]$ , and the integrator transfer function is  $g z^{-1} / (1 - \alpha z^{-1})$  instead of  $z^{-1} / (1 - z^{-1})$ .

The D/A gain may also not be perfectly unity, and assuming a gain of  $d$ , we find the STF and NTF are  $H_x = g z^{-1} / (1 + (g d - \alpha) z^{-1})$  and  $H_e = 1 - \alpha z^{-1} / (1 + (g d - \alpha) z^{-1})$ .

The original NTF ( $1 - z^{-1}$ ), which had a  $Z$  domain zero at  $z = 1$  (on the unit circle and at DC), now has a zero which is still at DC but is moved inside the unit circle. This degrades the NTF noise attenuation in the signal band and can thus affect the noise performance significantly. The term "leaky" comes from the fact that there is charge leakage in the switched capacitor implementation of the integrator circuit and only a portion of the charge from the input capacitor is transferred to the integrating capacitor.

The leakage factor  $\alpha$  is related to the open loop gain,  $A$ , of the operational amplifier (opamp) used to implement the switched capacitor integrator such that  $1 - \alpha \approx 1/A$ . Ignoring the denominator (which will add a slight ripple to the numerator) of the degraded  $H_e$ , i.e., by considering the degraded NTF to be  $(1 - \alpha z^{-1})$ , we find the in-band quantization noise power with integrator leakage is:

$$\sigma_{ey}^2 = \sigma_e^2 \left( \frac{2f_B}{f_s} \right) \frac{1}{A^2} + \alpha \sigma_e^2 \frac{\pi^2}{3} \left( \frac{2f_B}{f_s} \right)^3 \quad (18)$$

instead of Eq. 12. Note that the noise power now contains a term that is inversely proportional to the oversampling ratio,  $f_s/2f_B$ , as well as a term inversely proportional to the oversampling ratio cubed. However, the first term is divided by  $A^2$ , and it has been found that if the opamp open loop gain,  $A$ , exceeds the oversampling ratio,  $f_s/2f_B$ , leakage causes no significant degradation of the SNR [2]. Consequently, the

circuit constraint required to implement good integrators is not that difficult to meet unless the oversampling ratio is extremely high.

From the linearized analysis, the STF and NTF pole is stable only for  $0 < gd < 2$ . Consequently, there is a relatively wide margin over which the two gains may vary from this point of view. It has been reported [2] that variations of  $g$  by up to 10% from unity does not degrade the SNR significantly. The gain is implemented in practice as a ratio of two capacitors and so the corresponding precision on the capacitor matching is minimal—one part in 10, or three and a quarter bits.

Now consider an imperfect DAC gain,  $d$ , that is slightly different from unity. This can be modeled as a gain of  $1/d$  at the input of the modulator. To see this, consider that a gain of  $1/d$  is inserted at the input of the integrator. This gain can be moved past the summing node at the modulator input in Fig 8. The result is that the DAC gain  $d$  is cancelled by the gain  $1/d$  but the input now experiences a gain of  $1/d$  before the modulator. Consequently, the STF experiences a slight gain change but there is no great impact on the modulator SNR.

Finally, consider imperfections in the quantizer. Any non-linearity in the quantizer can be modeled as another noise source which adds to  $e[n]$ , the quantization error. However, the noise from this extra source is subject to noise shaping by the modulator and so its affect on SNR degradation is not significant.

If the 1 bit quantizer or comparator has a non-zero threshold,  $v_{th}$ , its output is given by:

$$y[n] = \begin{cases} 1 & v[n] - v_{th} \geq 0 \\ -1 & v[n] - v_{th} < 0 \end{cases}$$

This is equivalent to an offset at the input of the comparator, i.e., at the output of the integrator. However, an offset  $v_{th}$  at the output  $v[n]$  of the integrator corresponds to an impulse at its input  $u[n] = x[n] - y[n]$  which amounts to one incorrect output  $y[n]$  for a given  $x[n]$ . One such incorrect value will have a negligible impact on the overall performance of the modulator. The offset can also be modeled as an error source, with mean  $v_{th}$  and zero variance, that adds to  $e[n]$ .

If for any reason, the offset is input dependent or changes with time, this new error source will have non-zero variance. However, it will be subject to the noise shaping property of the modulator just as  $e[n]$  and its presence will not degrade the performance of the modulator significantly.

## Non-linear Behavior

A sigma-delta modulator is a non-linear system incorporating feedback. Not surprisingly, the modulator may display limit cycle oscillations that result in the presence of periodic (tone) components in the output. This phenomenon is analogous to limit cycles that occur in digital IIR filters operating with finite precision arithmetic, because, like a sigma-delta modulator, such a filter is a non-linear system that employs feed-

back. The quantizer error spectrum is not white, which is not surprising as the conditions for the white noise assumption are not perfectly satisfied—the quantizer has only two output levels, and due to the oversampling, successive quantizer input samples may be correlated.

Now consider the existence of limit cycles in the modulator, as has been done in [9] for the simple case of a DC input,  $x[n] = x$ . For a limit cycle of period  $T$ ,  $v[n]$  should be periodic with period  $T$ , i.e.,  $v[n] = v[n+T]$ . This clearly implies that  $y[n] = y[n+T]$ .

For the DC input, the input to the integrator,  $u[n] = x - y[n]$ , will likewise be periodic with period  $T$ . Thus, the modulator behavior can be represented by  $T$  equations.

Combining Eqs. 15 and 17, we obtain:  $v[n] - v[n-1] = x[n-1] - y[n-1]$ , which for a DC input becomes  $v[n] - v[n-1] = x - y[n-1]$ . Writing this equation for  $T$  different time instances starting (arbitrarily) at  $n = 1$ , and adding up all these equations we obtain,

$$v[T] - v[0] = \sum_{l=0}^{T-1} x - \sum_{l=0}^{T-1} y[l] \quad (19)$$

However,  $v[T] = v[0]$  by assumption, and consequently:

$$x = \frac{1}{T} \sum_{l=0}^{T-1} y[l] = \frac{(P - M)}{T} V \quad (20)$$

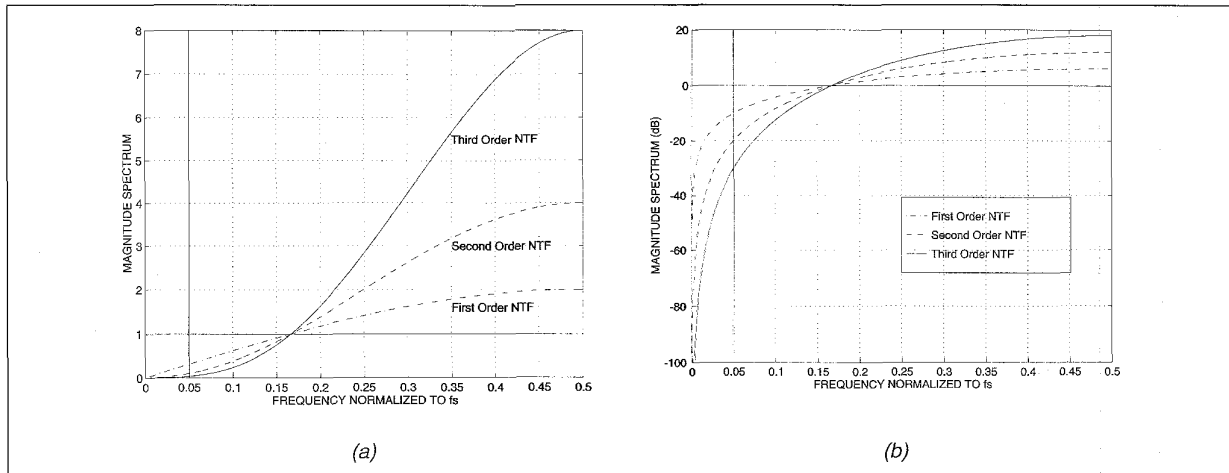
where  $P$  is the number of positive quantizer outputs over  $T$  output samples, and  $M$  is the number of negative quantizer outputs over  $T$  output samples. Since  $(P - M)$  is an integer as is  $T$  by assumption, we have  $x = bV/a$ , with  $a$  and  $b$  integers. Thus, the output  $y$  consists of a limit cycle with period  $T$ , provided that  $x$  is a rational multiple of  $V$ .

The limit cycle with period  $T$  will manifest itself in the output spectrum as tones present at frequency  $f_s/T$  and its harmonics. The period is  $T = 2a$  if  $a$  or  $b$  is even.  $T = a$  if both  $a$  and  $b$  are odd [11]. For the prior example of  $x = 11/20$  in Fig 10,  $a = 20$  and  $b = 11$ , and as can be seen in the figure,  $u[n]$ ,  $v[n]$ , and  $y[n]$  do have the expected period of 40.

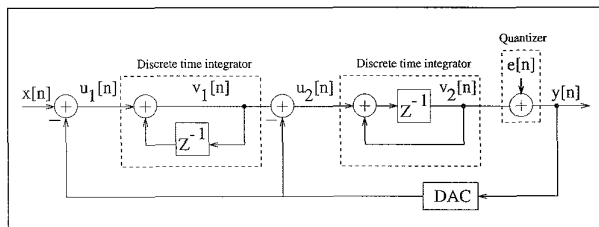
For the special case of  $x = 0$ , the output oscillates between  $V$  and  $-V$  and the output spectrum consists of a pure tone at  $f_s/2$ . More complete results, which are independent of the integrator initial condition,  $v[0]$ , are provided in [9, 10, 12]. In fact, it is shown in [10] and [12] that even if the DC input is an irrational multiple of  $V$ , the quantization noise will not be white and the spectrum at the output of the modulator will be discrete, consisting of tones.

Even for sinusoidal inputs, the quantization error is not white and strong tone components are observed in the output and the strength of the tone distortion components depends on the input amplitudes in a complicated way [14]. The tone structure present at the output of the modulator for very low DC or sinusoidal input amplitudes is often called idle-channel noise.

One other point is worth noting. It can be shown that for comparator output levels of  $\pm V$ , the output of the integrator can have magnitude of at most  $2V$  if the input to the modulator is bounded by  $\pm V$  [14]. This is easily seen from Eqs. 15 and



12. NTF for first, second, and third order sigma-delta modulators. (a) Magnitude spectra on linear scale. For comparison, the oversampled PCM NTF, which has unity gain, is shown; (b) magnitude spectra in dB.



13. Second order sigma-delta modulator.

17, from which we have  $v[n] = x[n-1] - (y[n-1] - v[n-1])$ , which is  $v[n] = x[n-1] - e[n-1]$ . If we assume  $|v[n-1]| \leq 2V$ , then from the transfer characteristic of the 1 bit quantizer (Fig. 2c),  $|e[n-1]| \leq V$ , i.e., the quantizer is not overloaded. We then have,  $|v[n]| = |x[n-1] - e[n-1]| \leq |x[n-1]| + |e[n-1]| \leq V + V = 2V$ . Thus if  $|x| \leq V$  and  $|v[n-1]| \leq 2V$ , then  $|v[n]| \leq 2V$ . This can be guaranteed by ensuring that  $|v[0]| \leq 2V$ , so that  $|v[1]| \leq 2V$ , and so on.

In practice, because of the significant tone structure present at the output of a first order sigma-delta ADC, it is rarely used in applications such as speech or audio, where the presence of such tones is objectionable even if the oversampling ratio,  $f_s/2f_B$ , is high enough to provide a good overall SNR based on the linearized white noise model.

### Higher Order Sigma-Delta Modulation

The fundamental ideas presented can be extended to create sigma-delta architectures in a variety of ways that provide different tradeoffs among resolution, bandwidth, circuit complexity, and modulator stability. Our discussion will include higher order, multi-bit, and multi-stage (cascaded) architectures. In general, to obtain a performance improvement, most of these converters require analog circuits that need to be more complex and precise than those used in the 1st order sigma-delta modulator. Of course, the precision required must still be significantly less than the overall high conversion resolution.

### Second Order Modulation

#### Operation and Performance Modeling

The standard 2nd order sigma-delta modulator A/D is widely used. This modulator realizes  $H_x(z) = z^{-1}$  and  $H_e(z) = (1-z^{-1})^2$ , so that

$$Y(z) = X(z)z^{-1} + E(z)(1-z^{-1})^2 \quad (21)$$

The second order modulator noise transfer function (NTF) is shown in Fig 12 along with the NTFs for the first order modulator and the third order modulator (a logical extension of the second order modulator which will be discussed later).

Note that, compared with the first order sigma-delta NTF, the second order NTF provides more quantization noise suppression over the low frequency signal band, and more amplification of the noise outside the signal band. Compared with a first order sigma-delta, more noise power is pushed outside the signal band.

A block diagram of the modulator is shown in Fig. 13. The structure now contains two integrators. The transfer function of the first one is  $1/1-z^{-1}$  and that of the second one is  $z^{-1}/1-z^{-1}$ . Assuming the modulator output is filtered by an ideal low-pass filter, the linearized white noise model yields the following for the in-band SNR:

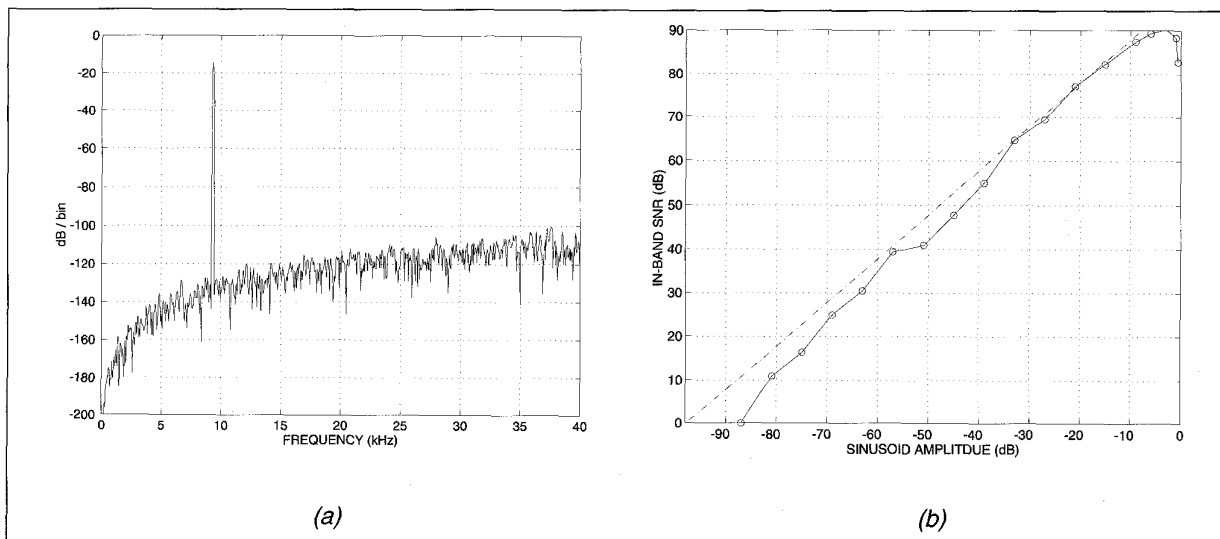
$$\begin{aligned} SNR = & 10 \log(\sigma_x^2) - 10 \log(\sigma_e^2) \\ & - 10 \log\left(\frac{\pi^4}{5}\right) + 50 \log\left(\frac{f_s}{2f_B}\right) \quad (dB) \end{aligned} \quad (22)$$

Again, letting  $f_s/2f_B = 2^r$ , we obtain:

$$\begin{aligned} SNR = & 10 \log(\sigma_x^2) - 10 \log(\sigma_e^2) \\ & - 10 \log\left(\frac{\pi^4}{5}\right) + 15.05r \quad (dB) \end{aligned} \quad (23)$$

Thus, for every increment in  $r$  or for every doubling of the oversampling ratio,  $f_s/2f_B$ , the SNR improves by 15 dB, or the





14. Simulation of a 2nd order modulator (a) power spectrum of the modulator output before decimation; (b) in-band SNR vs amplitude of input sinusoid.

equivalent resolution by 2.5 bits, which is 1 bit better than the improvement achieved by a first order sigma-delta.

Now consider the example used previously where a 20 kHz audio band needs to be converted to a resolution of 16 bits. The  $f_s$  needed by a 2nd order sigma-delta modulator using only a 1 bit quantizer is, from Eq. 22, only 6.12 MHz in contrast with the 96 MHz needed by the 1st order sigma-delta. This circuit speed is very reasonable in current CMOS technology.

Fig 14a shows the low frequency portion (0 to 40 kHz) of an FFT based power spectral density estimate of the output of a 2nd order modulator with a sinusoidal input frequency of 9.3 kHz and  $f_s$  of 6.20 MHz. The tall peak is, of course, the sinusoidal signal. Notice the noise shaping whereby the noise at lower frequencies is greatly attenuated. Such power spectra are often used to numerically calculate the in-band SNR. Here, the in-band SNRs are computed over the 20 kHz audio band. The SNR computation can be repeated for various sinusoid amplitudes to obtain a plot of in-band SNR vs amplitudes. This plot is shown in Fig 14b.

The dashed curve corresponds to the values which are obtained from the linearized white noise model SNR formula (Eq. 22). The discrepancy between the two plots will be discussed later. The dynamic range,  $R$ , can be determined from the in-band SNR vs amplitude plot by looking at the amplitude value for which the SNR is 0 dB. The peak SNR is about 90 dB and  $R$  is about 88 dB which, from Eq. 5, yields an equivalent resolution,  $N$ , of 14.32 bits.

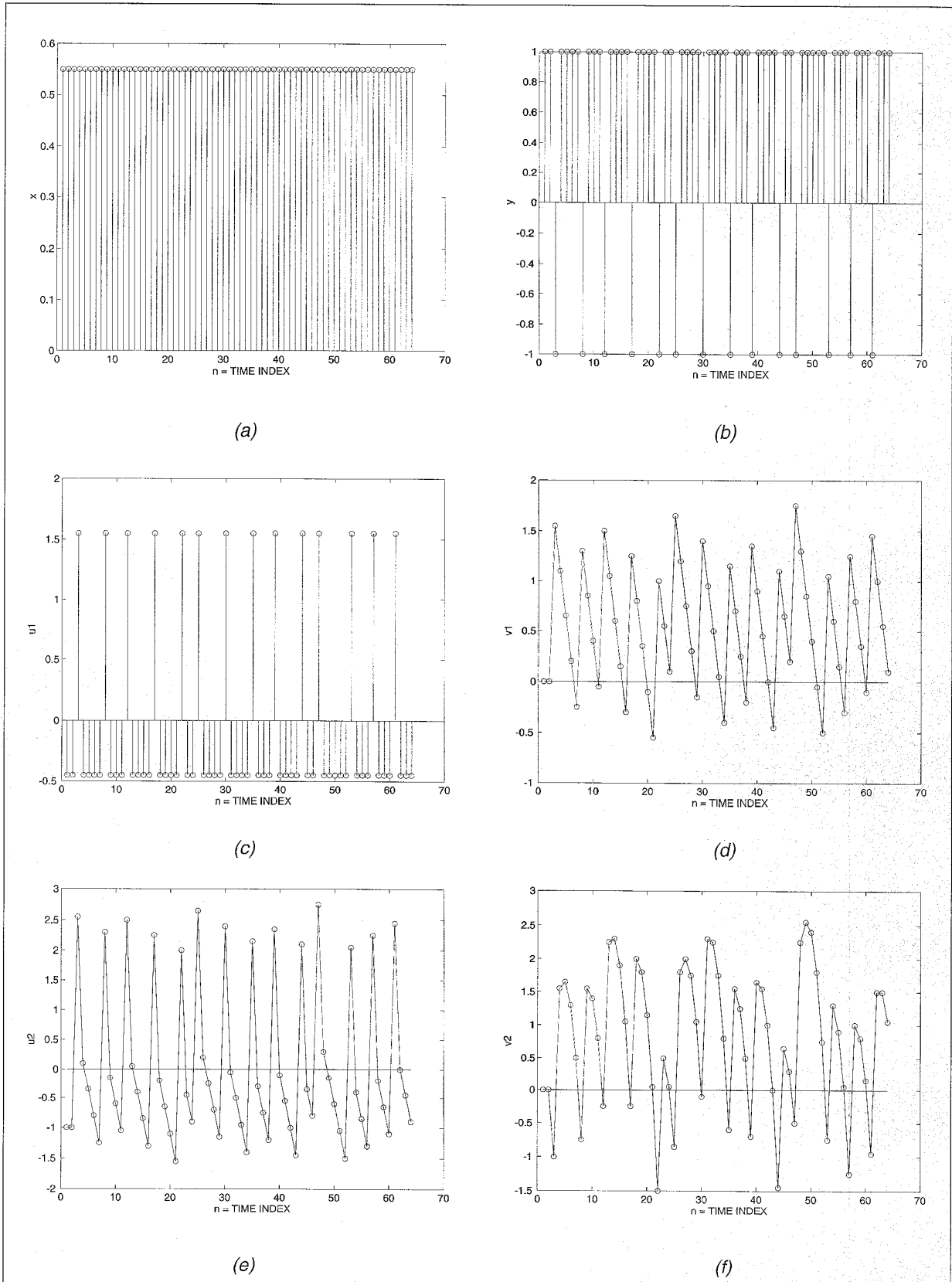
#### Qualitative Time Domain Behavior

In the time domain, referring to Fig. 13, we have,

$$\begin{aligned}
 u_2[n] &= v_1[n] - y[n] \\
 v_2[n] &= u_2[n-1] + v_2[n-1] \\
 y[n] &= \begin{cases} 1 & v_2[n] \geq 0 \\ -1 & v_2[n] < 0 \end{cases} \\
 u_1[n] &= x[n] - y[n] \\
 v_1[n] &= u_1[n] + v_1[n-1]
 \end{aligned}$$

Figures 15a and b show plots of a DC input,  $x[n] = 0.55$ , and the resulting output,  $y[n]$ , of a second order sigma-delta modulator. The input and output of the first integrator are  $u_1[n]$ , and  $v_1[n]$ , respectively, while the input and output of the second integrator are  $u_2[n]$ , and  $v_2[n]$ , respectively. The “error” between the modulator input and output is  $u_1[n]$ , which again is not the quantization error (given by  $e[n] = y[n] - v_2[n]$ ). Looking at  $u_2[n]$ , we see that it is the difference between an integrated (or low-pass filtered) version,  $v_1[n]$ , of the modulator “error,”  $u_1[n]$ , and the output,  $y[n]$ . Thus,  $u_2[n]$  can be considered to be a more fine or accurate version of the modulator error. The signal which is quantized,  $v_2[n]$ , is an integrated version of the “fine error,”  $u_2[n]$ . Consequently,  $u_1[n]$ , and  $v_1[n]$  are analogous to the  $u[n]$  and  $v[n]$  of the 1st order modulator. Outputs  $u_2[n]$  and  $v_2[n]$  are more accurate representations of  $u_1[n]$  and  $v_1[n]$ , and thus produce an output  $y[n]$ , which is more accurate than the output of a 1st order modulator. This should be clear from comparing Fig 15c with Fig 15e and Fig 15d with Fig 15f.

In comparing  $y[n]$  of the second order sigma-delta of Fig. 15b to the  $y[n]$  of the first order sigma-delta of Fig. 10b, the key point is that the distribution of 1’s and -1’s in Fig. 15b is such their average provides a more accurate representation of the input than the corresponding average of the first order modulator output. In other words, for a given block of output samples, the second order modulator uses its allocation of samples more efficiently to represent the input.



15. Second order sigma-delta responding to a DC input: (a) DC input  $x[n] = 0.55$ ; (b) output  $y[n]$ ; (c) "error" signal  $u_1[n]$ ; (d) first integrator output  $v_1[n]$ ; (e) more accurate "error" signal  $u_2[n]$ ; (f) second integrator output  $v_2[n]$ .

### Implementation Imperfections

Compared with a first order sigma-delta modulator, a second order modulator contains an extra integrator. Assuming the same leakage factor,  $\alpha$ , for both integrators, the degraded NTF is approximately  $(1 - \alpha z^{-1})^2$ . The leakage factor of the second order integrators can satisfy somewhat less stringent requirements than that of the first order modulator integrator.

One might expect this since despite the NTF zeros being moved inside the unit circle, there are still two of them providing attenuation of quantization noise. The reduced requirement on the second order modulator integrator leakage factors can also be seen if one calculates the in-band noise power,  $\sigma_e^2$ , from the  $(1 - \alpha z^{-1})^2$  NTF.

The noise term inversely proportional to the oversampling ratio,  $fs/2fB$ , is now divided by  $A4$  rather than  $A2$ , as was the case for the first order modulator in Eq. 18. Consequently, a lower op-amp gain can suffice for the second order modulator.

The other parameter to consider is the gain of the second integrator. From the linearized analysis, the STF and NTF poles are stable for integrator gains up to  $4/3$ , and so still allow a relatively wide variation from this point of view.

Simulations indeed show that the integrator gains are relatively insensitive to deviations from their nominal values over a wide range of oversampling ratios [8].

### Non-linear Behavior

Like the 1st order sigma-delta modulator, a second order modulator may also display limit cycle oscillations [2, 9], and this is easily illustrated in a manner similar to that used for the 1st order modulator. The nature of these limit cycles has been investigated and unlike the first order modulator depend on the initial conditions of the integrator outputs [9, 11].

Most of the exact analyses, e.g., [13-15], which provide an exact description of the spectrum of the quantization error and the modulator output for DC or sinusoidal inputs, have been performed for second order modulators using quantizers with 2 or more bits. In fact, a 1 bit quantizer used in a 2nd order modulator can become overloaded, thereby making the analysis much more difficult. The quantizer is overloaded because the output of the second integrator can significantly exceed values of  $\pm 2V$  [2] even with the modulator input bounded by the quantization levels  $\pm V$ . This is particularly true for large modulator inputs near the quantization levels. However, it has been determined from simulations that a modified 2nd order architecture [5] using a 1 bit quantizer can operate without the integrator outputs having to significantly exceed values of  $\pm 2V$ .

The spectral properties of overload noise have not been theoretically characterized but simulations have been reported in the literature [16, 17]. The simulations demonstrate that the noise significantly manifests itself as harmonic distortion tones for sinusoidal inputs as well as significant tone components near  $f/2$ .

As with the first order sigma-delta, idle channel tones may be observed for small DC or low amplitude tone inputs. According to the linearized model (Eq. 22),  $\sigma_e^2$  is fixed and the SNR should increase linearly with signal power  $\sigma_x^2$ .

However, due to the presence of overload or idle channel tones, the SNR of a second order modulator, using a 1 bit quantizer, increases linearly with signal power only over a certain range of signal power even though the modulator input may be between the quantization levels  $\pm V$ . More and more overload noise power is produced with increasing input values. Consequently, above a certain value of high signal power, the SNR will actually start to decrease when the increase in overload noise power dominates the increase in signal power.

This can be seen in Fig 14b for input amplitudes greater than about -5 dB. On the other hand, as the input power becomes low, the SNR decrease is caused both by the decrease in signal power and by the presence of idle channel tone noise in the signal band. This can be clearly be seen in Fig. 14 for low amplitude values.

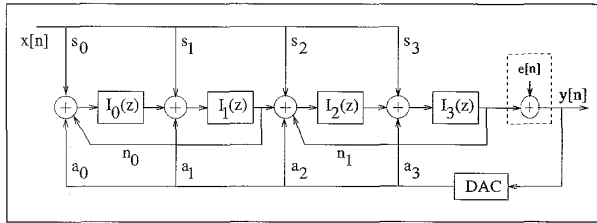
According to Eq. 22 based on the linearized white noise model, the peak SNR and dynamic range should have been about 98 dB. Thus, idle channel tone problems, resulted in a 10 dB degradation of the dynamic range predicted by the linearized model while overload noise prevented the modulator from reaching the peak SNR predicted by the model. Dithering techniques [17] will often break up tone structures including overload and idle channel tones, thereby producing a smoother power spectrum output, a more linear dependence of SNR on signal power, and a dynamic range, which is closer to the value predicted by Eq. 22.

Another factor which affects the in-band SNR is the sinusoid input frequency. This is particularly true if the sinusoid amplitude is such that strong harmonic distortion components are produced in the power spectrum output. In this case, the choice of a higher input frequency will result in these harmonics falling outside the 20 kHz bandwidth and not contributing to the in-band SNR. On the other hand, the choice of a lower input frequency will yield poorer values of in-band SNR because the harmonics will now fall inside the 20 kHz signal band.

### Other Types of Higher Order Sigma-Delta Modulation

Sigma-delta converters realizing higher order NTFs achieve even higher resolution by pushing even more noise power outside the signal band. Alternatively, a lower sampling rate can be used to achieve the same resolution for a given signal bandwidth. In this case, the speed requirements on the analog hardware is relaxed.

An order  $L$  modulator based on a straightforward extension of the first order sigma-delta realizes a STF given by  $H_x = z^{-1}$  and a NTF given by  $H_e = (1 - z^{-1})^L$ , which contains  $L$  zeros at  $z = 1$  or at DC frequency on the unit circle. A third order modulator structure can be created from the second order structure of Fig. 13 by inserting an integrator with transfer function  $1/(1 - z^{-1})$  between the the summing node of the modulator input and the first integrator of the second order modulator. The input to this new integrator is now  $x[n] - y[n]$  and the output of this integrator minus  $y[n]$  is fed as the input to



16. An example of a fourth order modulator topology.

what used to be the first integrator of the second order modulator.

The magnitude spectra for a third ( $L=3$ ) order NTF plotted on a linear scale and in dB are shown in Figs. 12a and b. Note that over the signal band, which is  $0.05 f_s$  in the figure, the 3rd order NTF provides more attenuation of the quantization noise than the second or first order NTFs and so is capable of pushing more noise power outside the signal band than the second or first order modulators. The ideal in-band SNR achieved by an  $L$ th order modulator is given by

$$SNR = 10 \log(\sigma_x^2) - 10 \log(\sigma_e^2) - 10 \log\left(\frac{\pi^2 L}{2L+1}\right) + (20L+10) \log\left(\frac{f_s}{2f_B}\right) \quad (24)$$

Thus, for every doubling of the oversampling ratio, this modulator provides an extra  $(6L+3)$  dB of SNR, or an extra  $(L+1/2)$  bits of resolution. For the CD example used throughout this article, a 3rd order modulator will need a  $f_s$  of 1.92 MHz to convert a 20 kHz band to 16 bits of resolution.

Modulator topologies are not constrained to always realize FIR NTFs as we have seen until now or to realize all the NTF zeros which provide attenuation at DC frequency. In fact, distributing the zeros over the signal band rather than placing them all at DC frequency can be more efficient in pushing quantization noise outside the signal band.

Examples of higher order topologies are described in [18-21]. One such fourth order topology described in [19] is shown in Fig 16, where  $I_k(z)$  denotes the  $k$ th integrator. This structure realizes Eq. 7 in the form

$$Y(z) = X(z) \frac{B_x(z)}{A(z)} + E(z) \frac{B_e(z)}{A(z)} \quad (25)$$

Note that the STF and NTF are IIR transfer functions in this case. The feedforward coefficients  $s_k$  realize  $B_x$ , which contains the  $Z$  domain zeros of the STF. The feedback coefficients  $n_k$  between every second integrator realize  $B_e$ , which contains the zeros of the NTF. Finally, the feedback coefficients  $a_k$  realize  $A$ , which contains the poles of both the STF and the NTF.

In order to implement Eq. 25, based on a sigma-delta modulator that incorporates feedback, the  $z^0$  coefficient in  $B_e$  and  $A$  must be equal to ensure a causal feedback loop (in other words to avoid non-computable delays, [3, pp. 308-309]). If the NTF is FIR, i.e., if  $A(z) = 1$ , then the causality constraint will require the  $z^0$  coefficient of  $B(z)$  to be 1.

Note that the performance will be limited by the degree to which the analog coefficients  $a_k$ ,  $s_k$ , and  $n_k$ , match their desired values. For the modulator to be useful, the degree of required matching should be significantly less than the overall resolution of the converter.

Higher order architectures also alleviate some of the tone problems mentioned earlier [2, 18]. The main difficulty with such higher order structures is that such modulators are only conditionally stable when a one bit quantizer is used. Stability may for example depend on the input signal being kept below a certain value or on precise circuit matching needed to satisfy a stability criterion. Stability is often described in the sense of the quantizer not being overloaded. This is useful because any higher order modulator structure can be transformed into an equivalent modulator in the form of the 1st order modulator of Fig 8 with the integrator being replaced by a general "loop filter"  $H(z)$ . If the quantizer is not overloaded, its input is bounded by  $\pm 2V$ , and this implies that the loop filter is also operating in a stable manner and all internal signals will be bounded. Stability is, in general, difficult to determine for a modulator using an 1 bit quantizer. One reason is the difficulty in characterizing the gain of the 1 bit quantizer. The gain of the 1 bit quantizer of Fig. 2c is variable—it depends on the quantizer input. The smaller the quantizer input, the larger is its gain (e.g., if the input is zero, the output is  $V$  and the gain is infinite). The larger the quantizer input, the smaller is its gain (e.g., if the input is infinite, the output is  $V$  and the gain is zero). Another way of looking at this is to observe that if one attempts to linearize the quantizer transfer curve of Fig. 2c by trying to fit a straight line to the curve, the correct slope of the straight line is arbitrary.

As we have seen earlier, the input to the quantizer clearly changes with time, even if the input to the modulator is a DC signal. The manner in which the quantizer gain changes over time will also depend on the type of input applied to the modulator. Consequently, performing a linear system analysis of the modulator signal and noise transfer functions in terms of modulator parameters is inadequate because the poles of the transfer function, which determine stability, depend on a time varying and input dependent quantizer gain. An attempt to characterize the quantizer gain more accurately for DC and sinusoidal inputs has been made for several modulators [22].

The phenomenon of limit cycle oscillations is also connected to stability. This is because the structure of limit cycles may be such that the amplitude of internal modulator variables is large, causing the quantizer to overload. The frequency of such a deleterious limit cycle oscillation can correspond to the point on the unit circle where the modulator transfer function pole crosses the unit circle into the unstable region. Properties of limit cycles in the context of stability have been investigated in [11]. A limit cycle that corresponds to the modulator transfer function pole moving from the inside to the outside of the unit circle may not necessarily result in unstable behavior in the long term, provided integrator outputs do not saturate before stability is restored [23].

Suppose an unstable limit cycle, corresponding to poles moving outside the unit circle, results when the quantizer gain is too high, i.e., when the input to the quantizer is small. If this is the case, growing signal values in the modulator which result from this instability will eventually increase the input to the quantizer thereby reducing its gain and so moving the poles back inside the unit circle [23].

One way to guarantee stability would be to reset the integrators if it was detected, by additional circuitry, that their values were becoming too large. However, this approach may cause a significant decrease in the SNR [24]. Similarly, allowing integrator outputs to clip or saturate may also cause degradations in the SNR performance. In particular, low frequency limit cycles which may introduce distortion components in the signal band may occur (albeit with reduced values of integrator output) and may persist for a long time thereby reducing the SNR [23]. An alternative approach uses local feedback loops in an attempt to gracefully return integrator outputs to their normal operating region [24]. Since the linearized transfer function of the system is modified due to the local feedback loops, the effect of these loops is cancelled in the digital domain.

Several quantitative criteria have also been proposed to characterize stability. The  $l_1$  norm criterion [14, 25, 26] relates the sum of the magnitudes of the modulator NTF impulse response coefficients, the number of bits in the quantizer, and the modulator input level to the no overload stability requirement. This is a sufficient but not necessary condition for stability, and it has unfortunately been found to be too conservative for practical use. An ad-hoc stability criterion which has been proposed [18] and found to be useful [21] is to design the NTF to possess less than 2 to 6 dB of out-of-band gain.

## Multi-bit Sigma-Delta Modulation

Until now, we have assumed that the quantizer and DAC inside our sigma-delta modulator were 1 bit devices. However, converters using a multi-bit internal quantizer offer more potential resolution from the internal quantizer. A 2nd order multi-bit sigma-delta converter would look exactly the same as the modulator shown in Fig. 13, except that  $e[n]$  in the figure would be the model for an  $N$  bit quantizer instead of a 1 bit quantizer, and the DAC would be an  $N$  bit DAC instead of a 1 bit DAC. The use of a multi-bit quantizer affects the  $\sigma_e^2$  term in the expressions for the SNR, where each additional bit used in the quantizer will yield a 6 dB improvement in the SNR. Using Eq. 3 without the approximation, it is easy to see that if a 5 bit internal quantizer is used instead of a 1 bit quantizer, a 30 dB improvement in SNR is possible. Alternatively, the sampling frequency can be reduced by a factor of 4, while keeping the resolution the same. For our CD example, a 2nd order modulator using a 5 bit internal quantizer can use a  $f_s$  of 1.53 MHz, rather than the  $f_s$  of 6.12 MHz needed by a modulator using a 1 bit quantizer.

The behavior of multi-bit sigma-delta systems more closely follow that predicted by the linearized model (in the

extreme case, if the quantizer has an infinite number of bits, there is no non-linearity). Consequently, the stability of higher order modulators using multi-bit quantizers is generally more accurately predicted. Another way of viewing enhanced stability is to consider the gain of the multi-bit quantizer. If a midtread multi-bit quantizer is used, its gain is relatively close to one for most output values (even for zero input, a midtread quantizer will have a zero output and hence unity gain unlike the midriser 1 bit quantizer, which has infinite gain for a zero input). This is because a straight line drawn through the multi-bit quantizer transfer characteristic (e.g., Fig. 2b) can no longer be arbitrarily drawn. Of course, if for any reason the quantizer does start to overload, its gain will start to deviate more and more from unity. Even though the  $l_1$  stability criterion mentioned earlier may be too conservative, it may allow one to obtain an initial idea about the number of bits needed for stable operation for a given NTF and input signal level.

Modulators using multi-bit quantizers also display less of the tone problems associated with the 1st and 2nd order sigma-delta converters using a 1 bit internal quantizer. The main disadvantage is that the multi-bit DAC cannot be easily fabricated in VLSI with sufficient linearity needed for high resolution conversion. Various techniques, examples of which can be found in [27-33], have been proposed to reduce the linearity required for the DAC. The multi-bit output also complicates the digital low-pass filter hardware following the modulator, because for multi-bit processing, the filter requires multi-bit hardware multipliers.

## Multi-stage (Cascaded) Sigma-Delta Modulation

Higher order NTFs can also be created by cascading independent modulator stages. This cascading does not adversely affect the stability of the overall modulator, provided the individual stages are stable. They may also suffer fewer of the tone problems than a first or second order sigma-delta alone [2, 14].

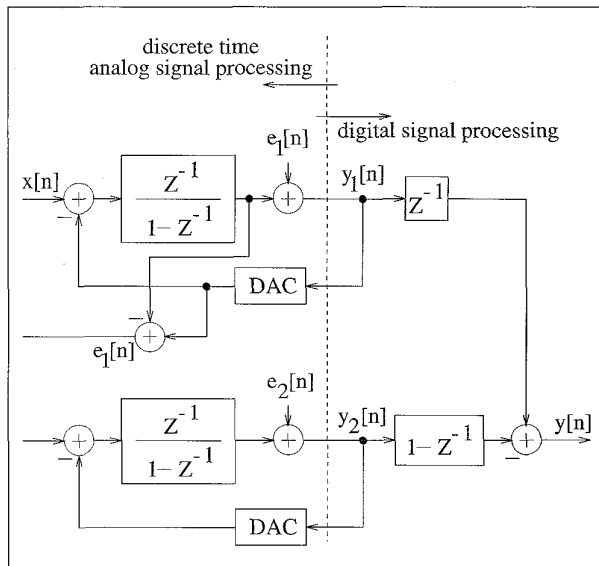
An example of a second order modulator obtained by cascading two 1st order modulators is shown in Fig 17. The signal  $x[n]$  is the input to the first modulator in the cascade but the quantization error,  $e_1[n]$ , of the first modulator is used as the input to the second modulator. Finally, the outputs of the first stage and second stage modulators are added in the digital domain after passing through a digital delay ( $z^{-1}$ ) and a digital differentiator ( $1-z^{-1}$ ), respectively.

From the  $Z$  domain analysis of the linear system model with the DACs replaced by unity gains, we have,

$$Y_1(z) = X(z)z^{-1} + E_1(z)(1-z^{-1}) \quad (26)$$

$$Y_2(z) = E_1(z)z^{-1} + E_2(z)(1-z^{-1}) \quad (27)$$

The output is computed as  $Y_1 z^{-1} - Y_2(1-z^{-1})$ . This sum results in a cancellation of the first order noise term  $E_1(z)$  to produce the overall output,



17. "1-1" cascade: a 2nd order modulator from a cascade of two 1st order modulators.

$$Y(z) = X(z)z^{-2} - E_2(z)(1 - z^{-1})^2 \quad (28)$$

Except for the sign on the noise that is irrelevant and an extra delay experienced by the input, the modulator realizes the same output as the standard second order sigma-delta modulator. One advantage of using this structure over the second order modulator is the fact that the quantizer in either of the first order modulator sections will never overload for  $x[n]$  bound by  $\pm V$ . However, the cascaded structure requires matching between the analog and digital transfer functions as well matching among the D/A output levels among various stages [2, 34]. In fact, mismatch effects and integrator leakage can lead to the propagation of unshaped or poorly shaped noise from an earlier section to the final output [36, 40].

Assume there are circuit imperfections in the "1-1" cascade of Fig. 17, such that the transfer function of the integrator in the first section is  $g z^{-1} / (1 - \alpha z^{-1})$  instead of  $z^{-1} / (1 - z^{-1})$ . Even if we assume the integrator in the second stage is ideal, the output is then

$$Y(z) = \frac{X(z)gz^{-2}}{1 - (g - \alpha)z^{-1}} + \frac{(1 - \alpha)z^{-2}E_1(z)}{1 - (g - \alpha)z^{-1}} + \frac{(g - \alpha)z^{-2}E_1(z)(1 - z^{-1})}{1 - (g - \alpha)z^{-1}} - E_2(z)(1 - z^{-1})^2 \quad (29)$$

The first term of the equation contains the signal, which is no longer a pure delay but will have a ripple to it determined by the factor  $g/(1 - (g - \alpha)z^{-1})$ . However, since the signal is oversampled, this additional signal transfer function will mostly be flat at lower baseband frequencies. Ignoring the ripple due to the factor  $1/(1 - (g - \alpha)z^{-1})$ , the second term in Eq. 29 is the unshaped noise from the first stage, the third term in Eq. 29 is the first order shaped noise from the first stage, and the fourth term in Eq. 29 is the desired second order shaped noise term. Clearly, for large imperfections, the unshaped noise

term might dominate the noise term subject to second order noise shaping. If more than two stages are cascaded, the cumulative effects of such quantization error leakage effects will yield diminishing returns in performance improvement.

Architectures using only first order modulators have been realized [35], as have architectures using second order modulators [37-39]. A comparison of some architectures can also be found in [40]. Finally, note that due to the addition of various single bit intermediate outputs, the architecture has a multi-bit final output, which complicates the decimation filter hardware.

## Band-pass Sigma-Delta Systems

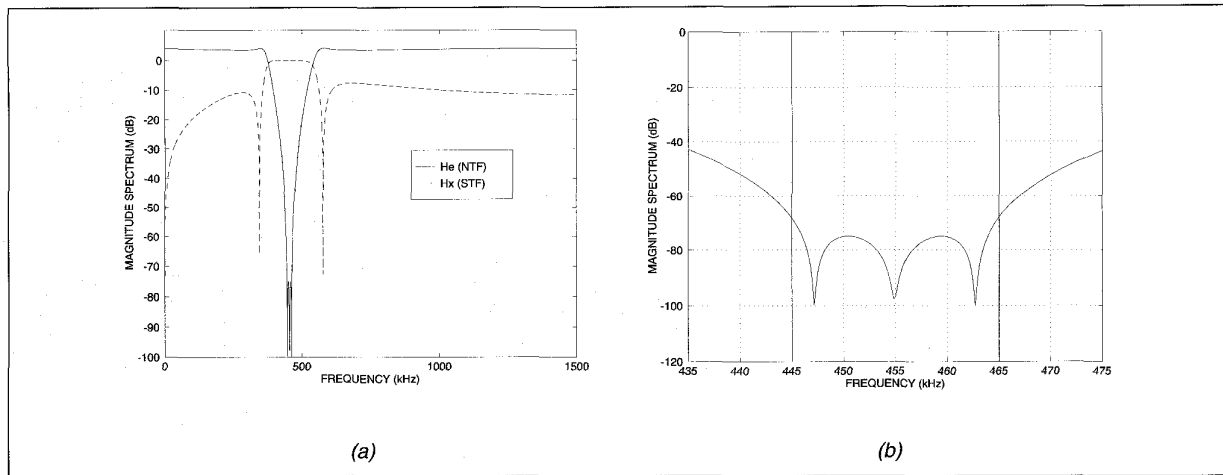
Thus far, we have assumed that the sampling frequency  $f_s$  is much greater than the Nyquist rate, which is twice the highest frequency component in the input signal. For low-pass signals, the highest frequency component is also the signal bandwidth  $f_B$ . If a signal with bandwidth  $f_B$  is band-pass but is located at a center frequency,  $f_c$ , its highest frequency is  $f_c + f_B/2$ . If  $f_c$  is large, choosing  $f_s$  to be much greater than the highest frequency will lead to an unreasonably large  $f_s$ , and does not take advantage of the band-pass nature of the signal.

Band-pass sigma-delta modulation [41] allows high resolution conversion of band-pass signals if  $f_s$  is much greater than the signal bandwidth  $f_B$ , rather than the highest signal frequency. Band-pass sigma-delta modulators can be used in AM digital radios [41] or receivers for digital cellular mobile radios [45].

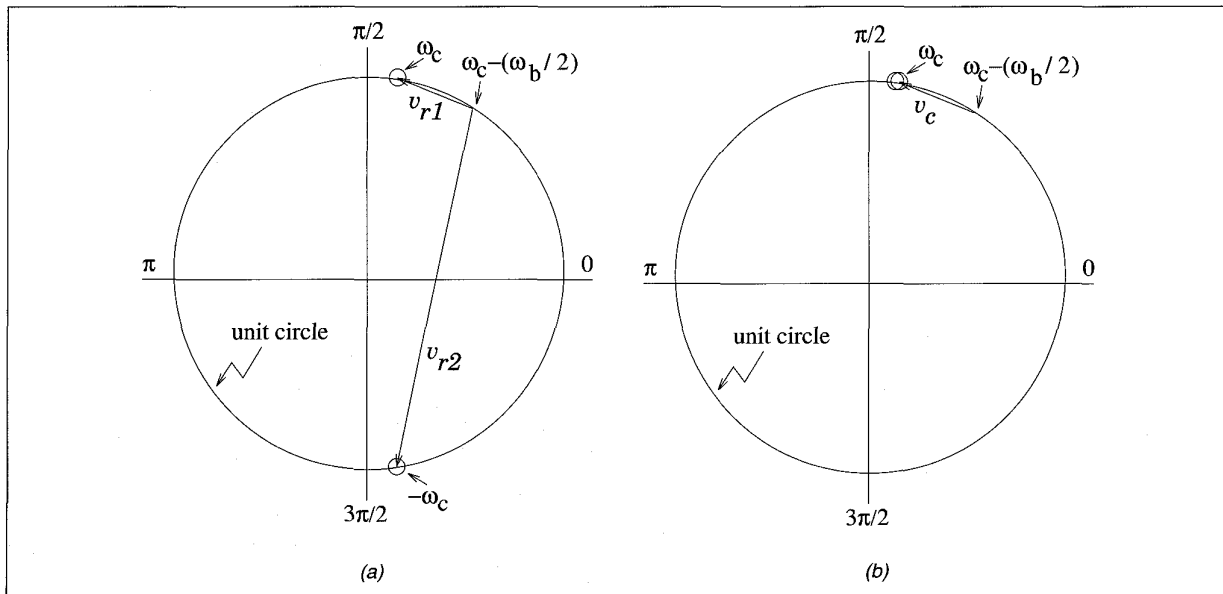
## Real Band-pass Modulation

Unlike low-pass sigma-delta modulators, which realize NTF zeros at DC or low frequencies on the unit circle of the Z plane, band-pass modulators have NTFs that realize zeros or notches in the signal band of interest,  $[f_c - f_B/2, f_c + f_B/2]$ . Consequently, quantization noise that occurs over the signal band is attenuated, and noise power is pushed outside this band. Regardless of where the signal band is centered, the smaller the signal band,  $f_B$ , relative to the sampling frequency,  $f_s$ , there is less in-band noise power for a given NTF. Noise outside the signal band can be attenuated with a digital decimation filter and so high resolution conversion is possible for large oversampling ratios  $f_s/2f_B$ . The modulator STF and the decimation filter will typically have a band-pass characteristic, providing unity gain over the signal band.

As an example, the NTF and STF magnitude spectra for the design in [42] are shown in Fig 18a. In this example, the signal band has a center frequency of  $f_c = 455$  kHz,  $f_s = 3$  MHz,  $f_B = 20$  kHz, so the oversampling ratio is 75. Figure 18a shows the magnitude spectra in dB of  $H_x$ , and  $H_e$ . Figure 18b shows a closeup view of the magnitude spectrum in dB of  $H_e$  over the 435 kHz to 475 kHz region. The vertical bars delineate the 20 kHz signal band centered at 455 kHz. The NTF is sixth order, and as should be clear from Fig. 18b, contains three notches or zeros, which provide attenuation of the quantization noise over the 20 kHz signal bandwidth (the other three



18. Band-pass sigma-delta: (a) magnitude spectra in dB of  $H_x$  (the STF, dashed plot), and  $H_e$  (the NTF); (b) closeup of the magnitude spectrum in dB of  $H_e$ .



19. Z plane zeros for (a) second order real FIR NTF; (b) second order complex FIR NTF.

zeros are complex conjugates of these). The STF is band-pass and has minimal ripple and is approximately linear phase in the signal band. Band-pass converters employing  $L$ th order modulators display a SNR performance that improves at the rate of  $(3L + 3)$  dB per octave increment with the oversampling ratio  $f_s/2f_B$  [42].

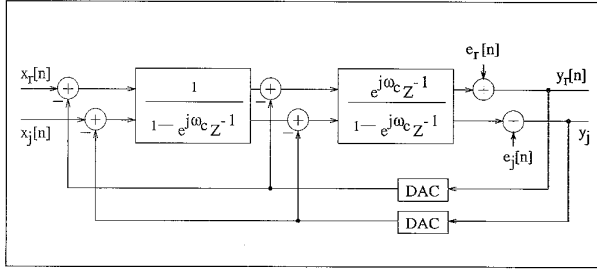
### Complex Band-pass Modulation

Most band-pass sigma-delta converters [43-45] use real NTFs. Thus, all the coefficients in the Z domain transfer function are real. Let us now discuss the idea of complex band-pass NTFs, which have been proposed independently in [46] and [47]. The use of complex NTFs can improve the resolution that can be obtained for real band-pass signals [47]. The reason for this improvement is best illustrated through an example using second order FIR NTFs.

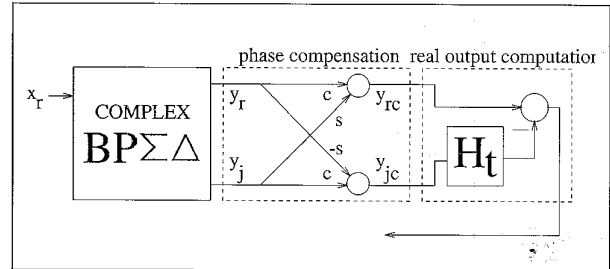
Figure 19a shows the unit circle on the Z plane containing zeros for a real second order FIR NTF. We find it more convenient to use the discrete time frequency  $\omega = 2\pi f/f_s$  instead of  $f$  in our discussion. A real NTF realizing a zero at center frequency  $\omega = \omega_c$  is constrained to also realize one at  $-\omega_c$ .

For a signal bandwidth  $\omega_b$  consider the attenuation of the NTF at half the bandwidth away from the zero, that is, at  $\omega_c - \omega_b/2$ . This attenuation is given by the product of the magnitude of the vectors  $v_{r1}$  and  $v_{r2}$ , i.e., by  $|v_{r1}| \times |v_{r2}|$ . The smaller this product, the better the attenuation provided by the NTF, and the better the quantization noise suppression over the signal band, hence a better SNR will be obtained.

Vector  $v_{r1}$  represents the contribution from the zero at  $\omega_c$ , and  $v_{r2}$  represents the contribution from the zero at  $-\omega_c$ . In the figure, the distance between  $\omega_c$  and  $\omega_c - \omega_b/2$  has been exaggerated—in practice, high resolution can be obtained only if



20. Complex second order modulator.



21. Complex BP sigma-delta modulation of a real input signal followed by phase compensation and computation of a real output.

the signal is narrowband and this distance is small. As the center frequency  $\omega_c$  increases,  $\nu_{r2}$  increases as well, and the attenuation provided by the NTF becomes worse and so there will be less quantization noise suppression.

For the complex case, the zeros of a second order FIR NTF with both zeros located at a center frequency  $\omega_c$ , are shown in Fig. 19b. The attenuation at  $\omega_c - \omega_b/2$  is given by the quantity  $|v_r| \times |v_c|$ , which depends only on  $\omega_b$  and not on  $\omega_c$ , as there is no influence from a zero at  $-\omega_c$ . Consequently, the attenuation provided by the complex NTFs does not suffer any degradation with increasing center frequency. Therefore, at higher center frequencies, complex NTFs can provide more attenuation in the signal band, i.e., better quantization noise suppression and so a better SNR than a real transfer function. Complex band-pass converters employing  $L$ th order modulators display a SNR performance which increases at the rate of  $(6L + 3)$  dB per octave increment in the the oversampling ratio  $f_s/2f_b$ . This is in contrast to the  $(3L + 3)$  dB rate of improvement for  $L$ th order real band-pass modulators.

Having discussed the complex NTF, we now discuss the implementation of a complex 2nd order modulator with a band centered at  $\omega_c$ . Such a modulator can be generated by modulating the  $Z$  domain NTFs and STF of a standard 2nd order modulator with the transformation  $z^{-1} \rightarrow z^{-1} \exp(j\omega_c) = z^{-1} \exp(j 2\pi f_c / f_s)$ . A block diagram of such a system obtained from Fig. 13 with the above transformation is shown in Fig. 20. Notice, for example, that the integrator  $z^{-1}/(1 - z^{-1})$  of Fig. 13 is replaced with the integrator  $(z^{-1} e^{j\omega_c})/(1 - e^{j\omega_c} z^{-1})$ , which will have complex inputs and outputs consisting of real and imaginary parts. Such a complex integrator can be physically realized in switched capacitor technology using two cross coupled integrators [48]. Note there also need to be two physical quantizers,  $E_r$  and  $E_j$ , one for the "real" channel and the other for the "imaginary" channel. The  $Z$  domain output of the complex modulator is given by

$$Y(z) = [X_r(z) + jX_j(z)]z^{-1} \exp(j\omega_c) + [E_r(z) + jE_j(z)][1 - z^{-1} \exp(j\omega_c)]^2 \quad (30)$$

The STF is no longer a pure delay but contains the phase factor  $e^{j\omega_c}$ . However, this phase factor can be compensated digitally by multiplying the output of the complex modulator with the complex constant  $e^{-j\omega_c}$  to obtain the phase compensated output. The phase compensated output is  $Y_c(z) =$

$Y(z)e^{-j\omega_c}$ . For a real input signal, we have  $x_i[n] = 0$  and  $X_j(z) = 0$  so that  $x[n] = x_r[n]$  and  $X(z) = X_r(z)$ . In this case,

$$Y_c(z) = X^r(z)z^{-1} + [E_r(z) + jE_j(z)] \exp(-j\omega_c)[1 - z^{-1} \exp(j\omega_c)]^2 \quad (31)$$

where  $Y_c(z) = Y_{rc}(z) + j Y_{jc}(z)$ .

$Y_c(z)$  consists of a signal term  $X_r(z) z^{-1}$ , which we have assumed results from a real signal, plus the shaped quantization noise term. From Eq. 31, we can also see that in the time domain, the signal component in  $y_c[n]$  is  $x_r[n-1]$ . However,  $y_c[n]$  will still be complex because of the complex noise and, accordingly, the spectrum of  $y_c[n] = y_{rc}[n] + j y_{jc}[n]$  will not be symmetric. A real output can be obtained without disturbing the signal or altering the SNR by considering the final output to be  $y_{rc}[n] - y_{jc}[n]^* h_t[n]$  where  $h_t$  is the impulse response of an ideal Hilbert transformer,  $H_t(z)$ . In the  $Z$  domain, the final output is then  $[Y_{rc}(z) - Y_{jc}(z)H_t(z)]$ . This operation amounts to keeping only positive frequencies (discarding negative frequencies) with the Hilbert transformer and then taking the real part to make the spectrum symmetric by folding the positive frequencies on to the negative frequency axis.

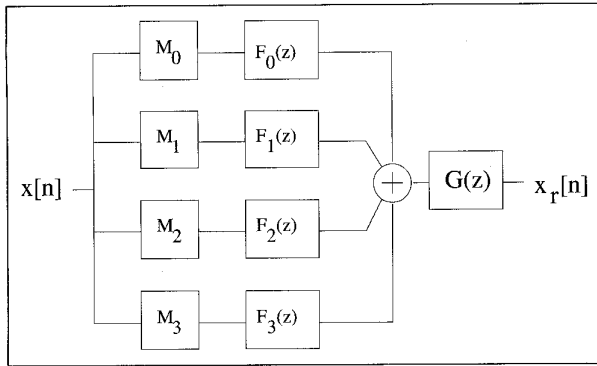
The phase compensation and computation of the real output are shown in Fig. 21 where  $c = \cos(\omega_c) = \cos(2\pi f_c / f_s)$  and  $s = \sin(\omega_c) = \sin(2\pi f_c / f_s)$ . Decimation of the sigma-delta output and demodulation of the band-pass signal to baseband are not shown here.

Note that in the case of a real input signal, the imaginary channel of the modulator is not directly connected to the input signal. One could feed  $x_r[n]$  into the imaginary channel input as well. This has the benefit of resulting in a  $\sqrt{2}$  gain of the STF which in principle will result in a 3 dB improvement in signal power. However, simulations show that with the input being fed to the imaginary channel, the quantizers overload much more often and the increase in quantization noise does not merit the gain in the signal power.

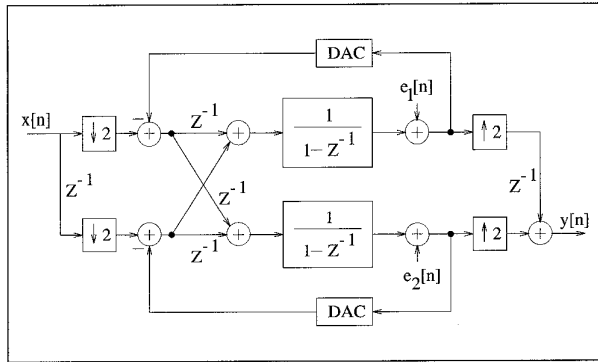
## Parallel Sigma-Delta Systems

The use of parallelism for PCM A/D conversion has been considered in [49] and [50]. This section very briefly discusses several schemes that use architectural parallelism to improve the performance of sigma-delta modulators A/D





22. Multi-band sigma-delta system architecture for  $P = 4$  channels.



23. Two channel time interleaved conversion using 1st order sigma-delta modulators.

converters. For a given signal bandwidth, modulator order, and sampling frequency, these architectures can attain higher resolution. The cost of realizing the improved performance lies clearly in the extra hardware needed for each parallel channel.

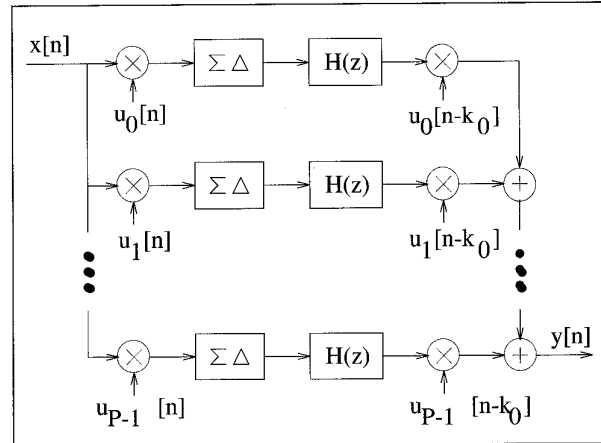
### Multi-band Sigma-Delta Modulation

One architecture [51] uses modulators that realize different band reject NTFs for different portions of the signal band. Each band (channel) is converted in parallel. A bank of FIR filters attenuates the out-of-band noise for each band and can achieve perfect reconstruction of the signal component assuming that the modulator STF is a simple delay [52].

A block diagram of the system architecture for four channels is shown in Fig. 22, where  $M_k$  denote the modulators, and  $F_k(z)$  and  $G(z)$  comprise the digital filter bank. Using  $L$ th order complex band-pass modulators, assuming equal sized bands, and assuming the quantization errors from the different channels to be mutually uncorrelated, the total in-band noise power,  $\sigma_{er}^2$ , at the output of the reconstructed signal is

$$\sigma_{er}^2 = \frac{4\sigma_e^2}{2^{2L+1}} \frac{\pi^{2L}}{2L+1} \left( \frac{2f_c}{f_s} \right)^{2L+1} \left( \frac{1}{P} \right)^{2L} \quad (32)$$

where  $f_B$  is the total signal bandwidth converted,  $f_c = f_B/P$  is the bandwidth per channel,  $P$  is the number of channels, and



24. Hadamard modulated sigma-delta A/D conversion system.

$\sigma_e^2$  is the quantizer noise power in any one modulator. The SNR improves at a rate of  $(6L + 3)$  dB, or the resolution by  $(L + 1/2)$  bits per octave increment in the oversampling ratio per channel,  $f_s/2f_c$ . The SNR and resolution also improve at rates of  $(6L)$  dB, and  $L$  bits, respectively, per octave increment in the number of channels,  $P$ .

### Time Interleaved Sigma-Delta Modulation

Another method for incorporating parallelism into sigma-delta converters is through time interleaving [53]. This architecture employs ideas of block filter theory to use  $P$  identical, mutually cross coupled, modulators running at a sampling rate  $f_s$  to generate the same modulator transfer function, which runs at an equivalent sampling rate of  $Pf_s$ . The block diagram for a two channel system using 1st order modulators running in each individual channel is shown in Fig 23. Note that downsampling the signal can result in aliasing, but the choice of the cross coupling terms ensure that the aliasing is cancelled in the final output for the ideal system.

If  $L$ th order low-pass sigma-delta modulators are used in each channel, the ideal in-band SNR improves by  $(6L + 3)$  dB, or the resolution by  $L + 1/2$  bits per octave increment in the number of channels,  $P$ , since each octave increment in  $P$  amounts to an octave increment in the oversampling ratio of the modulators.

### Hadamard System

Another parallel sigma-delta system has been described in the literature [54] recently. Here, each channel contains a Hadamard modulator, which multiplies the input signal by a  $\pm 1$  sequence,  $u_k[n]$ . This operation is called Hadamard modulation.

The Hadamard sequences are obtained from repeating the rows of the Hadamard transform matrix. The Hadamard modulated sequence of each channel is then quantized using a standard sigma-delta modulator. The output of each sigma-delta modulator is filtered to attenuate out-of-band noise, and

again multiplied by a Hadamard sequence before all the channel outputs are added to provide the final output.

The block diagram of the system is shown in Fig 24, where  $k_0$  is assumed to be the delay in the STF of the sigma-delta modulators. The overall effect of the system on the signal is to filter it with a subset of the filter coefficients of  $H(z)$ . The relevant  $H(z)$  coefficients affecting the signal can be chosen such that the STF experiences a delay, while the other coefficients can be chosen to maximize the quantization noise attenuation.

The quantization error does not see the first Hadamard sequence and the effect of the second Hadamard sequence modulation is to frequency shift the filtered quantization error power spectral density. Using  $L$ th order low-pass sigma-delta modulators, the SNR using this approach improves by  $(6L)$  dB or the resolution improves by  $L$  bits per octave increment in the number of channels  $P$ .

### Applications Using Actual Sigma-Delta Converters

We now present some applications using actual sigma-delta converters that have been fabricated in VLSI. The purpose here is to provide a sense of the final performance achieved by the converters as to resolution and bandwidth, rather than to compare them with respect to theoretical performance or to the multitude of other performance criterion (power, area, topology, technology, etc.). To aid the resolution/bandwidth evaluation, we use the abbreviation *osr* for the oversampling ratio,  $f_s/2f_B$ , where  $f_B$  is the signal bandwidth and  $f_s$  is the sampling frequency. To provide an overview of the converters discussed here, we have included all the converters in a common format in the Table.

Data conversion for instrumentation applications may require resolutions up to 19-20 bits, albeit at low bandwidths. One such converter used in instrumentation transducers [55] uses a 5th order modulator topology with a  $f_s = 128$  kHz ( $osr = 128$ ) to achieve a 118 dB dynamic range, or more than 19 bits of resolution over a 492 Hz bandwidth. Similarly, a

converter that is used for seismic activity measurements uses a 128 kHz sampling rate with an *osr* of 128, achieving more than 120 dB peak SNR, or almost 20 bits of resolution over about a 500 Hz bandwidth [56]. A 4th order topology is used. Note that for extremely high resolution such as reported here, the quantization noise floor approaches the level of circuit noise for state of the art technologies. Thus, very careful circuit design and optimization is required to fully take advantage of the potential performance that can be realized by the sigma-delta ADC.

Sigma-delta converters are good candidates for voiceband (speech) applications where the signal bandwidth is 4 kHz and 13-14 bits of resolution is desirable. One such converter [57] actually used a single bit 1st order modulator with  $f_s = 4$  MHz, or an *osr* of about 500 to achieve a 79 dB dynamic range, i.e., about 13 bits of resolution. Dithering was required to alleviate the tone problem associated with the first order modulator. Another converter [58] also achieved 13 bits resolution using a  $f_s = 1.024$  MHz with an *osr* of 128. The modulator used is a standard second order modulator employing a 1 bit quantizer.

Digital audio applications such Hi-Fi CD and DAT systems often use sigma-delta A/D converters. Consumer quality Hi-Fi audio needs to be digitized at 16 bits of resolution, and audiophiles prefer up to 18-20 bits of resolution. Many converters have been reported in the literature for audio bandwidths of 20-24 kHz. A high resolution audio range converter was reported as early as 1986 and achieved a dynamic range of 106 dB, or almost 18 bits of resolution over a 24 kHz bandwidth using a fourth order modulator and 4 bit internal A/D and D/A converters [59]. The sampling frequency used was 6.144 MHz with an *osr* of 128. Another converter that also uses a fourth order modulator to convert a 24 kHz bandwidth achieves near 16 bit dynamic range using single bit quantizers with an *osr* of 64 or  $f_s = 3.072$  MHz [60]. Audio band conversion has also been performed by a standard 2nd order modulator with a 4 bit internal ADC and DAC. The converter achieved nearly 16 bit peak SNR for a 20.5 kHz

Table: Sigma Delta Converters in Use Today

Signal Band ( $f_B$ )	Sampling Frequency ( $f_s$ )	OSR ( $f_s/2f_B$ )	Overall Resolution (# bits)	Modulator Structure	Internal Quantizer	Application	Reference
492 Hz	128 kHz	128	20	4th order	1 bit	instrumentation	[55]
500 Hz	128 kHz	128	20	4th order	1 bit	seismic	[56]
4 kHz	4 MHz	500	13	1st order	1 bit	speech	[57]
4 kHz	1.024 MHz	128	13	2nd order	1 bit	speech	[58]
20.5 kHz	5.25 MHz	128	16	2nd order	4 bits	audio	[30]
24 kHz	6.144 MHz	128	18	4th order	4 bits	audio	[59]
24 kHz	3.072 MHz	64	16	2nd order	1 bit	audio	[60]
25 kHz	6.4 MHz	128	17	"2-1" cascade	1 bit	audio	[39]
40 kHz	10.24 MHz	128	14	2nd order	1 bit	ISDN	[16]
40 kHz	2.56 MHz	32	13	"2-1" cascade	1 bit	ISDN	[37]
100 kHz	3.25 MHz	16	15	"2-2-2" cascade	3 level	digital cellular radio	[61]
160 kHz	20.48 MHz	24	16	"2-1" cascade	1 bit	-	[38]
250 kHz	32 MHz	32	14	4th order	1 bit	-	[20]
1 MHz	50 MHz	20	12	"2-1" cascade	1 and 3 bit	ultrasound	[62]

bandwidth using an *osr* of 128, or  $f_s = 5.25$  MHz [30]. Finally, an architecture consisting of a cascade of second order and a first order modulator and employing 1 bit quantizers achieves nearly 17 bit performance for a 25 kHz bandwidth, also using an *osr* of 128 with a  $f_s = 6.4$  MHz [39].

A sigma-delta converter has also been used as a receiver input of an ISDN U-interface 2B1Q access rate receiver [16]. The converter attains a dynamic range of 89 dB, or a resolution of 14 bits, for a 40 kHz bandwidth. A standard 2nd order modulator was used with a 1 bit quantizer running at a 10.24 MHz sampling frequency, and an *osr* of 128. Another converter [37] used for a similar ISDN U-interface consisted of a cascade of a 2nd order modulator, followed by a 1st order modulator. The resolution was 13 bits, using a  $f_s$  of 2.56 MHz, i.e., an *osr* of 32.

A sigma-delta ADC has been used as the baseband converter for a digital cellular radio that required a moderately higher bandwidth of 100 kHz [61]. The converter produced 15 bit peak resolution by using a three stage cascade employing three 2nd order modulators running at  $f_s = 3.25$  MHz. The *osr* was about 16.

At somewhat higher bandwidths, high resolution conversion has been attained for 160 kHz and 250 kHz bandwidths. The 160 kHz bandwidth converter achieves a dynamic range of 96 dB, or nearly 16 bit performance using a cascade of second and first order modulators that employ 1 bit quantizers [38]. The sampling rate is  $f_s = 20.48$  MHz and the resulting *osr* is 24. The 250 kHz bandwidth converter achieves 14 bit resolution using a 4th order modulator using a 1 bit quantizer [20]. The *osr* is 32 and thus the  $f_s$  is 32 MHz.

At high conversion bandwidths of about 1 MHz, a 12 bit converter has been realized with an *osr* of about 24, and a  $f_s$  of about 50 MHz [62]. The converter uses a cascade of second order and first order modulators. The second order modulator used in the first stage employs a 1 bit quantizer, while the 1st order modulator used in the second stage utilizes a 3 bit internal ADC and DAC. Such a converter can find use in data acquisition for ultrasound imaging systems.

Finally, sigma-delta ADCs have been integrated with digital signal processor (DSPs) to provide a single chip data conversion/computation engine solution. Sigma-delta ADCs require relatively imprecise analog circuits and digital decimation filtering, thus making them good candidates for fabrication using digital technology such as CMOS. The task of decimation can be handled entirely by a DSP or shared by a DSP and some extra digital hardware dedicated to performing a portion of the decimation.

## Conclusion

We have reviewed the basic principles of A/D conversion with sigma-delta modulators. The techniques of oversampling and noise shaping allows the use of relatively imprecise analog circuits to perform high resolution conversion using only a 1 bit A/D converter. Oversampling reduces the amount of quantization noise power present in the signal band, and noise shaping further attenuates quantization noise in the

signal band, thereby pushing noise power to out-of-band frequencies. The use of analog filtering combined with feedback around the 1 bit A/D can be used to implement the noise shaping sigma-delta modulator. The noise power that is pushed outside the signal band can be attenuated by a digital filter such that it has no further effect on the signal.

Various sigma-delta architectures exist and many of these have been used in applications such as instrumentation, speech and Hi-Fi audio digitization, ISDN and digital cellular radio. Sigma-delta techniques are also applicable to the high resolution A/D conversion of narrowband band-pass signals using band-pass sigma-delta modulators. Parallel sigma-delta systems offer the potential for extending high resolution operation to larger signal bandwidths than currently possible with single channel systems.

Pervez M. Aziz is a Member of Technical Staff at AT&T Bell Laboratories, Murray Hill, NJ. Henrik V. Sorensen is Senior Researcher at Ariel Corporation, Highland Park, NJ. Jan Van der Spiegel is Professor, Department of Electrical Engineering, University of Pennsylvania, Philadelphia. This work was completed while the first two authors were at the University of Pennsylvania.

## References

1. S. Renukunta and D. Wells, "Optical memory and blue lasers," *IEEE Potentials*, pp. 14-18, Oct/Nov 1994.
2. J. Candy and G. Temes, "Oversampling methods for A/D and D/A conversion" in *Oversampling Delta-Sigma Data Converters*, pp. 1-25, IEEE Press, 1992.
3. A. Oppenheim and R. Schaffer, *Discrete Time Signal Processing*, (Prentice-Hall, 1989).
4. W. Bennett, "Spectra of quantized signals," *Bell System Technical Journal*, pp. 446-472, July 1948.
5. Boser and B. Wooley, "The design of sigma-delta modulation analog-to-digital converters," *IEEE Journal of Solid State Circuits*, pp. 1298-1308, December, 1988.
6. B. Leung, "Theory of sigma-delta analog to digital converter," *IEEE International Symposium on Circuits and Systems Tutorials*, pp. 196-223, 1994.
7. H. Inose and Y. Yasuda, "A unity bit coding method by negative feedback," *Proceedings of the IEEE*, pp. 1524-1535, November, 1963.
8. J. Candy, "A use of double integration in sigma delta modulation," *IEEE Transactions on Communications*, pp. 249-258, March, 1985.
9. V. Friedman, "The structure of limit cycles in sigma delta modulation," *IEEE Transactions on Communications*, pp. 972-979, August, 1988.
10. J. Candy, O. Benjamin, "The structure of quantization noise from sigma-delta modulation," *IEEE Transactions on Communications*, pp. 1316-1323, September, 1981.
11. S. Hein and A. Zakhor, "On the stability of sigma delta modulators," *IEEE Transactions on Signal Processing*, pp. 2322-2348, July, 1993.
12. R. Gray, "Spectral analysis of quantization noise in a single-loop sigma-delta modulator with dc input," *IEEE Transactions on Communications*, pp. 588-599, June, 1989.
13. N. He, F. Kuhlmann, A. Buzo, "Double-loop sigma-delta modulation with dc input," *IEEE Transactions on Communications*, pp. 487-495, April, 1990.
14. R. Gray, "Quantization noise spectra," *IEEE Transactions on Information Theory*, pp. 1220-1244, November, 1990.
15. S. Rangan and B. Leung, "Quantization noise spectrum of double-loop sigma-delta converter with sinusoidal input," *IEEE Transactions on Circuits and Systems II*, pp. 168-173, February, 1994.

16. S. Norsworthy, I. Post, H. Fetterman, "A 14-bit 80kHz sigma-delta A/D converter: modeling, design, and performance evaluation," *IEEE Journal of Solid State Circuits*, pp. 256-266, April, 1989.
17. S. Norsworthy and D. Rich, "Idle channel tones and dithering in delta sigma modulators," *95th Convention of the Audioengineering Society*, preprint 3711, October, 1993.
18. K. Chao, S. Nadeem, W. Lee, and C. Sodini, "A higher order topology for interpolative modulators for oversampling A/D converters," *IEEE Transactions on Circuits and Systems*, pp. 309-318, March, 1990.
19. P. Ferguson, A. Ganesan, R. Adams, "One bit higher order sigma-delta A/D converters," *Proceedings, IEEE International Symposium on Circuits and Systems*, pp. 890-893, 1990.
20. F. Op't Eynde, G. Yin, W. Sansen, "A CMOS fourth-order 14b 500k-sample/s sigma-delta ADC converter," *Digest of Technical Papers, International Solid State Circuits Conference*, pp. 62-63, 1991.
21. R. Adams, "Design aspects of high-order delta-sigma A/D converters," *IEEE International Symposium on Circuits and Systems Tutorials*, pp. 235-259, 1994.
22. S. Ardanal and J. Paulos, "An analysis of nonlinear behavior in delta-sigma modulators," *IEEE Transactions on Circuits and Systems*, pp. 593-603, June, 1987.
23. R. Baird and T. Fiez, "Stability analysis of high-order delta-sigma modulation for ADCs," *IEEE Transactions on Circuits and Systems II*, pp. 59-62, January, 1994.
24. S. Moussavi and B. Leung, "High-order single-stage single-bit oversampling A/D converter stabilized with local feedback loops," *IEEE Transactions on Circuits and Systems II*, pp. 19-25, January, 1994.
25. D. Anastassiou, "Error diffusion coding for A/D conversion," *IEEE Transactions on Circuits and Systems*, pp. 1175-1186, September, 1989.
26. R. Schreier and Y. Yang, "Stability tests for single-bit sigma-delta modulators with second-order FIR noise transfer functions," *Proceedings, IEEE International Symposium on Circuits and Systems*, pp. 1316-1319, 1992.
27. L. Larson, T. Cataltepe, and G. Temes, "Multibit oversampled sigma-delta A/D converter with digital error correction," *Electronics Letters*, pp. 1051-1052, August 4, 1988.
28. T. Leslie and B. Singh, "An improved sigma-delta modulator architecture," *Proceedings, IEEE International Symposium on Circuits and Systems*, pp. 372-375, 1990.
29. A. Hairapetian, G. Temes, and Z. Zhang, "Multibit sigma-delta modulator with reduced sensitivity to DAC nonlinearity," *Electronics Letters*, pp. 990-991, May 23, 1991.
30. M. Sarhang-Nejad and G. Temes, "A high-resolution multibit sigma-delta ADC with digital correction and relaxed amplifier requirements," *IEEE Journal of Solid State Circuits*, pp. 648-660, June, 1993.
31. L. Carley, "A noise-shaping coder topology for 15+ bit converters," *IEEE Journal of Solid State Circuits*, pp. 267-273, April, 1989.
32. B. Leung and S. Sutartja, "Multibit sigma-delta A/D converter incorporating a novel class of dynamic element matching technique," *IEEE Transactions on Circuits and Systems II*, pp. 35-51, January, 1992.
33. F. Chen and B. Leung, "A high resolution multi-bit sigma-delta modulator with individual level averaging," *Digest of Technical Papers, IEEE Symposium on VLSI Circuits*, pp. 101-102, June, 1994.
34. Y. Matsuya, K. Uchimura et al, "A 16-bit oversampling A-to-D conversion technology using triple-integration noise shaping," *IEEE Journal of Solid State Circuits*, pp. 921-929, December, 1987.
35. K. Uchimura, T. Hayashi, T. Kimura and A. Iwata, "Oversampling A-to-D and D-to-A converters with multistage noise shaping modulators," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pp. 1899-1905, December, 1988.
36. M. Rebeschini, N. van Bavel, P. Rakers, et al, "A 16-b 160-kHz CMOS A/D converter using sigma-delta modulation," *IEEE Journal of Solid State Circuits*, pp. 431-440, April, 1990.
37. L. Longo and M. Copeland, "A 13 bit ISDN-band oversampled ADC using two-stage third order noise shaping," *Proceedings, IEEE Custom Integrated Circuits Conference*, pp. 21.2.1-21.2.4, 1988.
38. G. Yin, F. Stubbe, W. Sansen, "A 16-b 320-kHz CMOS A/D converter using two-stage third-order sigma-delta noise shaping," *IEEE Journal of Solid State Circuits*, pp. 640-647, June, 1993.
39. L. Williams and B. Wooley, "Third-order sigma-delta modulator with extended dynamic range," *IEEE Journal of Solid State Circuits*, pp. 193-202, March, 1994.
40. D. Ribner, "A comparison of modulator networks for high-order oversampled sigma-delta analog-to-digital converters," *IEEE Transactions on Circuits and Systems*, pp. 145-159, February, 1991.
41. R. Schreier and M. Snelgrove, "Bandpass sigma-delta modulation," *Electronics Letters*, pp. 1560-1561, November 9, 1989.
42. S. Jantzi, R. Schreier and M. Snelgrove, "Bandpass sigma-delta analog-to-digital conversion," *IEEE Transactions On Circuits and Systems*, pp. 1406-1409, November, 1991.
43. L. Longo and B. Horng, "A 15b 30kHz bandpass sigma delta modulator," *Digest of Technical Papers, International Solid State Circuits Conference*, pp. 226-227, 1993.
44. S. Jantzi, M. Snelgrove, P. Ferguson, "A 4th-order bandpass sigma-delta modulator," *IEEE Journal of Solid State Circuits*, pp. 282-291, March, 1993.
45. G. Tröster, H Dreßler, et al, "An interpolative bandpass converter on a 1.2-um BiCMOS analog/digital array," *IEEE Journal of Solid State Circuits*, pp. 471-477, April, 1993.
46. S. Jantzi, K. Martin, M. Snelgrove, A. Sedra, "Complex bandpass sigma-delta converter for digital radio," *Proceedings, IEEE International Symposium on Circuits and Systems*, pp. 453-456, 1994.
47. P. Aziz, H. Sorensen, J. Van der Spiegel, "Performance of complex noise transfer functions in bandpass and multi band sigma delta systems," *Proceedings, IEEE International Symposium on Circuits and Systems*, pp. 641-644, 1995.
48. Q. Liu, M. Snelgrove, A. Sedra, "Switched-capacitor implementation of complex filters," *Proceedings, IEEE International Symposium on Circuits and Systems*, pp. 1121-1124, 1986.
49. W. Black and D. Hodges, "Time interleaved converter arrays," *IEEE Journal of Solid State Circuits*, pp. 1022-1029, December, 1980.
50. A. Petraglia and S. Mitra, "High speed A/D conversion using QMF banks," *Proceedings, IEEE International Symposium on Circuits and Systems*, pp. 2797-2800, 1990.
51. P. Aziz, H. Sorensen, J. Van der Spiegel, "Multiband sigma-delta modulation," *Electronics Letters*, pp. 760-762, April 29, 1993.
52. P. Aziz, H. Sorensen, J. Van der Spiegel, "Multiband sigma delta analog to digital conversion," *Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 249-252, 1994.
53. R. Khoini-Poorfard, L. Lim, D. Johns, "Time-interleaved oversampling converters," *Electronics Letters*, pp. 1673-1674, September 16, 1993.
54. I. Galton and H. Jensen, "Delta-sigma modulator based A/D conversion without oversampling," to appear, *IEEE Transactions on Circuits and Systems*.
55. C. Thompson, S. Bernadas, "A digitally corrected 20b delta-sigma modulator," *Digest of Technical Papers, International Solid State Circuits Conference*, pp. 194-195, 1994.
56. D. Kerth, D. Kasha, et al, "A 120 dB linear switched-capacitor delta-sigma modulator," *Digest of Technical Papers, International Solid State Circuits Conference*, pp. 196-197, 1994.
57. B. Leung, R. Neff, P. Gray, R. Broderson, "Area-Efficient Multichannel Oversampled PCM Voice-Band Coder," *IEEE Journal of Solid State Circuits*, pp. 1351-1357, December, 1988.
58. V. Friedman, D. Brinthaup, et al, "A dual-channel voice-band PCM codec using sigma-delta modulation technique," *IEEE Journal of Solid State Circuits*, pp. 274-280, April, 1989.
59. R. Adams, "Design and implementation of an audio 18-bit analog-to-digital converter using oversampling techniques," *Journal of the Audioengineering Society*, pp. 153-166, March, 1986.
60. D. Welland, B. Del Signore, et al, "A stereo 16-bit delta-sigma A/D converter for digital audio," *Journal of the Audioengineering Society*, pp. 476-486, June, 1989.
61. I. Dedic, "A sixth-order triple-loop sigma-delta CMOS ADC with 90 dB SNR and 100 kHz bandwidth," *Digest of Technical Papers, International Solid State Circuits Conference*, pp. 188-189, 1994.
62. B. Brandt and B. Wooley, "A CMOS oversampling A/D converter with 12b resolution at conversion rates above 1 MHz," *Digest of Technical Papers, International Solid State Circuits Conference*, pp. 64-65, 1991.

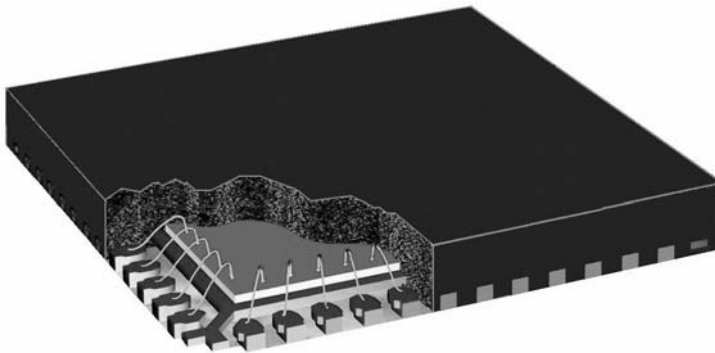
# ANNEXE E

## “Green” QFN Packages from Texas Instruments

Richard Saye

### INTRODUCTION

Texas Instruments (TI) has recently taken proactive steps to introduce its package options as “Green,” including its Quad Flat No-lead (QFN) packages. The QFN packages are near-chip-scale plastic encapsulated packages that use conventional copper lead frame technology, without peripherally protruding leads, within their construction. Leads that are plated with NiPdAu to achieve a RoHS compatible Pb-Free finish that avoid problematic Sn (tin) plating known to produce whiskers. Also “Green” by utilizing mold compounds that exclude hazardous substances antimony and bromine that may soon join Pb (lead) as prohibited RoHS material. This assembly results in a cost-effective environmentally friendly packaging solution utilizing superior materials that allow a 260C reflow, maximize board space with added benefits of improved electrical and thermal performance over traditional leaded packages. QFN packages have an exposed pad that enhances the thermal and electrical characteristics, enabling high-power and high-frequency applications.



Illustrated Cut Out QFN Package



Sectional View Soldered QFN Lead

### SOLDERABILITY TESTING

Normally solderability testing of finished integrated circuit (IC) packages is performed in order to predict performance of these devices in an actual printed circuit board reflow process. The "Dip-and-Look" (DNL) test method (TI's QSS 009-130) is the electronics industry standard test method for solderability testing the leads of finished IC devices (ANSI/JSTD-002). This test method consists of immersing the IC leads into a rosin type flux for 5-10 seconds prior to immersion into molten 63% Sn/ 37% Pb solder for 5-10 seconds. Upon removal from the solder, the leads are cleaned and visually inspected using 10x optical inspection. The percentage of the lead surface not wetted by the solder is calculated. Industry standards specify that "all terminations shall exhibit a continuous solder coating free from defects for a minimum of 95% of the critical surface area of any individual termination". Yet this ANSI/JSTD-002 solderability test is not as applicable in

today's Pb-free environment due to the lower temperature requirements of SnPb solder and non PWB attachment. The Surface Mount Solderability Test Method (SMT test) utilized within this report is intended to simulate the actual environment which surface mount devices encounter during the solder reflow process.

For this testing PWB and stencil were designed specifically for the QFN package. Refer to Application Note, Quad Flat-Pack No Lead Packages, Texas Instruments Literature No. SCBA017, SLUA271, Product Data Sheets for specific thermal information, via, and recommended PWB board layout. These documents are available at [www.ti.com](http://www.ti.com).

Although TI recommends NSMD (Non Solder Mask Defined) pads over SMD (Solder Mask Defined) pads when surface mounting QFN's both can be utilized. NSDM allows tighter tolerance on copper etching and by design provides a larger solderable area due to the exposed edges free from solder mask, thus providing improved solder-joint reliability. Additionally, SMD designs will inherently create stress points where the solder wets to the top of the pad on top of the lead. This stress concentration point is eliminated when the solder is allowed to contact the sides of the leads in NSMD designs.

There are several performance issues that need review when designing stencils.

- Control of the aperture size, including length with and thickness of stencil aperture's paste release characteristics as it is desirable that all paste is released to the PWB pads.
- Factures that can affect the paste release; are laser cutting with trapezoidal walls and rounding corners and preferably electropolished or electropolishing with nickel plating.
- Utilizing a print-area ratio (PAR) guideline to quantify the paste release capability of an aperture design. Acceptable results are normally archived when this PAR ratio is above 0.66, yet better release is accomplished when the ratio is above 0.75.
- Positional accuracy of the aperture location must be well controlled to avoid solder misregistration.

Customers should contact their board assembly site for recommended solder mask tolerances, via tenting, and stencil design recommendations. Refer to IPC 7525 for stencil design considerations and IPC-SM-782 as an alternate information source for PCB land pattern designs.

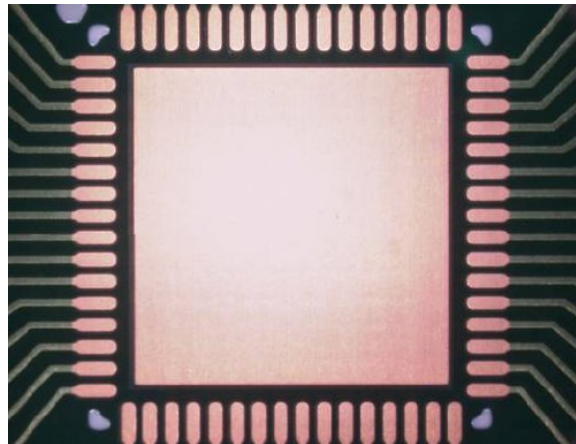
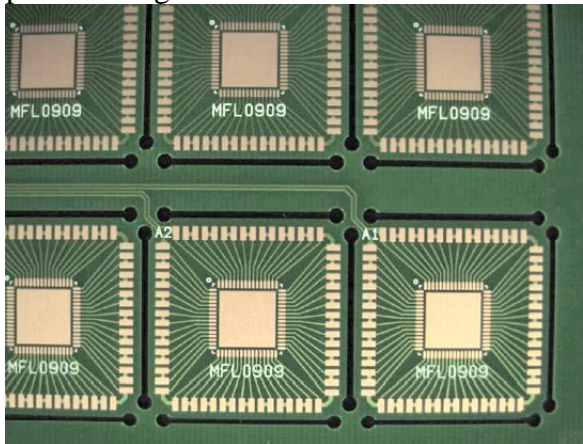


Figure 1: QFN Test Board designed per IPC standard.

Figure 2: QFN land pattern

QFN packages are soldered using SnAgAu paste and PWB mounted. TI recommends following the solder paste supplier's recommendations to optimize flux activity and also to achieve proper melting temperatures of the alloy within the guidelines of J-STD-20.

TI also recommends the use of type 3 or finer solder paste when mounting QFN's. The use of paste offers the following advantages:

- Contains flux to aid wetting of the solder to the PCB land.
- The adhesive properties of the paste will hold the component in place during manufacture.
- Paste by volume contains around 50% metal load typically and can be varied by print volume for calculating the amount of paste necessary to form a given solder joint. QFN's are typically manufactured with printed center pad volumes between 50% and 80% by area to facilitate wetting of the periphery solder joints and also to maintain a standoff from the board surface.
- Paste contributes to the final volume of solder in the joint, and thus allows this volume to be varied to give an optimum joint.
- Paste selection is normally driven by overall system assembly requirements. In general, the "no clean" compositions are preferred due to the difficulty in cleaning under the mounted components.

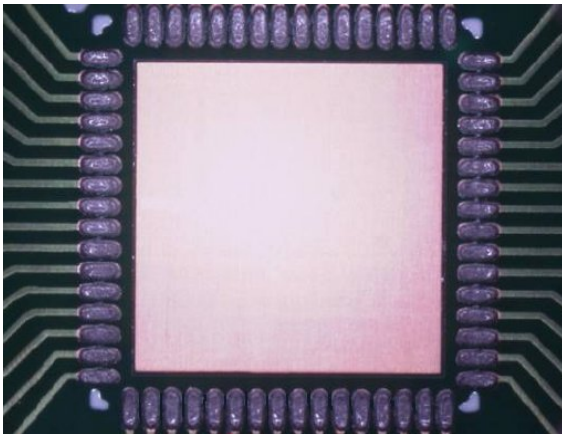


Figure 5; Solder Applied (center pad not soldered for this testing, normally minimum 50% coverage is recommended for the center pad)

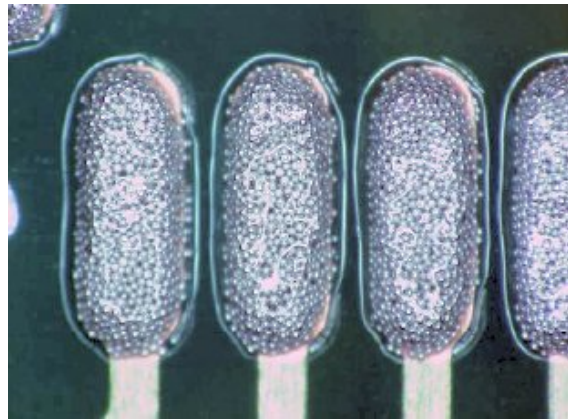
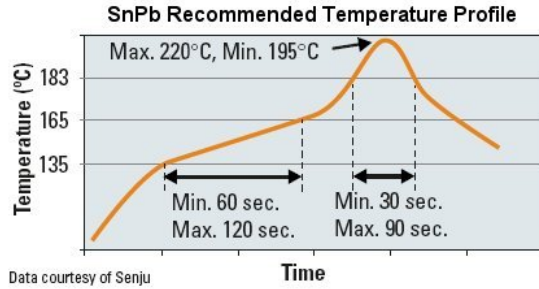


Figure 6; Close Up Soldered Leads

Note due to minimal package weight of TI QFN's reflow oven's air flow might need monitoring to insure package shift during reflow will not occur.





Data courtesy of Senju

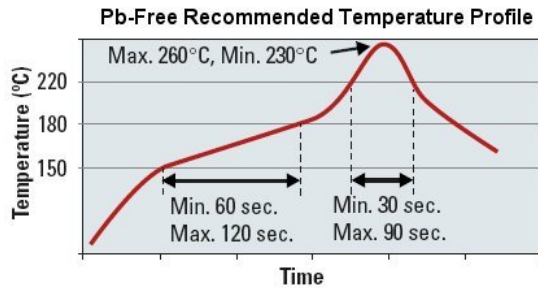


Figure 3; TI QFN's are compatible with SnPb Profiles (Shown only as reference)

Figure 4; Pb-Free temperature profile used in testing

Figures above illustrates a range of temperatures that TI QFN packages are capable of withstanding without risk to package reliability but TI prefers parts to be processed with the lowest peak temperature possible while also remaining below the components peak temperature rating as listed on the MSL label. In general, the temperature of the part should be raised not more than 2°C/sec during the initial stages of the reflow profile. The exact profile would depend on the maximum peak temperature for the component as rated on the MSL label, the solder paste manufacturer's recommendation, complexity of the PWB, and capability of the reflow equipment to be confirmed by the SMT assembly operation.

The requirement to steam age QFN's within this test has been designated to simulate extended time periods between testing and soldering (>6 months) and have either limited thermal exposure or multiple thermal exposures before soldering. This current requirement assumes that "properly applied NiPdAu plating can withstand the steam conditioning environment well beyond the eight hours specified and may survive natural aging well beyond 12 months". Steam aging samples prior to testing may become more critical as the shelf-life of components and boards is reduced.

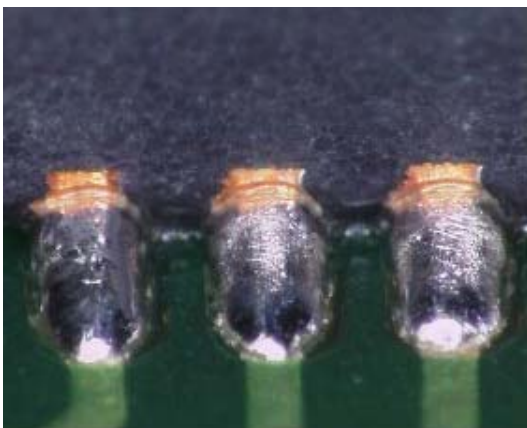


Figure 7; Solderability Testing Unit without Steam Ageing

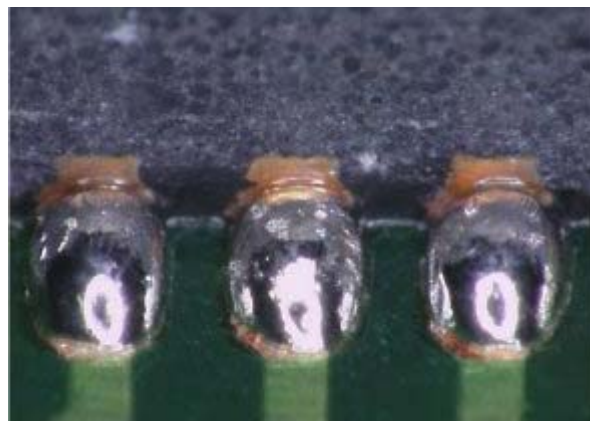


Figure 8; Solderability Testing Unit with 8 hour Steam Ageing

Results of solderability testing showed all units passed visual and X-ray inspection and electrical testing possible with utilization of a daisy chained die. The best possible

perimeter solder joints will have a good fillet on outside and a 50 – 75 micro standoff height. Additionally solder joints should have enough volume so an “Hour Glass” shaped connection is not formed. The electrical connection to the perimeter pads is made on the bottom of the package. Thus solder wetting and electrical connection to the sides of the QFN are not guaranteed as the sides are not plated. For additional industry standards information on solder requirements see IPC/EIA J-Std-001C Section 9.2.6.4 and/or IPC A-610D.

Additionally, testing was done temperature cycling PWB mounted QFN’s at a range of -40°C to +125°C profile with a 10 minute ramp 10 minute dwell. Failure definition is the sample reaching 1.2x Initial Resistance. Both preconditioned (steam aged) and non-preconditioned devices were utilized within this testing. Package utilized was a 9 x 9 mm 64-RGC QFN that is defined “Green”. (Pb-Free no antimony nor bromine within it’s construction) Conclusion was all units passed thermo cycling with average group one failures occurring at 5704 and group two occurring 5155.

As part of TI internal qualification requirements drop testing was also completed on these QFN packages with test. Testing conditions were as follows; Board 8 layers, 1mm thick 101 x 48 x 1.120, Drop Conditions are 1500G with a one millisecond pulse, Failure definition is sample reaching 1.2x Initial Resistance. Conclusion was all units passed beyond the 40 drop requirement with average determined to be 233 drops.

In today’s manufacturing environment aggressively miniaturizing electronic produces driven primarily by the consumer markets need in reducing weight and height while still improving electrical performance and functionality Texas Instruments QFN packaging is the right fit. These packages meet forthcoming RoHS material restrictions requiring Pb-free devices and exceed material restrictions by removing elements antimony & bromine used as fire retardants. Additional information on Texas Instruments QFN’s are available at [www.ti.com](http://www.ti.com).

# ANNEXE F

## USART: Universal Synchronous/Asynchronous Receiver/Transmitter

La communication est synchrone si l'émetteur et le récepteur disposent de la même référence d'horloge pour communiquer. Ceci ne veut pas dire que l'horloge est transmise en même temps que les données, mais que la transmission des données comporte des informations permettant à l'horloge réception de rester strictement en phase avec l'horloge émission.

Dans le cas contraire la communication est asynchrone. Emetteur et récepteur ont chacun une horloge qui est à peu près à la même fréquence. L'émetteur envoie ses données avec son horloge et le récepteur les échantillonne avec la sienne. Si on suppose que le premier bit est échantillonné à peu près au milieu, cela devient de moins en moins vrai avec les bits suivants du fait de la dérive des horloges. C'est pour cette raison que l'on échantillonne un groupe restreint de bits de cette façon, en général 8 bits, appelé caractère dans la suite du texte. Si l'on part d'une ligne « au repos » c'est à dire dans l'état logique 0 par exemple, l'émission d'un caractère commencera toujours par une transition 01 pour indiquer le début de la transmission et cette transition, pour des raisons de simplicité, va durer le temps d'un bit, d'où son nom de bit de START. La transition 01 permet de savoir qu'une transmission commence et d'échantillonner le bit de START à peu près au milieu. Pour réaliser cette opération on utilise une horloge d'échantillonnage de fréquence en général 16 fois plus élevée que la vitesse de transmission. Ainsi on échantillonne la ligne de transmission avec cette horloge et on détecte le début du bit de START avec une incertitude maximale de 1/16. 8 tops d'horloge plus tard on est au milieu (ou à peu près) du bit de START, et 16 tops plus tard on est au milieu (à peu près, mais avec une incertitude plus élevée) du premier bit de donnée, et ainsi de suite pour les bits suivants, échantillonnés toujours à peu près au milieu mais avec une incertitude de plus en plus forte, mais qui reste très raisonnable pour un nombre restreints de bits échantillonnés. Une fois ce caractère émis il est possible de commencer l'émission du caractère suivant. Mais pour avoir une transition 01 de début de bit de START permettant une nouvelle synchronisation, il faut être sûr que la ligne soit à 0. C'est pour cette raison qu'une fois un caractère émis il faut mettre la ligne à zéro pendant un temps minimal, en général choisi pour des raisons de simplicité égal à la durée d'un bit, d'où le nom de bit de STOP. Bit de START et bit de STOP « encadrent » donc le caractère émis : il faut donc 10 bits pour envoyer un caractère de 8 bits. Cette transmission asynchrone est aussi appelée transmission START-STOP.

La transmission synchrone a un meilleur rendement que la transmission asynchrone (pas de bits de START et STOP à chaque caractère), mais elle implique de maintenir les horloges synchrones et est donc plus complexe. On utilise la transmission synchrone pour transférer des volume importants d'information. La transmission asynchrone est bien adaptée à la communication entre un terminal très simple comme un ensemble clavier-écran et un calculateur. Chaque appui sur une touche du clavier provoque l'émission série immédiate du code ASCII correspondant, encadré par un bit de START et un bit de STOP. On conçoit qu'il est difficile dans ce contexte de faire autrement que de l'asynchrone.

Le circuit AT91X40 contient 2 contrôleurs de communication série synchrone et asynchrone identiques. En salle de TP, les 2 sont connectés sur 2 ports série du PC. L'un permet la communication entre l'environnement de développement du PC et le circuit, l'autre permet la communication entre un mode terminal du PC et l'application qui s'exécute sur la carte. On obtient ainsi deux systèmes de communication complètement indépendants.

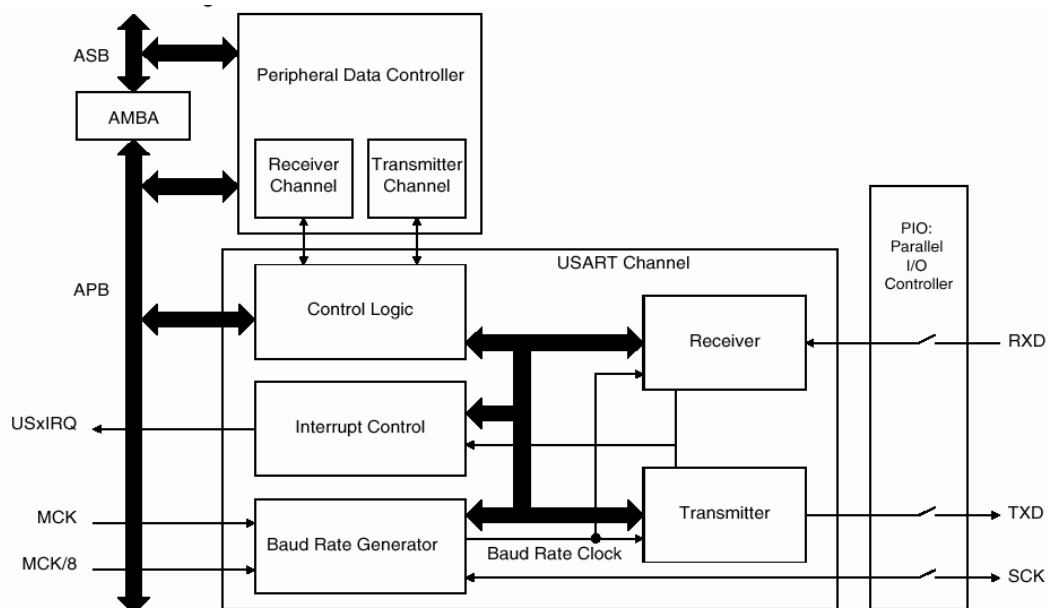
Les deux circuits USART0 et USART1 sont identiques et différenciés par leur adresse :

**USART0:** 0xFFFD0000 (Code Label USART0\_BASE)

**USART1:** 0xFFFC0000 (Code Label USART1\_BASE)

L'USART comporte une partie émission (« transmitter ») et une partie réception (« receiver ») qui fonctionnent de façon complètement indépendantes.

Les bits de la partie émission sont envoyés en série sur la broche TXD, les bits de la partie réception sont reçus sur la broche RXD. La vitesse de transmission est programmable via le module « Baud Rate Generator ». Elle est définie à partir d'une horloge qui peut être externe au circuit et entrer par la broche SCK, ou dérivée de



l'horloge MCK interne du circuit et dans ce cas être une source disponible « à l'extérieur » du circuit via la broche SCK, donc bidirectionnelle. Nous n'utiliserons pas cette dernière fonctionnalité : l'horloge sera dérivée de l'horloge MCK et la broche SCK pourra être utilisée par le PIO. Seules les broches RXD et TXD devront être retirées du contrôle du PIO. Emission et réception peuvent être contrôlées par test d'état ou par interruption via la logique « Interrupt Control » et le transfert peut se faire octet par octet ou automatiquement par bloc d'octets via la logique « Peripheral Data Controller » et ceci toujours de façon indépendante pour la partie émission et la partie réception. Par exemple on peut envoyer des caractères un par un en fonctionnant par test d'état tout en étant en réception d'un bloc de caractères qui déclenchera une interruption lorsque le dernier caractère du bloc sera arrivé.

Toutes ces possibilités sont obtenues par la programmation des 15 registres de chaque USART.

Les registres de l'USART sont les suivants :

Offset	Register	Name	Access	Reset State
0x00	Control Register	US_CR	Write Only	–
0x04	Mode Register	US_MR	Read/Write	0
0x08	Interrupt Enable Register	US_IER	Write Only	–
0x0C	Interrupt Disable Register	US_IDR	Write Only	–
0x10	Interrupt Mask Register	US_IMR	Read Only	0
0x14	Channel Status Register	US_CSR	Read Only	0x18
0x18	Receiver Holding Register	US_RHR	Read Only	0
0x1C	Transmitter Holding Register	US_THR	Write Only	–
0x20	Baud Rate Generator Register	US_BRGR	Read/Write	0
0x24	Receiver Time-out Register	US_RTOR	Read/Write	0
0x28	Transmitter Time-guard Register	US_TTGR	Read/Write	0
0x2C	Reserved	–	–	–
0x30	Receive Pointer Register	US_RPR	Read/Write	0
0x34	Receive Counter Register	US_RCR	Read/Write	0
0x38	Transmit Pointer Register	US_TPR	Read/Write	0
0x3C	Transmit Counter Register	US_TCR	Read/Write	0

## REGISTRE DE CONTROLE US\_CR

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	SENDA	STTTO	STPBRK	STTBRK	RSTSTA
7	6	5	4	3	2	1	0
TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX	–	–

Les 11 bits utilisés dans ce registre permettent de réaliser les fonctions suivantes :

• **RSTRX: Reset Receiver**

L'écriture d'un 0 est sans effet, l'écriture d'un 1 provoque un RESET de la logique de réception

• **RSTTX: Reset Transmitter**

L'écriture d'un 0 est sans effet, l'écriture d'un 1 provoque un RESET de la logique de réception

*Note : ces deux fonctionnalités ne sont utilisés que dans le cas d'un redémarrage « à chaud », c'est à dire une réinitialisation sans que ce soit un RESET matériel.*

• **RXEN: Receiver Enable (Code Label US\_RXEN)**

L'écriture d'un 0 est sans effet, l'écriture d'un 1 valide la logique de réception, si RXDIS est égal à 0

• **RXDIS: Receiver Disable (Code Label US\_RXDIS)**

L'écriture d'un 0 est sans effet, l'écriture d'un 1 invalide la logique de réception, quelque soit l'état de RXDIS

• **TXEN: Transmitter Enable (Code Label US\_TXEN)**

L'écriture d'un 0 est sans effet, l'écriture d'un 1 valide la logique d'émission, si TXDIS est égal à 0

**• TXDIS: Transmitter Disable (Code Label US\_TXDIS)**

L'écriture d'un 0 est sans effet, l'écriture d'un 1 invalide la logique d'émission, quelque soit l'état de TXDIS  
*Note : il faut valider les logiques d'émission et de réception avant de pouvoir les utiliser. A faire normalement qu'une seule fois dans la séquence de RESET.*

**• RSTSTA: Reset Status Bits (Code Label US\_RSTSTA)**

L'écriture d'un 0 est sans effet, l'écriture d'un 1 met à 0 les bits d'états PARE, FRAME, OVRE et RXBRK du registre US\_CSR.  
*Note : à chaque réception d'un caractère il faudra vérifier qu'il n'y a pas eu d'erreur de réception et donc de remettre à 0 ces bits indicateurs d'erreurs. Il n'y a pas d'erreur de ce type possible en émission*

**• STTBRK: Start Break (Code Label US\_STTBRK)**

Un "Break" est souvent utilisé comme un caractère particulier de contrôle de communication. Emettre un break revient à placer la ligne TXD à 0 pour une durée supérieure à l'émission d'un caractère, ce qui provoque à l'arrivée une erreur de cadrage (FRAME error), c'est à dire une absence de bit stop. Ce type d'erreur plus une lecture de caractère égal à 0 doit être interprété comme un break.  
 L'écriture d'un 0 est sans effet, l'écriture d'un 1 met la ligne TDX à 0 après que les caractères présents dans US\_THR et le registre à décalage aient été émis.

**• STPBRK: Stop Break (Code Label US\_STPBRK)**

L'écriture d'un 0 est sans effet, l'écriture d'un 1 (si un break a été programmé) termine le break après un minimum d'un moins un caractère et laisse la ligne TXD à 1 pour 12 périodes d'horloge.  
*Note : en TP on n'a pas de raison d'utiliser cette fonctionnalité.*

**• STTTO: Start Time-out (Code Label US\_STTTO)**

L'écriture d'un 0 est sans effet, l'écriture d'un 1 provoque la mise en route d'un délai programmé dans le registre US\_RTOR, dont l'échéance peut déclencher une interruption. Ce mécanisme permet de suivre la réception des caractères d'un message : en cas de problème côté émetteur par exemple, la réception s'arrête, et on peut ainsi être automatiquement prévenu de cette absence de transmission.

**• SENDA: Send Address (Code Label US\_SENDA)**

L'écriture d'un 0 est sans effet, l'écriture d'un 1 provoque l'émission du prochain caractère avec le bit de parité, qui sert dans ce cas d'indicateur d'adresse, positionné. Voir le paragraphe « liaison multipoint » à la fin.

**REGISTRE DE MODE US\_MD**

31	30	29	28	27	26	25	24
—	—	—	—	—	—	—	—
23	22	21	20	19	18	17	16
—	—	—	—	—	CLKO	MODE9	—
15	14	13	12	11	10	9	8
CHMODE		NBSTOP		PAR		SYNC	
7	6	5	4	3	2	1	0
CHRL		USCLKS		—	—	—	—

USCLKS	Source horloge
00	MCK
01	MCK/8
1x	SCK

Le champ USCLK permet de choisir la source de l'horloge : soit celle du processeur MCK, soit externe via la broche SCK. Si la vitesse est lente on doit prendre MCK/8. La vitesse est définie dans le registre Baud Rate generator.

CHRL	Longueur caractère
00	5 bits
01	6 bits
10	7 bits
11	8 bits

Le champ CHRL permet de définir la longueur du caractère envoyé : 5 bits (code du TELEX, obsolète), 6 bits (pas d'usage ?), 7 bits (ASCII), 8 bits (binaire ou ASCII sans traitement de la parité)

Sync (Synchronous Mode Select)	
0	Mode asynchrone
1	Mode synchrone

PAR	Type de parité
000	Paire
001	Impaire
010	Bit toujours à 0
011	Bit toujours à 1
10x	Pas de parité
11x	Fonctionnement en mode multipoint

Le fonctionnement en mode multipoint se sert de la position bit de parité pour indiquer la nature du caractère émis ou reçu : adresse ou donnée. Voir le paragraphe « liaison multipoint » à la fin.

NBSTOP	Mode asynchrone	Mode synchrone
00	1 bit de stop	1 bit de stop
01	1,5 bit de stop	Réservé
10	2 bits de stop	2 bits de stop
11	réservé	Réservé

Le nombre de bits de stop permet de séparer les caractères. 1,5 et 2 sont des valeurs anciennes qui ne sont plus utilisées.

CHMODE	Mode de fonctionnement
00	Mode normal
01	Mode écho : TXD reliée à la réception
10	Bouclage local : liaison émission réception
11	Bouclage distant : liaison RXD-TXD

Les modes autres que normal servent à tester le fonctionnement de la liaison série en cas de panne ou d'erreur fréquentes.

• **MODE9: 9-bit Character Length**

Si ce bit vaut 0 c'est le tableau CHRL qui définit la longueur du caractère. Si ce bit vaut 1 alors le caractère fait 9 bits. Utilise pour envoyer du binaire octet par octet en y ajoutant un bit de parité géré par logiciel. Contraignant : on ne peut écrire que 8 bits dans les registres émission réception, et ce mode ne fonctionne pas avec le DMA.

• **CLKO: Clock Output Select**

Ce bit mis à 1 permet de disposer de l'horloge MCK ou MCK/8 sur la broche SCK . Dans le cas 0 la broche SCK n'est pas prise en compte.

**REGISTRES INTERRUPT ENABLE, DISABLE ET MASK REGISTERS.**

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	TXEMPTY	TIMEOUT
7	6	5	4	3	2	1	0
PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY

- **RXRDY: Enable/Disable/Mask RXRDY Interrupt**
- **TXRDY: Enable/Disable/Mask TXRDY Interrupt**
- **RXBRK: Enable/Disable/Mask Receiver Break Interrupt**
- **ENDRX: Enable/Disable/Mask End of Receive Transfer Interrupt**
- **ENDTX: Enable/Disable/Mask End of Transmit Interrupt**
- **OVRE: Enable/Disable/Mask Overrun Error Interrupt**
- **FRAME: Enable/Disable/Mask Framing Error Interrupt**
- **PARE: Enable/Disable/Mask Parity Error Interrupt**
- **TIMEOUT: Enable Time-out Interrupt**
- **TXEMPTY: Enable TXEMPTY Interrupt**

Chacune de ces 10 bascules peut être mise individuellement à 0 ou à 1, et lue.

3 de ces bascules concernent la gestion des IT de la partie émission et 7 celles de la partie réception.

**Pour la partie émission** : lorsque l'on écrit un caractère dans le registre de transmission US\_THR, ce dernier est aussitôt transféré dans un registre à décalage si ce dernier est libre, et la transmission commence avec les bits de START et de STOP. On peut ainsi être prévenu par interruption de l'échéance de ces étapes :

- **TXRDY** : Transmitter ready signifie que le registre de transmission US\_THR a été transféré dans le registre à décalage
- **TXEMPTY** : Transmitter empty signifie que les deux registres ont été vidés
- **ENDTX** : signifie que la transmission en mode DMA est terminée.

**Pour la partie réception** : les bits sont assemblés dans un registre à décalage et le caractère reconstitué est transféré dans le registre de réception US\_RHR. On peut être prévenu par interruption des situations suivantes :

- **RXRDY**: un caractère vient d'être transféré du registre à décalage dans le registre de réception
- **ENDRX**: signifie que la réception en mode DMA est terminée.
- **RXBRK**: on a reçu un caractère BREAK (caractère à 0 et absence de bit de STOP)
- **PARE**: (**PARity Error**) erreur de parité, si en service.
- **OVRE**: (**OveRRun**) erreur d'écrasement : un nouveau caractère assemblé dans le registre à décalage a été écrit dans le registre US\_RHR sans que ce dernier ait été au préalable lu (on a pas été assez rapide...)
- **FRAME**: erreur de cadrage c'est à dire absence de bit STOP à la fin du caractère
- **TIMEOUT**: indique que le délai maximum d'attente entre 2 caractères programmé dans le registre US\_TROP est dépassé.

*Note : si l'on travaille par interruption caractère par caractère, alors l'interruption TXRDY suffit pour gérer la partie émission et l'interruption RXRDY suffit pour la partie réception. Dans ce dernier cas, à chaque lecture d'un caractère il faudra aussi lire le registre US\_SSR pour voir s'il n'y a pas eu d'erreur de réception.*

## REGISTRE CHANEL STATUS REGISTER US\_CSR

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	TXEMPTY	TIMEOUT
7	6	5	4	3	2	1	0
PARE	FRAME	OVRE	ENDTX	ENDRX	RXBRK	TXRDY	RXRDY

Ce registre contient les 10 bascules qui mémorisent l'état de l'USART, à raison donc de 3 pour l'émission et 7 pour la réception :

- **RXRDY: Receiver Ready** : bit lu à 1 indique qu'un caractère au moins est arrivé depuis la dernière lecture du registre US\_RHR.
- **TXRDY: Transmitter Ready** : bit lu à 1 indique que le registre de transmission est vide et qu'un caractère peut être écrit dans le registre US\_THR.
- **RXBRK: Break Received/End of Break** : bit lu à 1 prévient de la détection d'un BREAK
- **ENDRX: End of Receiver Transfer** : bit lu à 1 indique une fin de transmission en mode DMA
- **ENDTX: End of Transmitter Transfer** : bit lu à 1 indique une fin de réception en mode DMA
- **OVRE: Overrun Error** : bit lu à 1 indique un écrasement
- **FRAME: Framing Error** : bit lu à 1 indique une erreur de cadrage
- **PARE: Parity Error** : bit lu à 1 indique une erreur de parité
- **TIMEOUT: Receiver Time-out** : bit lu à 1 indique un time-out
- **TXEMPTY: Transmitter Empty** : bit lu a un indique qu'il n'y a plus de caractère en cours d'émission

### USART Receiver Holding Register US\_RHR

C'est le registre de réception. Seuls les 8 bits de faibles poids sont utilisés. Si le caractère reçu fait moins de 8 bits, les bits absents sont lus à 0.

### USART Transmitter Holding Register US\_THR

C'est le registre d'émission. Seuls les 8 bits de faibles poids sont utilisés

### USART Baud Rate Generator Register US\_BRGR



Seuls les 16 bits de faibles poids de ce registre sont utilisés. Les valeurs que l'on doit y écrire sont les suivantes :

0 : invalide l'horloge. Valeur par défaut si l'USART n'est pas en service.

1 : indique que la source de l'horloge est externe, via la broche SCL

toute valeur entre 2 et 65535 donne le rapport de division de l'horloge.

Exemple, en mode asynchrone pour fonctionner à 1000 bits par seconde (bit par seconde est synonyme ici de baud rate) avec comme source l'horloge processeur on doit placer la valeur : fréquence de l'horloge processeur / 1000 avec comme horloge processeur soit MCK soit MCK/8, comme indiqué dans le registre MODE. Avec une horloge à 66MHz et un mode MCK la valeur à mettre est alors  $66000000/(16*1000)$  soit 4125.

### **USART Receiver Time-out Register US\_RTOR**

Permet de déclencher un time-out si le délai entre 2 caractères en réception devient supérieur à un seuil donné. Surtout utile en mode synchrone et/ou en mode DMA.

### **USART Transmitter Time-guard Register US\_TTGR**

Permet en émission d'introduire un délai entre l'émission de chaque caractère. Utile en cas de communication avec un périphérique lent (et probablement ancien) en mode asynchrone.

### **MODE de fonctionnement DMA ( Direct Memory Access ) : registres USART Receive Pointer Register, USART Receive Counter Register, USART Transmit Pointer Register, USART Transmit Counter Register.**

Le fonctionnement en mode DMA permet de transférer des paquets de caractères sans intervention du logiciel.

En émission il suffit de mettre l'adresse mémoire du tampon dans le registre US\_TPR et le compte de caractères dans le registre US\_TCR (max 64KO). Le transfert commence dès que le registre d'émission est vide.

Dans ce mode on fonctionne plutôt par interruption de fin de transfert ENDTX, lorsque le compteur passe à zéro.

On fait de même en réception avec les registres US\_RPR et US\_RCR.

Pour la réception il faut prévoir les cas d'erreurs possibles sur chaque caractère ainsi que de la fin de la réception avant le passage à zéro du compteur : il faut utiliser la possibilité de time-out en réception.

### **FONCTIONNEMENT EN MODE MULTIPPOINT**

En mode multipoint, une station est maître et les autres sont esclaves. Toutes sont connectées à la ligne. Le maître gère la ligne, c'est à dire qu'il interroge régulièrement les esclaves pour savoir s'ils ont quelque chose à émettre (« polling »). L'interrogation se fait esclave par esclave. Le message doit donc comporter une adresse, correspondante à l'adresse de l'esclave concerné. Le mécanisme mis en place dans cet USART permet de distinguer un octet de donnée d'un octet d'adresse et permet donc ce fonctionnement.

Si une station esclave a quelque chose à émettre, un dialogue s'établit entre le maître et l'esclave en respectant un protocole donné (pour la gestion des erreurs notamment) suite à une phase de polling.

Si le maître doit envoyer un message à une station esclave une séquence dite d' « addressing » doit alors s'établir : émission du message avec dans l'en-tête l'adresse du destinataire.

Ce type de fonctionnement a pratiquement disparu et remplacé par les protocoles plus sophistiqués des réseaux locaux

# ANNEXE G

Elle peut être due à une erreur sur la référence de tension, sur les gains des ampli utilisés ou encore un mauvais appareillage d'un réseau de résistance. On la définit par rapport à la pleine échelle du convertisseur (code 11...1).

On peut la compenser par un circuit externe qui annule l'erreur.

### II.3 Erreur de linéarité différentielle (DNL)

Elle est définie pour chaque palier du convertisseur et elle représente la différence entre la largeur du palier réelle et la largeur idéale. On l'exprime en nombre de LSB.

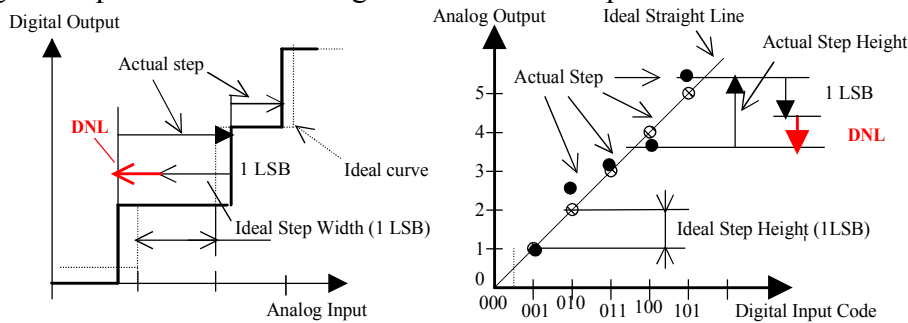


Figure 5 : Differential NonLinearity CAN - CNA

### II.4 Erreur de linéarité intégrale (INL)

Elle est définie par la position de la courbe réelle par rapport à la courbe idéale.

Dans le cadre du CAN, il existe deux définitions de l'INL. La première caractérise la différence entre la position des flancs, une erreur nulle correspond donc à une INL nulle. La deuxième caractérise la différence entre le milieu des paliers et les flancs. Une erreur nulle correspond alors à un INL d'un demi-quantum.

Pour le CNA, il n'y a qu'une seule définition. C'est la différence entre le point théorique et le point pratique.

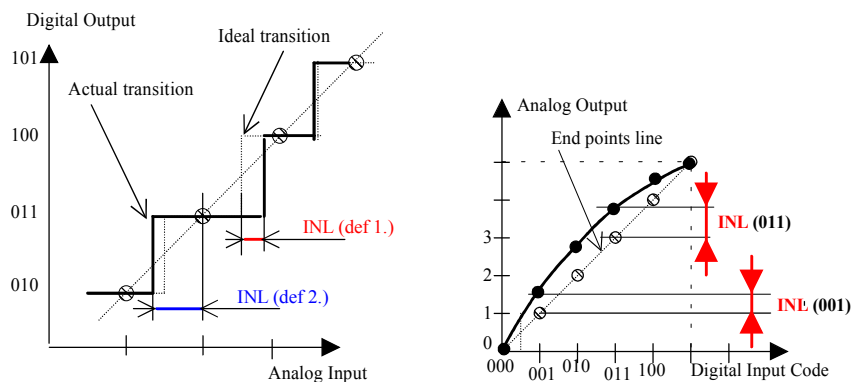


Figure 6 : Integral NonLinearity CAN - CNA

### II.5 Caractéristique totale sans compensation

C'est l'erreur obtenue sans aucune compensation (offset, gain, INL, DNL...). On la définit par la différence maximum à chaque palier entre les flancs et le milieu d'un palier idéal.

## الملخص

تم تحقيق هذه الدراسة في مركز تكنولوجيا Schlumberger Riboud Product Center تهدف إلى تقييم القدرات جهاز رقابة مخصص لتطبيقات في ميدان البترول ذوا الحرارة العالية . النتائج المحصل عليها بينت أن إستغلال ال MSC1201Y3 من تيكصاص أنسترومنت يبقى مقبول حتى درجة حرارة 150 درجة مئوية . لكن بنسبة لدرجة أعلى القدرات على مستوى ذكره البرنامج والساعة الداخلية تتدهور بصفة سريعة

## Résumé :

Notre étude a été réalisée au sein du centre technologique Schlumberger Riboud Product Center ( **SRPC** ).

Elle consiste à évaluer les performances d'un microcontrôleur dédié à des applications oilfield en haute température.

Les résultats obtenus ont démontré que le **MSC1201Y3** de Texas Instruments reste très fiable jusqu'à **150°C**, au delà de cette température les performances au niveau de la mémoire programme et l'horloge interne diminuent considérablement.

## Abstract :

This Project has been fulfilled in Schlumberger Riboud Product Center ( **SRPC** ).

It consists in the qualification of a microcontroller dedicated to oilfield applications in high temperature.

The tests realized on the MSC1201Y3 of Texas Instruments showed that it steels very efficient until 150°C, above this temperature the performance of the flash memory and the internal clock decrease heavily.